

*Журнал «Научное обозрение.
Технические науки»
зарегистрирован Федеральной службой
по надзору в сфере связи, информационных
технологий и массовых коммуникаций.
Свидетельство ПИ № ФС77-57440
ISSN 2500-0799*

*Учредитель, издательство и редакция:
ООО НИЦ «Академия Естествознания»,
Почтовый адрес: 105037, г. Москва, а/я 47
Адрес редакции: 410056, Саратовская область,
г. Саратов, ул. им. Чапаева В.И., д. 56*

**Founder, publisher and edition:
LLC SPC Academy of Natural History,
Post address: 105037, Moscow, p.o. box 47
Editorial address: 410056, Saratov region,
Saratov, V.I. Chapaev Street, 56**

*Подписано в печать 28.10.2020
Дата выхода номера 29.11.2020
Формат 60×90 1/8*

*Типография
ООО НИЦ «Академия Естествознания»,
410035, Саратовская область,
г. Саратов, ул. Мамонтовой, д. 5*

**Signed in print 28.10.2020
Release date 29.11.2020
Format 60×90 8.1**

**Typography
LLC SPC «Academy Of Natural History»
410035, Russia, Saratov region,
Saratov, 5 Mamontovoi str.**

*Технический редактор Байгузова Л.М.
Корректор Галенкина Е.С., Дудкина Н.А.*

*Тираж 1000 экз.
Распространение по свободной цене
Заказ НО 2020/5
© ООО НИЦ «Академия Естествознания»*

Журнал «НАУЧНОЕ ОБОЗРЕНИЕ» выходил с 1894 по 1903 год в издательстве П.П. Сойкина. Главным редактором журнала был Михаил Михайлович Филиппов. В журнале публиковались работы Ленина, Плеханова, Циолковского, Менделеева, Бехтерева, Лесгафта и др.

Journal «Scientific Review» published from 1894 to 1903. P.P. Soykin was the publisher. Mikhail Filippov was the Editor in Chief. The journal published works of Lenin, Plekhanov, Tsiolkovsky, Mendelev, Bekhterev, Lesgaft etc.



М.М. Филиппов (M.M. Philippov)

С 2014 года издание журнала возобновлено
Академией Естествознания

**From 2014 edition of the journal resumed
by Academy of Natural History**

Главный редактор: М.Ю. Ледванов
Editor in Chief: M.Yu. Ledvanov

Редакционная коллегия (**Editorial Board**)

А.Н. Курзанов (**A.N. Kurzanov**)
Н.Ю. Стукова (**N.Yu. Stukova**)
М.Н. Бизенкова (**M.N. Bizenkova**)
Н.Е. Старчикова (**N.E. Starchikova**)
Т.В. Шнуровозова (**T.V. Shnurovozova**)

НАУЧНОЕ ОБОЗРЕНИЕ • ТЕХНИЧЕСКИЕ НАУКИ

SCIENTIFIC REVIEW • TECHNICAL SCIENCES

www.science-education.ru

2020 г.



***В журнале представлены научные обзоры,
статьи проблемного
и научно-практического характера***

***The issue contains scientific reviews,
problem and practical scientific articles***

СОДЕРЖАНИЕ

Технические науки (05.09.00, 05.11.00, 05.12.00, 05.13.00)

СТАТЬИ

ИССЛЕДОВАНИЕ ДЕЯТЕЛЬНОСТИ СЕРВИСНОГО ЦЕНТРА КОМПЬЮТЕРНОЙ ТЕХНИКИ И АНАЛИЗ ОСУЩЕСТВИМОСТИ АВТОМАТИЗАЦИИ БИЗНЕС-ПРОЦЕССОВ	
<i>Абсатаров Р.Н., Абрамова О.Ф.</i>	5
МИГРАЦИЯ ПРОЕКТА НА ЯЗЫК ПРОГРАММИРОВАНИЯ TYPESCRIPT ПРИ РАЗРАБОТКЕ НА ФРЕЙМВОРКЕ NODE.JS	
<i>Вахрамов С.В., Мусин А.М., Ризванов Д.А., Хамитов М.А.</i>	11
ОБРАБОТКА СТАТИСТИЧЕСКИХ ДАННЫХ МЕТОДОМ ГЛУБОКОГО ОБУЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ МОДУЛЯ KERAS	
<i>Ильичев В.Ю., Юрик Е.А.</i>	16
ПРИМЕНЕНИЕ ЯЗЫКА PYTHON ДЛЯ РАЗРАБОТКИ КОНВЕРТЕРА ВАЛЮТ	
<i>Ошурков А.Ю., Бужинская Н.В.</i>	21
АНАЛИЗ ДОСТОИНСТВ И НЕДОСТАТКОВ КВАНТОВЫХ ЭМУЛЯТОРОВ НА ПРИМЕРЕ ВЗАИМОДЕЙСТВИЯ С ПОЛЬЗОВАТЕЛЕМ	
<i>Третьяк М.А., Щекатурин А.Е., Пилипенко И.А., Кравченко В.О., Черкесова Л.В.</i>	27
ИССЛЕДОВАНИЕ МОДЕЛЕЙ СО СКРЫТЫМ РАССТОЯНИЕМ ДЛЯ ВНЕДРЕНИЯ СУЩНОСТЕЙ И ИХ ОТНОШЕНИЙ В ГРАФ ЗНАНИЙ	
<i>Хлопенкова А.Ю., Белов Ю.С.</i>	38
СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ОБЪЕКТОВ В БАЗЕ ДАННЫХ ORACLE	
<i>Журавлев А.А., Аксенов К.А.</i>	43
ОЦЕНКА МИРОВОГО ОПЫТА ОСНАЩЕНИЯ И ЭКСПЛУАТАЦИИ ЭКСПОРТНЫХ ПЕРЕГРУЗОЧНЫХ КОМПЛЕКСОВ ПО ПЕРЕВАЛКЕ УГЛЯ В МОРСКИХ ПОРТАХ	
<i>Мазуренко О.И.</i>	48
ОСНОВНЫЕ ПРИНЦИПЫ УПРАВЛЕНИЯ РАБОТОЙ ФЛОТА МОРСКОЙ СУДОХОДНОЙ КОМПАНИИ В СОВРЕМЕННЫХ РЫНОЧНЫХ УСЛОВИЯХ	
<i>Кирносов Д.А.</i>	58
ИССЛЕДОВАНИЕ И РАЗРАБОТКА ЭКСПЕРИМЕНТАЛЬНОГО ОБРАЗЦА АКУСТИЧЕСКОГО ИЗМЕРИТЕЛЯ СКОРОСТИ И НАПРАВЛЕНИЯ ТЕЧЕНИЙ	
<i>Лискин В.А., Тихонова Н.Ф.</i>	65

CONTENTS

Technical sciences (05.09.00, 05.11.00, 05.12.00, 05.13.00)

ARTICLES

A STUDY OF THE ACTIVITIES OF A COMPUTER EQUIPMENT SERVICE CENTER AND AN ANALYSIS OF THE FEASIBILITY OF AUTOMATING BUSINESS PROCESSES	
<i>Absatarov R.N., Abramova O.F.</i>	5
MIGRATION OF A PROJECT WRITTEN USING NODE.JS FRAMEWORK TO TYPESCRIPT PROGRAMMING LANGUAGE	
<i>Vakhramov S.V., Musin A.M., Rizvanov D.A., Khamitov M.A.</i>	11
PROCESSING OF STATISTICS BY DEEP LEARNING USING KERAS MODULE	
<i>Ilichev V.Y., Yurik E.A.</i>	16
USING PYTHON TO DEVELOP A CURRENCY CONVERTER	
<i>Oshurkov A.Y., Buzhinskaya N.V.</i>	21
ANALYSIS OF THE ADVANTAGES AND DISADVANTAGES OF QUANTUM EMULATORS ON THE EXAMPLE OF INTERACTION WITH THE USER	
<i>Tretyak M.A., Schekaturin A.E., Pilipenko I.A., Kravchenko V.O., Cherkesova L.V.</i>	27
RESEARCH OF LATENT DISTANCE MODELS FOR EMBEDDING OF ENTITIES AND THEIR RELATIONSHIPS IN THE KNOWLEDGE GRAPH	
<i>Khlopenkova A.Yu., Belov Yu.S.</i>	38
CREATING AND USING OBJECTS IN ORACLE DATABASE	
<i>Zhuravlev A.A., Aksenov K.A.</i>	43
WORLD EXPERIENCE OF EQUIPPING AND OPERATING OF COAL EXPORT TRANSSHIPMENT COMPLEXES	
<i>Mazurenko O.I.</i>	48
BASIC PRINCIPLES OF FLEET MANAGEMENT OF A MARITIME SHIPPING COMPANY IN MODERN MARKET CONDITIONS	
<i>Kirnosov D.A.</i>	58
RESEARCH AND DEVELOPMENT OF EXPERIMENTAL SAMPLE OF ACOUSTIC SPEED METER AND DIRECTION OF FLOW	
<i>Liskin V.A., Tikhonova N.F.</i>	65

СТАТЬИ

УДК 004.418

**ИССЛЕДОВАНИЕ ДЕЯТЕЛЬНОСТИ СЕРВИСНОГО ЦЕНТРА
КОМПЬЮТЕРНОЙ ТЕХНИКИ И АНАЛИЗ ОСУЩЕСТВИМОСТИ
АВТОМАТИЗАЦИИ БИЗНЕС-ПРОЦЕССОВ****Абсатаров Р.Н., Абрамова О.Ф.***Волжский политехнический институт (филиал) ФГБОУ ВПО «Волгоградский государственный
технический университет», Волжский, e-mail: absatarow.ruslan@yandex.ru*

Темой данной статьи является исследование предметной области деятельности сервисного центра по ремонту и обслуживанию компьютерной техники. В статье проведен анализ осуществимости автоматизации выделенных бизнес-процессов предприятия. Целью работы является получение знаний о методических подходах и технологических средствах разработки проектов информационных систем, методике моделирования и анализа предметной области. Ежедневно в сервисный центр поступает большой поток заявок от клиентов на оказание услуг по ремонту компьютерного оборудования. Обслуживание каждой такой заявки предполагает наличие большого числа бизнес-процессов, часть из которых можно автоматизировать, сняв тем самым рутинную нагрузку с сотрудников, повысив производительность труда, уменьшив возникновение возможных ошибок. Прежде чем внедрить информационную систему, которая поможет усилить контроль выполнения ремонтных работ и оптимизировать рабочее время персонала, необходимо оценить целесообразность и осуществимость разработки программно-информационной системы для учета работ по ремонту и обслуживанию компьютерной техники в сервисном центре. Для этого необходимо исследовать предметную область деятельности сервисного центра, выделить существующие недостатки, оценить положительный эффект, который может дать внедрение информационной системы.

Ключевые слова: предметная область, анализ, автоматизация**A STUDY OF THE ACTIVITIES OF A COMPUTER EQUIPMENT
SERVICE CENTER AND AN ANALYSIS OF THE FEASIBILITY
OF AUTOMATING BUSINESS PROCESSES****Absatarov R.N., Abramova O.F.***Volzhskiy Polytechnical Institute, branch of the Volgograd State Technical University,
Volzhskiy, e-mail: absatarow.ruslan@yandex.ru*

The topic of this article is the study of the subject area of the service center for the repair and maintenance of computer equipment. This article analyzes the feasibility of automating the selected business processes of an enterprise. The aim of this work is to gain knowledge about the methodological approaches and technological tools for the development of information systems projects, methods of modeling and analysis of the subject area. Every day, a large flow of requests from customers for the provision of computer equipment repair services arrives at the service center. Serving each such application requires the presence of a large number of business processes, some of which can be automated, thereby removing the routine burden from employees, increasing labor productivity, and reducing the occurrence of possible errors. Before introducing an information system that will help strengthen control over the progress of repair work and optimize staff time, it is necessary to assess the feasibility and possibility of developing a software-information system for accounting for repair and maintenance of computer equipment in a service center. For this, it is necessary to study the subject area of the service center, highlight the existing shortcomings, evaluate the positive effect that the introduction of the information system can give.

Keywords: subject area, analysis, automation

На данный момент в сервисном центре не применяются какие-либо средства автоматизации существующих бизнес-процессов. Вся деятельность компании производится вручную, а информация, необходимая для функционирования предприятия, представлена исключительно на бумажных носителях. Это свидетельствует о наличии серьезных проблем, приводящих к снижению производительности труда, сокращению прибыли и снижению качества обслуживания клиентов [1]. Один из ключевых недостатков в текущей ситуации – большая зависимость качества обслужива-

ния от опыта конкретных сотрудников. Все свидетельствует о низком уровне зрелости предприятия.

Целью данной работы является исследование предметной области, выявление существующих проблем и анализ осуществимости разработки программно-информационной системы для учета работ по ремонту и обслуживанию компьютерной техники в сервисном центре.

Материалы и методы исследования

Для исследования были использованы такие методы, как анализ и обобщение биз-

нес-процессов, протекающих на предприятии, теоретическое исследование методик и технологических средств разработки проектов информационных систем.

Результаты исследования и их обсуждение

Для выявления и анализа недостатков в деятельности сервисного центра необходимо произвести исследование предметной области [2].

Организационная структура предприятия изображена на рис. 1.

Матрица организационных проекций сервисного центра приведена в табл. 1.

Из данной модели видно, что ключевым источником информации служит журнал, который может потребоваться одновременно в различных отделах СЦ [3]. Также возникают трудности с ограничением прав доступа к прейскуранту и журналу: инженер и приемщик могут просматривать цены услуг, но не имеют права вносить изменения, это касается и журнала.

Проектируемая ПИС должна устранить существующие недостатки: предоставить параллельный доступ к ключевой информации, ограничить права доступа к базе данных.

Общая диаграмма «Обслуживание клиентов» представлена на рис. 2.

Модель отражает общий принцип работы предприятия по сервисному обслуживанию компьютерной техники клиентов [3]. Первичным входным потоком данных служит заявка клиента на ремонт оборудования, а входным материальным объектом является неисправная техника. Конечным выходным объектом является акт выполненных работ и техника, прошедшая обслуживание и осмотренная при выдаче клиентом. В процессе обслуживания клиентов управленческими потоками данных выступают законодательство, должностные инструкции, а также инструкции по охране труда и технике безопасности.

Ресурсами являются: клиент, персонал СЦ (приемщик, инженер, менеджер) и средства труда, которые включают в себя офисную технику, информационное оборудование, сервисное оборудование, производственные здания, инструменты и приспособления.

Для более углубленного анализа необходимо произвести функциональную декомпозицию, результат которой представлен на рис. 3.

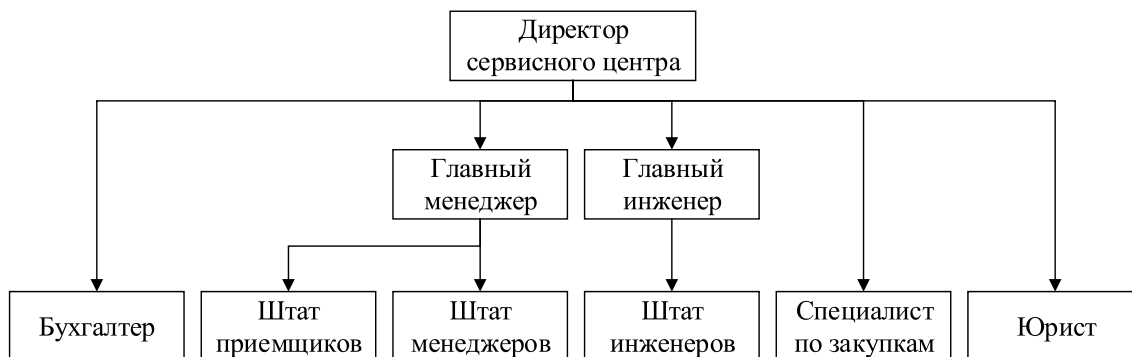


Рис. 1. Организационная структура предприятия

Таблица 1

Матрица организационных проекций

	Приемщик	Инженер	Менеджер
Прием заказа, составление акта приема	✓		
Просмотр перечня заказов	✓	✓	✓
Изменение статуса заказа	✓	✓	✓
Просмотр прейскуранта	✓	✓	✓
Внесение изменений в прейскурант			✓
Внесение изменений в журнал		✓	✓
Выдача заказа, составление акта выполненных работ			✓

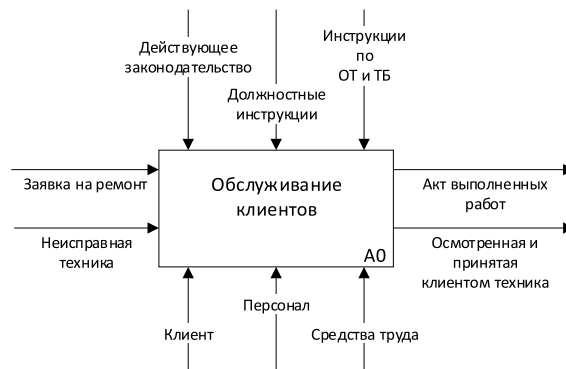


Рис. 2. Диаграмма процесса «Обслуживание клиентов»

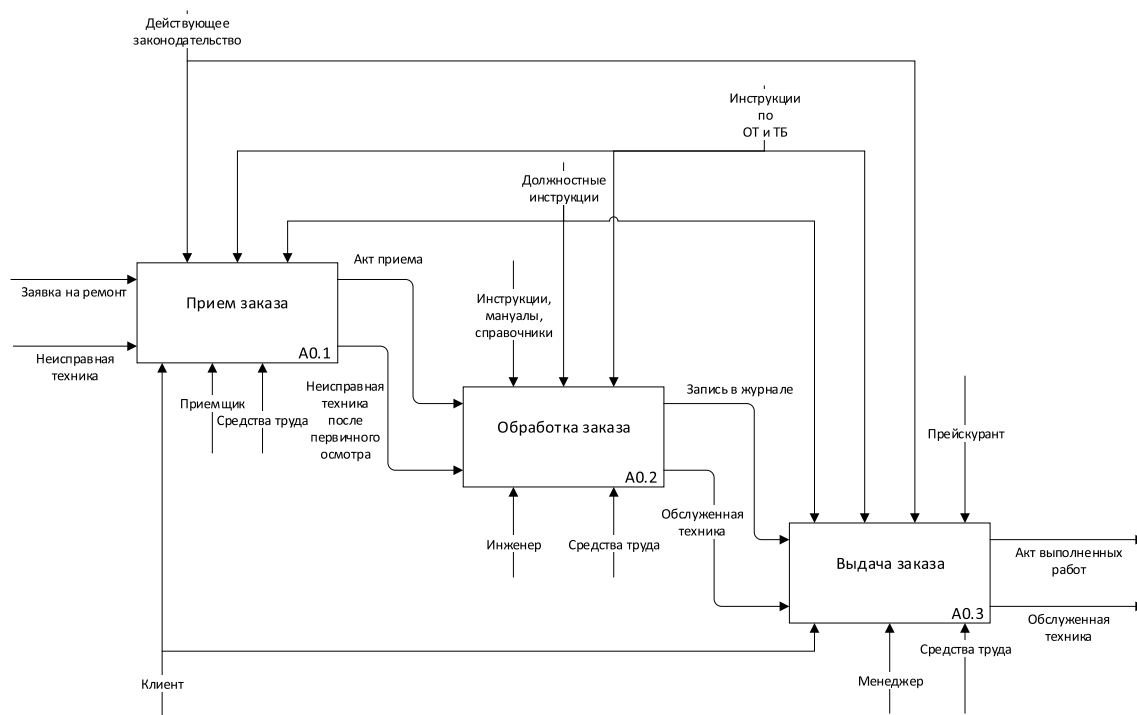


Рис. 3. Диаграмма декомпозиции «Обслуживание клиентов»

На данной диаграмме представлена декомпозиция наиболее общего процесса «Обслуживание клиентов». Модель отражает внутреннюю структуру процесса, в частности видно, что главный процесс состоит из подпроцессов:

1. Прием заказа.
2. Обработка заказа.
3. Выдача заказа.

На рис. 4–6 отображен результат дальнейшей декомпозиции каждого из представленных процессов.

Документооборот предприятия приведен в табл. 2.

Движение документов отражает организационно-управленческую структуру аппарата управления и характер распределения обязанностей между его составными частями. Проектируемая система должна усовершенствовать существующую модель документооборота, что приведет к совершенствованию системы управления в целом.

Разрабатываемая ПИС должна позволять автоматически формировать и корректировать такие документы, как акт приема, акт выполненных работ, прейскурант.

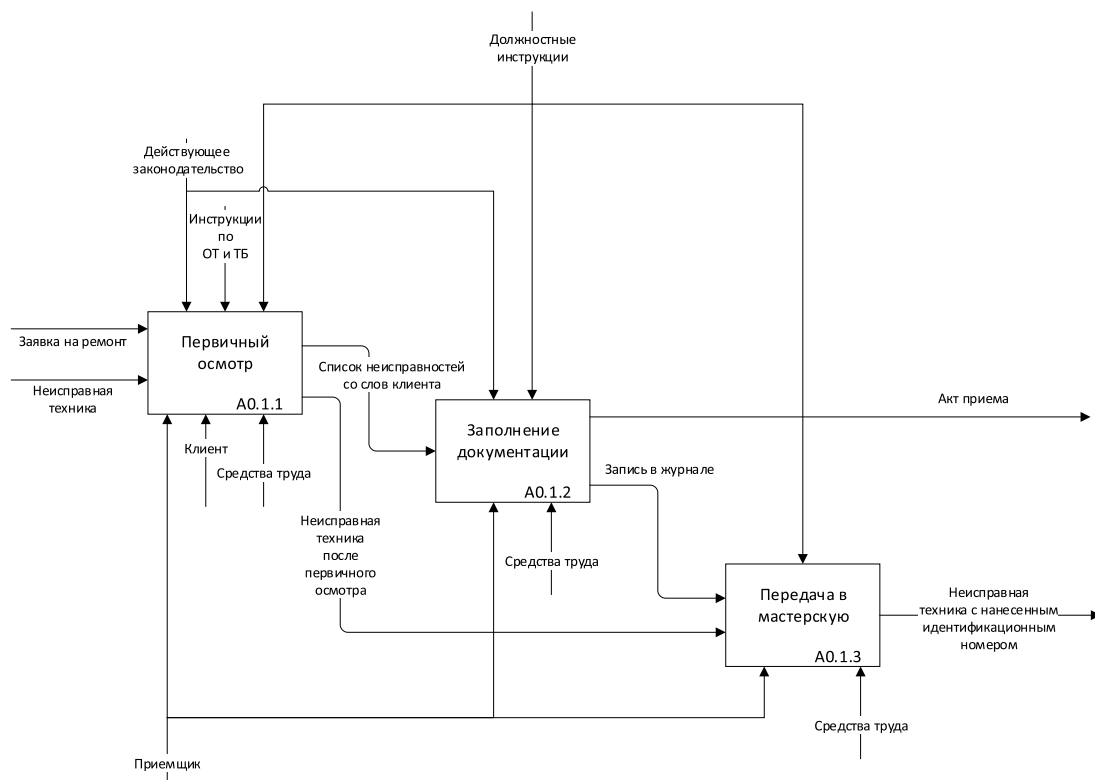


Рис. 4. Диаграмма процесса «Прием заказа»

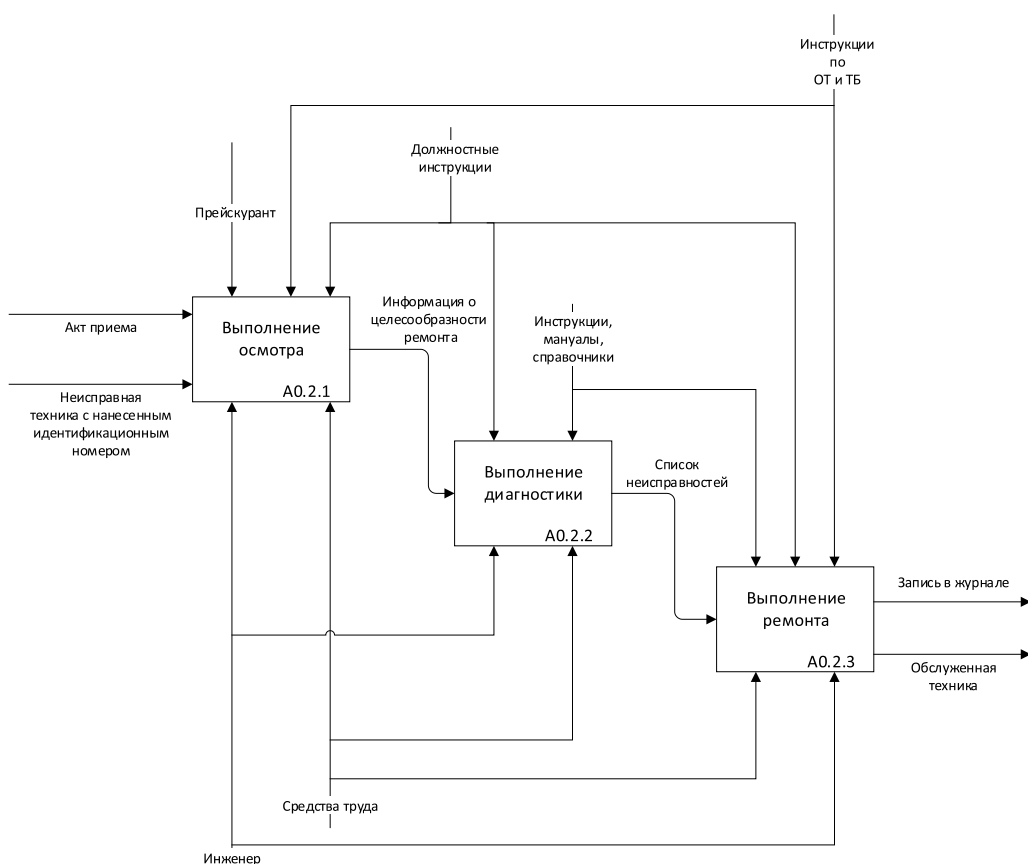


Рис. 5. Диаграмма процесса «Обработка заказа»

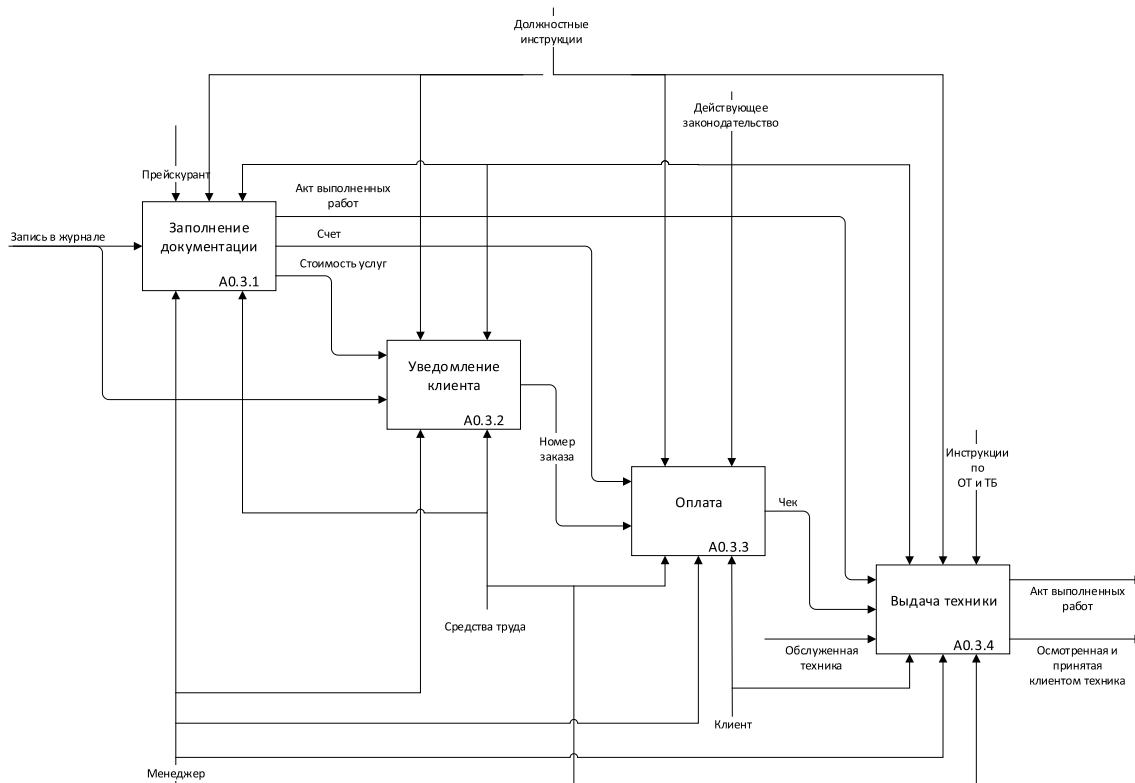


Рис. 6. Диаграмма процесса «Выдача заказа»

Таблица 2

Документооборот сервисного центра

Составляемый документ	Операция	Исполнитель	Частота	Входящие документы	Реестр документа
Акт приема	Прием техники в СЦ	Приемщик	При каждом новом заказе	Технический паспорт устройства	Реестр актов приема
Прейскурант	Формирование цен	Менеджер	При добавлении нового вида ремонтных работ	Каталог предоставляемых услуг	Реестр прейскурантов
Акт выполненных работ	Отчет о проделанной работе	Менеджер	При каждом выполненном заказе	Запись в журнале о выполненном заказе	Реестр актов выполненных работ
Счет	Формирование итоговой стоимости предоставленных услуг	Менеджер	При каждом выполненном заказе	Акт выполненных работ	Реестр счетов
Чек	Оплата предоставленных услуг	Бухгалтер	При каждой оплате счета	Счет	Реестр чеков

При исследовании предметной области деятельности сервисного центра был выявлен перечень проблем:

- невозможность одновременного доступа к информации, необходимой для исполнения заказа клиента;
- невозможность разграничить права доступа для разных сотрудников;

- сложность с отслеживанием передвижения заказа между различными этапами;
- невозможность получения своевременной отчетной информации по заказам, находящимся в обработке;
- отсутствие единой базы данных по неисправностям компьютерного оборудования и путей устранения этих неисправностей.

Заключение

Внедрение автоматизированной системы для учета работ по ремонту и обслуживанию компьютерной техники поможет решить указанные проблемы, а также повысить скорость и качество обслуживания клиентов [4]. Это поможет усилить контроль за ходом выполнения ремонтных работ и оптимизировать рабочее время персонала. Актуальность разработки обуславливается переходом организации на новый уровень зрелости благодаря внедрению информационной системы, поскольку это приводит к снижению зависимости качества обслуживания клиентов от профессионализма отдельных сотрудников.

Основными целями создания ПИС являются:

- повышение скорости обслуживания клиентов за счет оперативного анализа неисправностей и сокращения времени на выбор варианта их устранения;
- формирование единой базы знаний по неисправностям компьютерного оборудования, поступающего в фирму;

– повышение качества обслуживания клиентов путем снижения влияния человеческого фактора на ключевые этапы обработки поступающих заявок;

– снижение нагрузки на сотрудников во время пиковых нагрузок.

Реализация вышеизложенных пунктов преследует ключевую цель – повышение прибыли предприятия [5].

Список литературы

1. Корсакова М. Ивент-агентство «Альфа» на грани краха. Правила построения эффективной сервисной компании. М.: Альпина Паблишер, 2012. 192 с.
2. Васильев С.С., Абрамова О.Ф., Адамов А.С. Исследование и анализ проблем в области автоматизации бизнес-процессов отдела снабжения // Форум молодых учёных: электрон. науч. журнал. 2017. № 5 (9). С. 382–392. [Электронный ресурс]. URL: http://forum-nauka.ru/_5_9_may_2017 (дата обращения: 12.07.2020).
3. Рыбанов А.А., Филиппова Е.М., Свиридова О.В., Федотова Л.А. Система количественных показателей мониторинга за процессом развития навыка ввода информации // Педагогическая информатика. 2020. № 1. С. 136–142.
4. Остроух А., Нгуен Дык Тхань, Чернов Э. Автоматизация управления производством. М.: LAP Lambert Academic Publishing, 2013. 284 с.
5. Щербаков В.В., Мерзляк А.В., Коскур-Оглы Е.О. Автоматизация бизнес-процессов в логистике. СПб.: Питер, 2016. 464 с.

УДК 004.416.6

МИГРАЦИЯ ПРОЕКТА НА ЯЗЫК ПРОГРАММИРОВАНИЯ TYPESCRIPT ПРИ РАЗРАБОТКЕ НА ФРЕЙМВОРКЕ NODE.JS

Вахрамов С.В., Мусин А.М., Ризванов Д.А., Хамитов М.А.

*ФГБОУ ВО «Уфимский государственный авиационный технический университет»,
Уфа, e-mail: s@vakhramoff.ru*

Большинство современных Node.js-проектов написаны на языке программирования JavaScript. За последние два года TypeScript (надмножество над JavaScript) набрал очень большую популярность. В статье рассматривается процесс миграции Node.js проектов с языка программирования JavaScript на TypeScript. Проанализированы тренды популярности языка, приведены графики, показывающие её рост, пошагово описан алгоритм действий для решения поставленной задачи. Данный способ учитывает особенности языка TypeScript и специализированных библиотек, которые работают с фреймворком Node.js. Кроме того, описан процесс типизации библиотек, включая коды команд, позволяющие выполнить миграцию с помощью утилиты терминала в macOS или командной строки в Windows. В дополнение описаны особенности данного перехода, а также его необходимость при использовании достаточно больших проектов, имеющих свойство увеличиваться в объёме в пересчёте на количество строк кода и количество исходных файлов. Результаты миграции подробно описывают преимущества и недостатки самого языка TypeScript. В заключение приводится статистика, полученная в результате применения данного подхода на одном из собственных проектов. Удалось снизить количество ошибок разработки на 15%, облегчить отладку кода на 10%, а знакомство с проектом новых разработчиков стало до 50% быстрее.

Ключевые слова: TypeScript, JavaScript, Node.js, npm, API, back-end, lebab, EcmaScript, миграция проекта, популярность TypeScript

MIGRATION OF A PROJECT WRITTEN USING NODE.JS FRAMEWORK TO TYPESCRIPT PROGRAMMING LANGUAGE

Vakhramov S.V., Musin A.M., Rizvanov D.A., Khamitov M.A.

Ufa State Aviation Technical University, Ufa, e-mail: s@vakhramoff.ru

Many Node.js projects are written in TypeScript (a superset of JavaScript). This article describes the migration process of Node.js projects from JavaScript to TypeScript programming language. The popularity of TypeScript has been analyzed, and step-by-step algorithm of migration process was described to solve the problem. This approach also considers TypeScript and special libraries' features, which work with Node.js framework. Besides that, the typification process for libraries was described, including special terminal (in macOS) or console (in Windows) commands which lead to perform necessary tasks. In addition to all of that, the migration specificity and necessity was described in case of developing huge projects, which grow in a long-term period of time counting in lines of codes and source files amount. The results of the migration show advantages and disadvantages of the TypeScript language itself. In conclusion, the statistics of this approach was shown with regard to one of our personal projects. By using this approach mistakes in project were reduced by 15%.

Keywords: TypeScript, JavaScript, Node.js, npm, API, back-end, lebab, EcmaScript, project migration, TypeScript popularity

Успешное продвижение языка TypeScript в рейтинге языков программирования обусловлено его основанностью на шаблонах: это способствует реализации принципа повторного использования и позволяет смешиваться с существующей кодовой базой в JavaScript, способствуя получению более безопасного типизированного кода.

По данным Google Trends с рисунка можно увидеть растущую популярность TypeScript. За последние 5 лет рост популярности запроса составил 80 баллов: с 12 до 92 баллов, приблизившись к максимальному пику в 100 баллов – когда запрос показывает наивысший уровень популярности.

Весомая причина перехода на TypeScript – уменьшение количества затрачиваемого времени на тестирование своего кода: объём кода уменьшается на 10–20% при миграции с JavaScript на TypeScript.

Интерес среди пользователей сети Интернет равномерно увеличивается из года в год на протяжении более пяти лет, об этом свидетельствует рейтинг популярности языков программирования, по версии аналитической компании RedMonk, за период первого квартала 2020 года, изображенного на рис. 2.

Согласно рейтингу, TypeScript набирает популярность, занимая 9-ю строчку, что видно из рейтинга, представленного на рис. 3.

Согласно статистике с сайта github.com, TypeScript находится на 5-м месте среди языков, набирающих популярность.

Сообщество активно развивает TypeScript: на сегодняшний день существует около 7000 пакетов, поддерживающих описание типов для языка программирования TypeScript для уже существующих JavaScript-пакетов [4]. Количество пакетов,

изначально написанных на самом языке TypeScript, исчисляется десятками тысяч.

Проекты на Node.js используются для организации работы с серверной частью, преимущественно с данными пользователей. Эти данные часто включают в себя персональные данные людей.

Правильный проект на Node.js должен быть грамотно структурирован: отдельно

выделены модели и контроллеры. Использование TypeScript позволяет с помощью интерфейсов дополнительно описать представления, что в будущем не даст допустить ошибку в программном коде проекта, например путём передачи строки вместо числа, что является самой частой проблемой при разработке на JavaScript, которую особенно трудно обнаружить при отладке.

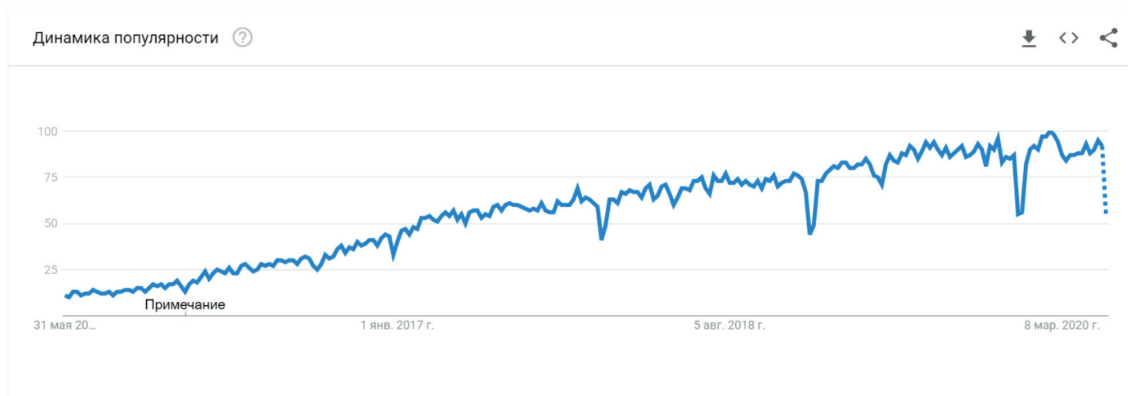


Рис. 1. Динамика популярности Google Trend по запросу TypeScript за 2015–2020 гг. [1]

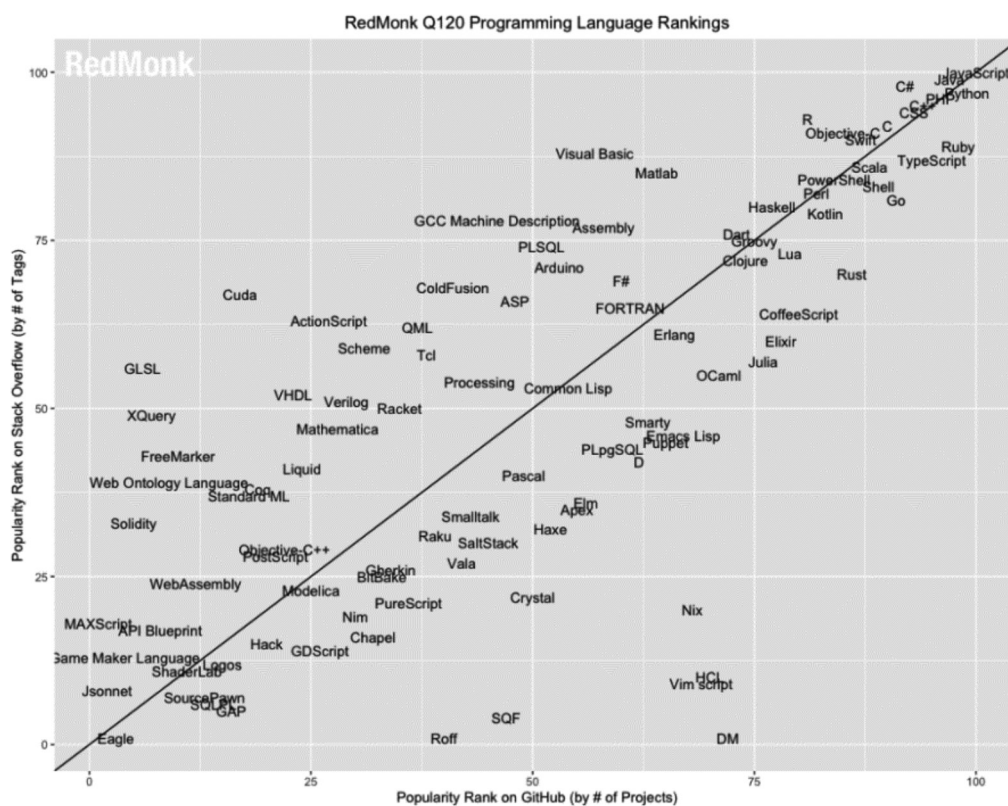


Рис. 2. Рейтинг популярности языков программирования, по версии аналитической компании RedMonk, за первый квартал 2020 года [2]

CHANGE IN PROGRAMMING LANGUAGE USE, 2018-2019		
01	Dart	532%
02	Rust	235%
03	HCL	213%
04	Kotlin	182%
05	TypeScript	161%
06	PowerShell	154%
07	Apex	154%
08	Python	151%
09	Assembly	149%
10	Go	147%

Рис. 3. Рост популярности языков программирования за 2018–2019 гг. [3]

Дополнительным преимуществом использования TypeScript является мощный механизм описания типов: мы не просто описываем базовые типы, с помощью общих (Generic) типов имеем

возможность описать абстрактную конструкцию, которую в дальнейшем сможем использовать вкупе с различными базовыми типами.

Например:

```
interface GenericIdentityFn {
  <T>(arg: T): T;
}
function identity<T>(arg: T): T {
  return arg;
}
let myIdentity: GenericIdentityFn = identity;
```

JavaScript является языком динамическим строго типизированным, в то время как TypeScript статическим слабо типизированным. Рассмотрим далее преимущество каждого из подходов [5].

Преимущества статической типизации (static typing):

- проверки типов происходят один раз на этапе компиляции, не нужно выяснять, нет ли попытки деления числа на строку (и затем выдать ошибку или осуществить преобразование);

- скорость выполнения: статически типизированные языки в большинстве случаев быстрее динамически типизированных ввиду предыдущего пункта;

- позволяет обнаруживать потенциальные ошибки уже на этапе компиляции;

- ускоряет процесс разработки (при выдаче подсказок заранее отсеиваются варианты, не подходящие по типу).

Преимущества динамической типизации (dynamic typing):

- легко создаются универсальные коллекции, содержащие в себе данные разных типов;

- удобно писать обобщённые алгоритмы (сортировка не только числового массива, а массива строк или произвольных структур данных);

- языки с динамической типизацией легки для освоения. Они очень хороши для того, чтобы начать программировать.

Преимущества сильной типизации (strong typing):

– надежность – вместо неожиданного поведения в рантайме программист получает ошибку на этапе компиляции;

– скорость – в коде нет скрытых преобразований типов, при написании кода программист может точно предположить, будет ли участок кода затрачивать много времени на выполнения;

– понимание работы программы – чётко понятно, какая структура с какой сравнивается, как преобразуются данные: не возникает ситуации «волшебного» приведения типов;

– определенность – при выполнении преобразований вручную ясно, какие данные и каким образом преобразуются из начальной формы в заданную.

Преимущества слабой типизации (weak typing):

– возможно использование смешанных выражений (например, из вещественных и целых чисел);

– абстрагирование от типизации и сосредоточение на задаче;

– краткость записи.

Цель исследования: исследование процесса миграции проекта, написанного на фреймворке Node.js, с языка программирования JavaScript на язык TypeScript с целью ускорения разработки проекта. Выявление преимуществ и недостатков использования языка TypeScript.

Материалы и методы исследования

Миграция проекта требует использования терминала (командной строки). При выполнении команд, описанных далее, будут установлены необходимые зависимости для перехода на язык TypeScript.

1. Установить npm-пакет TypeScript:

```
npm install --save-dev typescript@latest
```

2. Установить TS-Node:

```
npm install --save-dev ts-node@latest
```

3. Сконфигурировать компилятор TypeScript минимально рекомендуемыми настройками:

```
{
  "compilerOptions": {
    "module": "commonjs",
    "target": "esnext",
    "sourceMap": true,
    "moduleResolution": "node",
    "outDir": "dist"
  },
  "lib": ["es2020"],
  "compileOnSave": false,
  "exclude": [
    "node_modules"
  ]
}
```

4. Переписать код на ES6-синтаксис.

5. Автоматизировать процесс можно с помощью утилиты lebab [6].

6. Переименовать файлы с расширением .js на расширения .ts.

7. Установить типизированные пакеты для инструментов, используемых с Node.js.

В нашем случае необходимыми для установки были следующие пакеты:

```
npm install --save-dev @types/argon2
npm install --save-dev @types/dotenv
npm install --save-dev @types/express
npm install --save-dev @types/express-jwt
npm install --save-dev @types/jsonwebtoken
npm install --save-dev @types/mongoose
npm install --save-dev @types/node
```

8. Настроить nodemon для перекомпиляции проекта при изменении ts-файлов путём изменения nodemon.json

```
{
  "watch": ["**/*.ts"],
  "ignore": ["**/*.spec.ts"],
  "exec": "ts-node ./index.ts"
}
```

9. Написать скрипт запуска проекта, добавив необходимую конфигурацию в package.json

```
"scripts": {
  "start": "tsc && ts-node dist/app.js"
}
```

10. Дальнейший запуск и разработка с автоматической перекомпиляцией проекта могут вестись по следующей команде в терминале:

```
npm run start
```

Результаты исследования и их обсуждение

TypeScript активно развивается и обладает рядом преимуществ над JavaScript. В следующем разделе будут описаны положительные и отрицательные стороны миграции на TypeScript.

Преимущества:

– Строгая типизация.

– При использовании Node.js можно пользоваться последними возможностями стандартов ES7 и ES8 [7; 8], в то время как на JavaScript максимальный доступный стандарт – ES6.

– Типизированные сигнатуры методов в IDE позволяют видеть полное описание метода и принимаемых параметров.

– При разработке фронтенда с помощью TypeScript можно понимать, результаты каких типов ждать [9].

– Упрощённая автогенерация документации.

– Тестирование и отладка кода проходят легче [9].

– Понятия из мира ООП:

• Interface – позволяет описывать контракты взаимодействующих сущностей, а также методы и свойства, доступные для кода внутри и вне приложения [10].

• Enum – определяет именованные константы.

• Generics – механизм декларации классов и методов для работы со множественными типами данных.

– Async/await

– Await-функции высшего уровня:

Код

```
let f = async () => {
  return new Promise((resolve) => {
    setTimeout(() => resolve(`I'm
a cool async response!`), 1000);
  });
};
console.info(await f());
```

Выведет в консоль через 1 секунду сообщение:

I'm a cool async response!

Недостатки:

– Типизировано большинство библиотек, но не все. Могут возникнуть ситуации, когда придётся использовать нетипизированный JavaScript код [11].

– Много времени уходит на описание интерфейсов.

– Некоторые думают, что привнесение TypeScript в проект: увеличит время на изучение проекта новыми разработчиками, усложнит поддержку, увеличит время разработки, оставит проект в хипстерском состоянии (когда через 1 год язык станет неподдерживаемым и ненужным), предотвратит вход в проект JavaScript-разработчиков, сделает невозможным использование кодовой базы не на TypeScript и усложнит изменение функциональности приложения в отдалённом будущем.

Выводы

Мигрируя на TypeScript, удалось снизить количество ошибок на 15% при разработке и сократить время написания кода

на примерно 10%: облегчилась отладка, а благодаря инструментам автоматической генерации документации отпала необходимость в ручном комментировании кода. Знакомство с проектом для новых разработчиков при этом стало на 30% быстрее. Для опытных разработчиков этот показатель достиг 50%.

Список литературы

1. Google Trends: Typescript. [Electronic resource]. URL: <https://trends.google.com/trends/explore?date=today%205-y&q=%2Fm%2F0n50hvx> (date of access: 12.07.2020).

2. The RedMonk Programming Language Rankings: January 2020. [Electronic resource]. URL: <https://redmonk.com/sogady/2020/02/28/language-rankings-1-20> (date of access: 12.07.2020).

3. ECMAScript 2016+ compatibility table. [Electronic resource]. URL: <http://kangax.github.io/compat-table/es2016plus> (date of access: 12.07.2020).

4. Search: @types – npm. [Electronic resource]. URL: <https://www.npmjs.com/search?q=%40types&ranking=popularity> (date of access: 12.07.2020).

5. Ликбез по типизации в языках программирования. [Электронный ресурс]. URL: <https://habr.com/ru/post/161205> (дата обращения: 12.07.2020).

6. The State of the Octoverse celebrates a year of building across teams, time zones, and millions of merged pull requests. [Electronic resource]. URL: <https://octoverse.github.com> (date of access: 12.07.2020).

7. Turn your ES5 code into readable ES6. Lebab does the opposite of what Babel does. [Electronic resource]. URL: <https://github.com/lebab/lebab> (date of access: 12.07.2020).

8. What is TypeScript and why would I use it in place of JavaScript? [Electronic resource]. URL: <https://stackoverflow.com/questions/12694530/what-is-typescript-and-why-would-i-use-it-in-place-of-javascript/35048303#35048303> (date of access: 12.07.2020).

9. TypeScript Roadmap: January – June 2020. [Electronic resource]. URL: <https://github.com/microsoft/TypeScript/issues/36948> (date of access: 12.07.2020).

10. Why TypeScript is the best way to write Front-end in 2019–2020. [Electronic resource]. URL: <https://medium.com/@jtomaszewski/why-typescript-is-the-best-way-to-write-front-end-in-2019-feb855f9b164> (date of access: 12.07.2020).

11. The Benefits of Migrating from JavaScript to TypeScript [Electronic resource]. URL: <https://www.appdynamics.com/blog/engineering/the-benefits-of-migrating-from-javascript-to-typescript> (date of access: 12.07.2020).

УДК 004.048

ОБРАБОТКА СТАТИСТИЧЕСКИХ ДАННЫХ МЕТОДОМ ГЛУБОКОГО ОБУЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ МОДУЛЯ KERAS

Ильичев В.Ю., Юрик Е.А.

Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», Калуга, e-mail: patrol8@yandex.ru

В статье описана разработанная авторами методика работы с нейронными сетями, предназначенными для классификации результатов статистических исследований. Методика реализована с использованием нейросетевого модуля Keras и ряда других модулей языка Python. Работа со статистическими данными с помощью библиотек Python организована путём последовательного осуществления ряда этапов: подбора и подготовки данных, выбор типа и топографии создаваемой сети, способа её обучения (реализовано так называемое глубокое обучение), оценки точности получаемых результатов обучения. Способы осуществления каждого из описанных этапов в настоящее время не являются достаточно отработанными, т.к. язык Python и методы, реализуемые с его помощью, находятся в постоянном непрерывном развитии. Так как Python, как и его библиотеки, являются свободно распространяемым программным обеспечением, то это упрощает и интенсифицирует развитие по всему миру этого универсального языка и специализированных модулей для него. Но, с другой стороны, каждый разработчик программ на этом языке реализует свои идеи по-разному. Из всех используемых в данной работе модулей следует отметить библиотеки работы с массивами Numpy, с графическим выводом результатов Matplotlib, выводом топографии сети Graphviz и Pydot. Хотя в программе применяется много модулей, но команды, реализуемые с их помощью, являются несложными для освоения даже неспециалистами в программировании. Созданный программный продукт был обучен на выборке экспериментальных клинических исследований показателей здоровья большой группы пациентов, результатом предсказаний должно являться наличие заболеваний сердца у тестируемых пациентов. Расчёты проведены на организованной средствами модуля Keras нейронной сети, произведена оценка точности прогнозирования в зависимости от количества циклов (эпох) обучения. В результате найдено число эпох, при котором достигается максимальное качество обучения созданной сети. В заключение сформулированы выводы по результатам представленной работы и даны рекомендации для её дальнейшего использования.

Ключевые слова: нейросетевое программирование, глубокое обучение, язык Python, модуль Keras, прогнозирование событий

PROCESSING OF STATISTICS BY DEEP LEARNING USING KERAS MODULE

Ilichev V.Y., Yurik E.A.

Kaluga Branch of Bauman Moscow State Technical University, Kaluga, e-mail: patrol8@yandex.ru

The article describes the methodology developed by the authors to work with neural networks designed to classify the results of statistical studies. The technique is implemented using the Keras neural network module and a number of other Python language modules. Work with statistical data using Python libraries is organized by sequentially implementing a number of stages: selecting and preparing data, choosing the type and topography of the network being created, how to train it (the so-called «deep learning» is implemented), assessing the accuracy of the resulting training results. The methods for carrying out each of the described steps are not currently sufficiently developed, since the Python language and the methods implemented by it are in constant continuous development. Since Python, like its libraries, are freely available software, this simplifies and attenuates the development of this universal language and its specialized modules around the world. But, on the other hand, each developer of programs in this language implements their ideas in different ways. Of all the modules used in this work, note the libraries for working with arrays Numpy, with the graphical output of results Matplotlib, and the topography of the network Graphviz and Pydot. Although the program uses many modules, the commands implemented using them are easy to master even by specialists in programming. The software product created was trained on a sample of experimental clinical studies of health indicators of a large group of patients, the result of the predictions should be the presence of heart diseases in the patients tested. Calculations were carried out on a neural network organized by the Keras module, and prediction accuracy was estimated depending on the number of learning cycles (eras). As a result, we found the number of eras at which the maximum quality of training of the created network is achieved. Conclusions were drawn on the results of the submitted work and recommendations were made for its further use.

Keywords: neural network programming, deep learning, Python language, Keras module, event prediction

Получение новых знаний практически в любой отрасли науки невозможно представить без сбора и обработки статистических данных.

Обработка данных наблюдений или экспериментов осуществляется с целью вывода зависимостей, позволяющих описать исследуемые процессы (эмпирико-теоретический метод исследований). В насто-

ящее время анализ данных превратился в отдельную науку на стыке математики и информатики.

Известно множество способов анализа данных: регрессионный, канонический, кластерный, дискриминантный, корреляционный, факторный и множество других [1; 2]. При выполнении каждого типа анализа преследуются разные цели.

В последнее время широкое распространение получил нейросетевой анализ, основанный на имитации работы нервной системы человека. Для осуществления данного метода обработки данных необходимо создать так называемую нейронную сеть, а затем обучить её на исторических данных. Модель, полученная в результате обучения, при правильной организации этого процесса обладает свойствами предсказания событий с определённой высокой долей вероятности.

Целью данной работы является разработка программы, реализующей один из вариантов нейронной сети, позволяющей на раннем этапе прогнозировать наличие либо отсутствие заболеваний сердца у пациентов по известным параметрам их исследования в центре здоровья.

При этом также ставится задача демонстрации порядка работы с модулем нейросетевого программирования Keras, реализующим метод «глубокого обучения» (deep learning). Также используются другие библиотеки языка Python. Для «тренировки» сети набор экспериментальных данных обрабатывается специальным образом.

После обучения нейронной сети необходимо решить задачу оценки качества созданной на её основе модели исследуемого процесса возникновения заболеваний.

Материалы и методы исследования

Часто традиционными методами не удаётся описать отклик системы на воздействие входных факторов, но с этой задачей во многих случаях может справиться обучаемая нейронная сеть. Способ построения нейронной сети напоминает организацию нервной системы человека из нейронов [3].

Рассмотрим, например, использование нейронной сети при проверке предполагаемого диагноза больного на основании анализа показателей физиологических показателей подаются на нейроны первого слоя сети, обрабатываются, и уже в усиленном или ослабленном виде передаются на следующий слой. Таких слоёв необходимо создать несколько (входной, выходной и промежуточные). При обучении сети по факторам и известным результатам их воздействия на систему формируются коэффициенты усиления или ослабления сигналов межнейронными связями. Если система должна выдать значение одного результата (отсутствия или наличия заболевания), то последний слой такой сети состоит только из одного нейрона, на котором и вырабатывается отклик – ответ системы, необходимый исследователю.

При создании нейронной сети ключевыми моментами, от которых зависит качество дальнейшего прогнозирования, являются её архитектура (количество слоёв и нейронов в каждом слое, их взаимные связи или их отсутствие), набор факторов для обучения и формирования прогноза, а также способ обучения сети.

Особенности любой научной работы в первую очередь определяются составом исходных данных для исследования. В нашем случае были использованы открытые базы данных по болезням сердца в некоторых странах, хорошо подходящие для целей машинного обучения [4]. В них приводятся только данные анкетирования и медицинских замеров, без указания личных данных пациентов.

Эти данные собраны по следующим медицинским учреждениям:

1. Кливлендская клиника.
2. Венгерский институт кардиологии, Будапешт.
3. Ветеранский медицинский центр, Лонг-Бич, Калифорния.
4. Университетская больница, Цюрих, Швейцария.

По каждому учреждению для каждого пациента собрано по 74 параметра, из которых для диагностики болезней сердца имеют смысл только 14 (причём многие из них выражены условными обозначениями, отображающими показатель не количественно, а качественно), а именно: возраст, пол, тип грудной боли, кровяное давление в покое, холестерин в крови, сахар в крови, результаты кардиографии, частота сердечного ритма, стенокардия при физической нагрузке, наклон пика ST-сегмента на кардиограмме при нагрузке, форма пика ST-сегмента, количество крупных сосудов (0–3), окрашенных при флюорографии, тип сердечного ритма. Результат (14-й параметр) выражен бинарным образом: 0 – риск сердечных заболеваний незначительный, 1 – риск сердечных заболеваний высокий (сужение диаметров крупных сосудов превышает 50%).

Из 4 баз данных для исследования выбрана наибольшая по количеству записей (объёму) – база Венгерского института кардиологии. Для обучения нейронной сети не использовались столбцы 10–13, в которых практически отсутствуют данные (они имеются лишь по некоторым пациентам). Кроме того, исключено небольшое количество строк, в которых часть записей в столбцах 1–9 отсутствует. В результате такой предварительной обработки базы в ней осталась 261 строка. Для обучения базы использовались столбцы 1–9 с факторами и столбец 14 с результатами.

Для загрузки файла данных и перемешивания строк базы данных случайным образом (для исключения ложного обучения) использована библиотека Numpy [5]. Затем для реализации алгоритма работы с нейронной сетью использован современный высокоуровневый модуль Keras, обладающий широкими возможностями, написанный на языке Python и являющийся надстройкой для модуля более низкого уровня организации Tensorflow (так называемого фреймворка).

Рассмотрим порядок создания нейронной сети и работы с ней с помощью модуля Keras.

1. Указываем столбцы факторов – входных параметров X (с 1 по 9) и столбец результата – целевой переменной Y (14-й).

2. Выбираем модель нейронной сети Sequential (прямого распространения СПР) – с последовательным соединением нейронных слоев, в которой информация проходит только в прямом направлении. Как показала практика, такая сеть хорошо подходит для задач классификации и идентификации событий, подобных рассматриваемой.

3. Определяем слои нейронной сети. Входной слой имеет размерность 9 согласно количеству действующих факторов. Число нейронов выбрано достаточно условно равным 12 (в результате оптимизации сети для достижения наилучших результатов предсказаний). Далее добавлено ещё 3 слоя, состоящих соответственно из 15, 8 и 10 нейронов.

Созданная топология сети отображена средствами графических библиотек Graphviz и Pydot для Python [5] и показана на рис. 1.

На данном рисунке наглядно видно количество слоёв, нейронов в каждом слое, способ взаимного соединения слоёв, тип каждого слоя.

Тип активизации сетей выбран Rectified linear units (ReLU), считающийся в настоящее время наиболее подходящим для сетей с глубоким обучением. Выходной слой имеет один нейрон, т.к. целью сети является идентификация только одного параметра, для активизации этого слоя использован метод активизации с помощью сигмоидной функции, хорошо фильтрующей шум.

4. Для возможности использования сети нужно скомпилировать. Для этого использован популярный метод оптимизации Adam (полное название «метод адаптивной инерции») [6].

5. Выбираем метод обучения сети fit, включающий в себя разбиение данных на тренировочные (в данном случае матрица X

из столбцов 1–9) и валидационные (матрица-вектор Y из столбца 14). Также задаётся количество проходов (эпох) обучения и количество строк для одного шага обучения. В дальнейшем будет рассмотрено влияние количества эпох на качество обучения созданной сети.

6. В конце программы задаём функцию evaluate модуля Keras, оценивающую точность прогноза по созданной модели и вывода значения точности на экран.

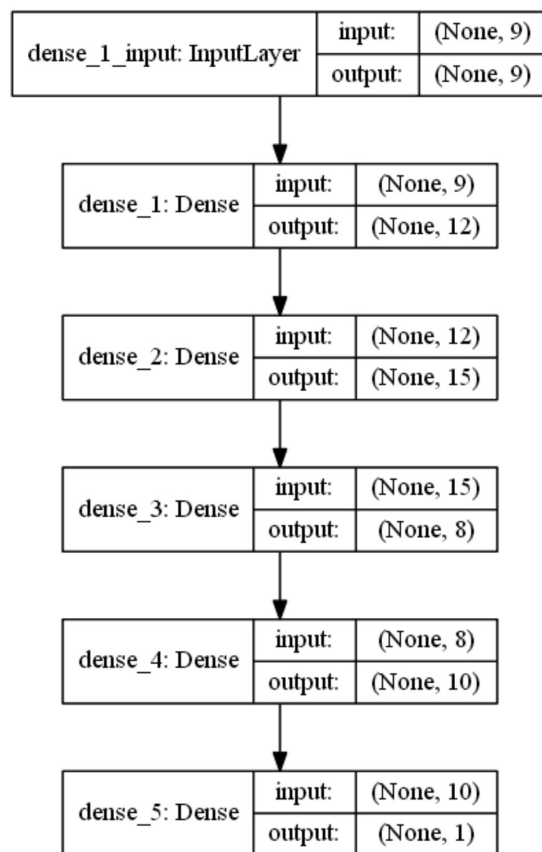


Рис. 1. Топография созданной нейронной сети

Результаты исследования и их обсуждение

Одной из целей исследований была оценка влияния количества эпох обучения на точность достигаемого результата. Нейросеть обучалась последовательно по 100 эпохам 500 раз (итого по 50000 эпохам), после каждого обучения оценивалась точность прогнозирования и заносилась в массив. Созданный массив результатов оценки точности в процентах в зависимости от количества эпох выведен в графическом виде с помощью библиотеки Matplotlib на рис. 2.



Рис. 2. Зависимость качества прогнозирования нейросетью от количества эпох обучения

Из рис. 2 можно сделать вывод, что качество обучения сети нестабильно в зависимости от количества эпох, хотя явно недообученная система (с количеством эпох до 1000) даёт более низкую точность прогноза. Нестабильность результата обучения можно объяснить снижением и повышением корреляционной связи между количеством нейронов в слоях и типом использованных слоёв и количеством данных для обучения при их варьировании. Максимальная точность прогноза наблюдается вблизи количества эпох 10000 и 20000. С помощью функции `max` библиотеки `Numpy` найдено максимально достигнутое качество прогноза 95% при количестве эпох обучения 22100. Поэтому такое количество эпох и рекомендуется использовать для обучения созданной нейросети.

С использованием описанных методов организации, обучения и оценивания нейронных сетей можно создавать сети различной топографии и подбирать из них подходящие наилучшим образом для решения поставленных задач. В случае необходимости кроме описанных библиотек в программе на Python можно использовать и дополнительные, специализированные библиотеки [7].

В данной работе рассмотрена методика работы с нейронными сетями с использованием библиотек языка Python. К настоящему времени эта задача является нетривиальной, так как язык Python и дополнительные модули для работы с ним

динамично развиваются, появляются новые команды, изменяется синтаксис. Это создает определённые сложности в освоении этого универсального языка, но одновременно сильно расширяет его возможности.

В частности, для обеспечения гибкости работы с нейросетями язык Python с подключением модуля глубокого обучения Keras и прочих дополнительных модулей обладает следующими неоспоримыми преимуществами, одновременного сочетания которых нет в других программных продуктах [8]:

- модульность, компактность, расширяемость;
- лёгкость освоения неспециалистами;
- множество инструментов работы с нейронными сетями;
- реализация принципов глубокого обучения, получившего известность в последнее время и способного создавать прогнозы для более широкого круга явлений, чем традиционные способы обучения (такие как ненаправленная графическая модель);
- возможность эффективного использования графических процессоров (видеокарт) для вычислений;
- широкие возможности визуализации результатов с использованием модуля Matplotlib.

Данная статья иллюстрирует лишь частный случай – пример работы с модулем глубокого обучения Keras для распознавания заболеваний. Кроме этой цели, модуль Keras всё чаще начинают использовать при

машинном переводе текстов, распознавании условных графических обозначений, образов и даже людей по фотографиям [9].

Заключение

Из выполненной и описанной работы можно сформулировать следующие особенности работы с модулем Keras, которые необходимо учитывать для дальнейшей оптимизации работы с ним:

– важно правильно определить состав исходных факторов для обучения и прогноза, действительно оказывающих влияние на результат;

– необходимо выбрать тип и топографию нейронной сети, наиболее легко и быстро обучаемой и дающей наиболее точный прогноз;

– целесообразно выбрать наиболее эффективный метод компиляции сети из всех существующих, которые со временем совершенствуются;

– обязательно нужно подобрать количество эпох обучения, причём в этом случае важно не получить недообучения или переобучения сети, т.к. в этих случаях точность прогнозирования с её помощью уменьшается. Оптимальное количество эпох можно подобрать методом, описанным в данной статье.

Принципы использования модуля глубокого обучения Keras, изложенные в статье, можно применить для создания нейросетей с целью классификации многих других событий и объектов [10].

Список литературы

1. Москвитин А.А., Сокиев Т.М. Особенности современных методов интеллектуального анализа данных // Современные методы интеллектуального анализа данных в экономических, гуманитарных и естественнонаучных исследованиях: материалы международной научно-практической конференции. Пятигорск, 2016. С. 11–18.
2. Низаметдинов Ш.У., Румянцев В.П. Анализ данных: учебное пособие для студентов высших учебных заведений. М., 2012. 285 с.
3. Голоскоков К.П. Применение нейронных сетей в задачах прогнозирования и проблемы идентификации моделей прогнозирования на нейронных сетях // Современные проблемы прикладной информатики: сборник научных трудов. 2006. С. 116–120.
4. Machine Learning Repository. Center for Machine Learning and Intelligent Systems. [Electronic resource]. URL: <https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease> (date of access: 03.07.2020).
5. Сысоева М.В., Сысоев И.В. Программирование для «нормальных» с нуля на языке Python. Учебник. В двух частях. Часть 1. М.: ООО «МАКС Пресс», 2018. 176 с.
6. Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014). [Electronic resource]. URL: <https://arxiv.org/abs/1412.6980v8> (date of access: 03.07.2020).
7. Зубов М.В., Пустыгин А.Н. Анализ путей исполнения программ по исходному коду с использованием универсальных промежуточных представлений // Известия Юго-Западного государственного университета. 2015. № 4 (61). С. 12–19.
8. Тур А.И., Кокоулин А.Н., Южаков А.А., Лукичев А.Н. Подготовка системы распознавания объектов на базе Tensorflow и Keras // Международная конференция по мягким вычислениям и измерениям. 2018. Т. 1. С. 651–653.
9. Keras: The Python Deep Learning library. Keras documentation. [Electronic resource]. URL: <https://keras.io/#why-thisname-keras> (date of access: 03.07.2020).
10. Хайбрахманов С.А. Основы научных расчётов на языке программирования Python: учебное пособие. Челябинск: Изд. Челябинского государственного университета, 2019. 96 с.

УДК 004.43

ПРИМЕНЕНИЕ ЯЗЫКА PYTHON ДЛЯ РАЗРАБОТКИ КОНВЕРТЕРА ВАЛЮТ

Ошурков А.Ю., Бужинская Н.В.

*Нижнетагильский государственный социально-педагогический институт (филиал)
ФГАОУ ВО «Российский государственный профессионально-педагогический университет»,
Нижний Тагил, e-mail: nadezhda_v_a@mail.ru*

В связи с усложнением функций программного обеспечения наблюдается интерес к различным языкам программирования. В настоящее время одним из самых популярных языков программирования является Python. Python – это высокоуровневый язык программирования, который может использоваться для решения задач математики, создания программ для распознавания образов, построений нейронных сетей и др. В статье демонстрируется применение данного языка для создания конвертера валют. С помощью данной программы можно ознакомиться с курсами трех мировых валют, производить конвертирования из RUB в USD, EUR и CNY. В статье расписаны этапы создания конвертера на языке Python. На этапе проектирования определяются требования к данной программе, определяется набор библиотек и модулей, которые будут применяться в процессе работы. Разработка программы включает несколько этапов. Вначале создается интерфейс посредством PyQt5, затем данный интерфейс компилируется в файл с расширением .py. На втором этапе оформляется код для каждой функции, которые будут вызваны при нажатии на соответствующие кнопки. Актуальный на сегодняшний день курс валют можно получить из Интернета. Для более удобной работы создана база данных, которая хранит информацию о результатах конвертирования. Заключительным этапом работы является тестирование, в процессе которого были выявлены и исправлены ошибки.

Ключевые слова: программирование, программный продукт, конвертер, база данных, валюта, функция

USING PYTHON TO DEVELOP A CURRENCY CONVERTER

Oshurkov A.Y., Buzhinskaya N.V.

*Nizhny Tagil State Socio-Pedagogical Institute (branch) of Federal State Autonomous educational
institution «Russian state vocational pedagogical University», Nizhny Tagil, e-mail: nadezhda_v_a@mail.ru*

Due to the complexity of software functions, there is an interest in various programming languages. Currently, one of the most popular programming languages is Python. Python is a high-level programming language that can be used for solving math problems, creating programs for pattern recognition, building neural networks, and so on. This article demonstrates how to use this language to create a currency Converter. With this program, you can familiarize yourself with the exchange rates of three world currencies, make conversions from RUB to USD, EUR and CNY. This article describes the steps for creating a Python Converter. At the design stage, the requirements for this program are determined, and the set of libraries and modules that will be used in the process is determined. The program development includes several stages. First, an interface is created using PyQt5, then this interface is recompiled to a file with the .py extension. At the second stage, the code is generated for each function that will be called when the corresponding buttons are clicked. The current exchange rate can be obtained from the Internet. For more convenient operation, a database has been created that stores information about the conversion results. The final stage of the work is testing, during which errors were identified and corrected.

Keywords: programming, software product, Converter, database, currency, function

На современном этапе развития общества большой популярностью среди учащихся пользуются различные языки программирования, а также программные продукты, разработанные с их помощью. Одним из самых популярных языков программирования является Python. Python представляет собой высокоуровневый язык программирования, который поддерживает такие парадигмы, как императивное программирование, объектно-ориентированное, функциональное и др. [1]. Целью данного исследования является описание возможностей языка программирования Python как средства разработки программы с пользовательским интерфейсом для конвертации валют с возможностью сохранения истории действий в базе данных SQLite. Выбор данного языка для разработ-

ки нашего программного продукта объясняется не только его популярностью. В настоящее время Python активно развивается, появляются его новые версии, улучшаются и добавляются библиотеки и модули [2].

Материалы и методы исследования

В данном исследовании описан процесс реализации программы «Конвертер валют» на языке Python. Тема для разработки была выбрана таким образом, чтобы посмотреть возможности Python с разных точек зрения – работа с базой данных, с информацией в Интернете, создание дизайна.

Процесс работы включал несколько этапов:

– проектирование программного продукта, во время которого были обозначены основные направления работы;

- разработка дизайна конвертера;
- оформление внутренней логики программы;
- обработка событий;
- создание базы данных для отображения курса валют;
- подготовка файла с расширением .exe;
- оформление необходимой документации;
- тестирование программы.

Результаты исследования и их обсуждение

На этапе проектирования были определены основные требования к программе (рис. 1) [3].

Пользователь вводит необходимую сумму. Если он хочет увидеть курс, актуальный на сегодняшний день, он нажимает кнопку «Получить курс», иначе он выбирает команду «Конвертировать» и получает инте-

ресующие его значения. Далее посмотрим процесс разработки программы на языке *Python*.

Структура программы представлена на рис. 2.

В папке *Images* хранится иконка приложения. Сама база данных *db.currency_convert.db* хранится в общей папке с проектом, *mainWindow_converter.ui* интерфейс программы в *XVIL*, который конвертирован в файл *mainWindow_converter.py*, файл *parse_currency.py* содержит код для парсинга с официального сайта Google, чтобы собирать актуальную информацию о курсе. В *main.py* прописана основная логика сайта.

Запустим файл *main.py*. Появится окно программы. При вводе необходимой суммы в рублях программа переводит ее в другую денежную единицу по курсу на сегодняшний день (рис. 3).

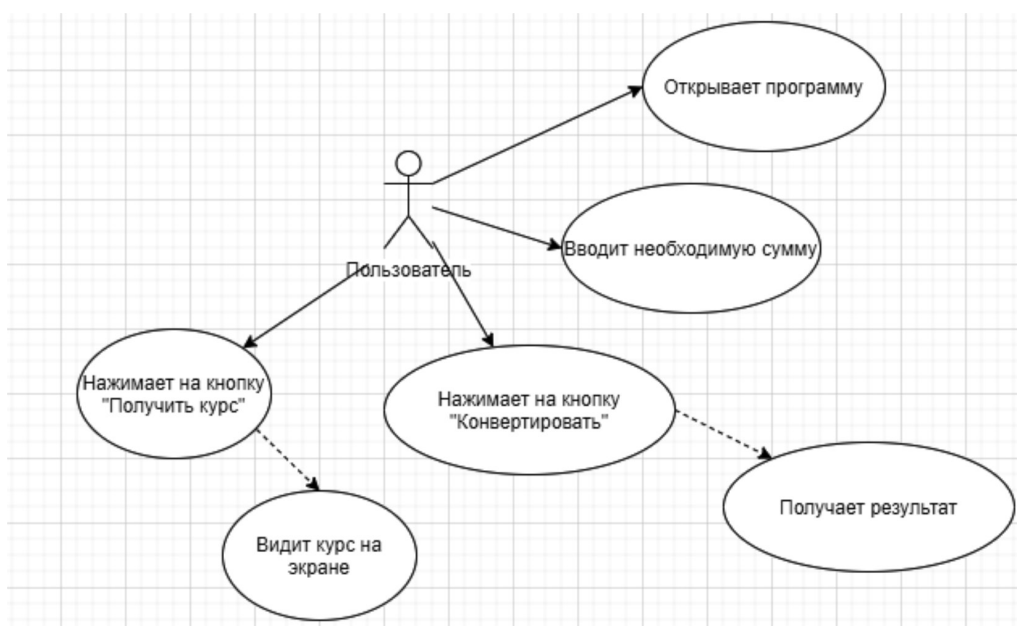


Рис. 1. Диаграмма прецедентов

Images	25.06.2020 14:10	Папка с файлами	
db.currency_convert	25.06.2020 17:31	Data Base File	12 КБ
main	25.06.2020 17:32	Python File	4 КБ
mainWindow_converter	25.06.2020 15:40	Python File	5 КБ
mainWindow_converter.ui	25.06.2020 15:34	Файл "UI"	5 КБ
parse_currency	25.06.2020 16:15	Python File	1 КБ

Рис. 2. Структура программы

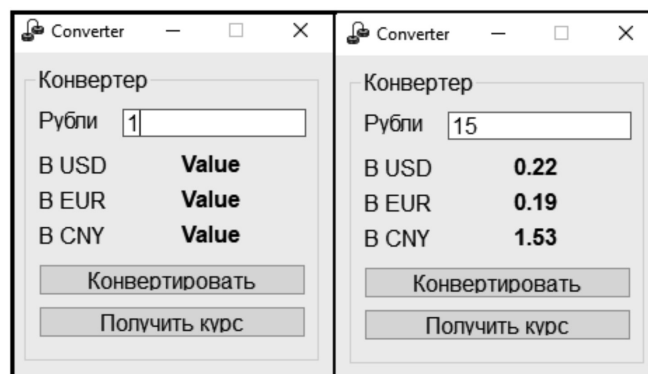


Рис. 3. Интерфейс программы

```
from mainWindow_converter import window_converter
from parse_currency import check_current
from PyQt5 import QtCore, QtGui, QtWidgets
import sys
from typing import *
from datetime import datetime
import sqlite3 as sql
from sqlite3 import Error
```

Рис. 4. Список подключаемых библиотек и модулей для реализации программы

Список используемых библиотек и модулей представлен на рис. 4.

Интерфейс программы создавался в *PyQt5*, с помощью *Qt Designer*. Для создания интерфейса использовались такие элементы, как *groupBox*, семь виджетов *label*, *lineEdit*, два виджета *pushButton*.

SQLite предназначена для работы с базой данных, которая отвечает за создание и поддержку базы данных для данного приложения. Библиотека *sys* предоставляет программисту набор функций, которые предоставляют информацию о том, как работает интерпретатор Python, взаимодействуя с операционной системой. Библиотека *typing* предназначена для создания аннотаций. Модель *datetime* предоставляет классы для обработки времени и даты разными способами. В *parse_currency* находится функция *check_currency* для получения информации с сайта Google.

На рисунке ниже изображена база данных *db.currency_convert.db*, которая хранит информацию о дате и времени использования конвертера, сумму в рублях для конвертирования, курс доллара, евро и конвертированные значения по курсам (рис. 5).

Далее осуществлялось подключение к базе данных (рис. 6). Если подключение

завершилось с ошибкой, программа сигнализирует об этом с помощью исключения *Try Except*.

Затем выполняется запрос, согласно которому о переменной *currency* присваивается кортеж последней записи в базе данных. Переменные получают значения актуального на сегодняшний день курса валют (рис. 7).

На рис. 8 продемонстрировано, что функция *toFixed* на входе получает число типа *float* (переменная *numObj*) и количество знаков после запятой (переменная *digits*). На выходе получаем число типа *str* с количеством знаков после запятой, которое было указано.

В переменную *data* присваивается кортеж с данными, которые нужно внести в базу данных (рис. 9).

На рис. 10 продемонстрирован пример запроса на добавление данных из переменной *data* в базу данных.

Затем функция на вход получает ссылку на источник с актуальным на сегодняшний день курсом, затем отправляет запрос, получает HTML-файл, обрабатывает его и возвращает курс валюты. Реализация функции на языке Python представлена на рис. 11.

	id	time_use	input_currency	USD	EUR	CNY	output_USD	output_EUR	output_CNY
1	1	25/06/20 17:11:27	150	69.45	77.74	9.81	2.16	1.93	15.29
2	2	25/06/20 17:11:31	150000	69.45	77.74	9.81	2159.83	1929.51	15290.52
3	3	25/06/20 17:11:35	250000	69.45	77.74	9.81	3599.71	3215.85	25484.2
4	4	25/06/20 17:31:38	1	69.45	77.74	9.81	0.01	0.01	0.1
5	5	25/06/20 17:31:47	10	69.45	77.74	9.81	0.14	0.13	1.02
6	6	25/06/20 23:51:59	65	69.14	77.55	9.77	0.94	0.84	6.65
7	7	25/06/20 23:52:04	650	69.14	77.55	9.77	9.4	8.38	66.53
8	8	25/06/20 23:52:09	1	69.14	77.55	9.77	0.01	0.01	0.1
9	9	25/06/20 23:52:17	4000	69.14	77.55	9.77	57.85	51.58	409.42

Рис. 5. База данных db.currency_convert.db

```
def create_connection(database: str) -> str:
    try:
        database = sql.connect(database)
        print("Connection database")
        return database
    except Error:
        print(f"Error '{Error}' connection")
```

Рис. 6. Подключение к базе данных

```
[currencys] = sqlCursor.execute("select id, USD, EUR, CNY "
                                "from history_converter_currency "
                                "order by id desc limit 1")
value_USD = currencys[-3]
value_EUR = currencys[-2]
value_CHY = currencys[-1]
```

Рис. 7. Запрос к базе данных

```
def toFixed(numObj: float, digits: int = 0) -> str:
    return f"{numObj:.{digits}f}"
```

Рис. 8. Реализация функции toFixed

```
data = [(str(datetime.strftime(datetime.now(), "%d/%m/%y %H:%M:%S")),
        input_value_RUS, value_USD,
        value_EUR, value_CHY,
        output_value_USD, output_value_EUR,
        output_value_CNY)]
```

Рис. 9. Работа с переменной data

```
sqlCursor.executemany("INSERT INTO history_converter_currency(time_use,"
                      "input_currency,"
                      "USD,"
                      "EUR,"
                      "CNY,"
                      "output_USD,"
                      "output_EUR,"
                      "output_CNY)"
                      "VALUES (?, ?, ?, ?, ?, ?, ?, ?)", data)
database.commit()
```

Рис. 10. Выполнение запроса


```
def check_current(url):
    response = rec.get(url=url, headers=HEADERS)
    HTML = response.content

    convert_html = BeautifulSoup(HTML, 'lxml')
    head = convert_html.find_all('span', {'class': 'DFlfde',
                                          'class': 'SwHCTb',
                                          'data-precision': 2})

    value = head[0].text
    value = value.replace(',', '.')
    return value
```

Рис. 11. Реализация функции

```
for i in range(len(input_value_RUS.lower())):
    if not input_value_RUS.isdigit():
        window.output_USD_2.setText("Error 01")
        window.output_EUR_2.setText("Error 01")
        window.output_CNY_2.setText("Error 01")
    return
```

Рис. 12. Проверка введенных пользователем данных

Таблица 1

Результаты тестирования программы. Тест-кейс № 1. Корректный

Номер	1
Заголовок	Перевод нужной суммы в рублях в другую денежную единицу
Предусловие	Программа запущена. Открыто главное окно программы. На ПК есть доступ в Интернет
Шаг	Ожидаемый результат
Ввести нужную сумму в поле «Рубли». Вводить только цифры	Ввод успешен
Нажать на кнопку «Конвертировать»	Программа выдает необходимый результат
Нажать на кнопку «Получить курс»	Программа выдает курс, актуальный на данном этапе

Таблица 2

Результаты тестирования программы. Тест-кейс № 2. Некорректный

Номер	2
Заголовок	Перевод нужной суммы в рублях в другую денежную единицу
Предусловие	Программа запущена. Открыто главное окно программы. На ПК есть доступ в Интернет
Шаг	Ожидаемый результат
Ввести нужную сумму в поле «Рубли». Вводить любую последовательность символов	Программа выдает сообщение об ошибке. Кнопки «Конвертировать» и «Получить курс» не активны

Кроме того, для данной функции была написана проверка, чтобы пользователь вводил только числовые значения. Если пользователь введет символ, отличный от числового значения, программа выдает ошибку. Реализация функции представлена на рис. 12.

На заключительном этапе работы было оформлено «Руководство пользователя»

и проведено тестирование программы. Тестирование направлено на решение двух важнейших задач. С одной стороны, разработчику нужно показать, что программа удовлетворяет заданным требованиям к ней, а с другой – исправить все ошибки, которые были обнаружены во время внедрения программы [4; 5]. Результаты тестирования программы представлены в табл. 1–2 [6].

Заключение

Данная программа может использоваться в качестве демонстрационной для объяснения обучаемым основных возможностей языка Python, а именно:

- установка различных библиотек через командную строку;
- создание базы данных и запросов к ней с помощью SQLite;
- получение нужной информации с различных сайтов;
- разработка интерфейса с помощью PyQt;
- перекомпилирование файла с расширением .ui в расширение .py.

В дальнейшем планируется добавить в данную программу визуальное отобра-

жение курсов валют с помощью библиотеки Matplotlib.

Список литературы

1. Свейгарт Эл. Учим Python, делая крутые игры. М.: Бомбора, 2018. 416 с.
2. Прохоренок Н., Дронов В. Python 3 и PyQt5. Разработка приложений. СПб.: БХВ-Петербург, 2019. 833 с.
3. Каюмова А.В. Визуальное моделирование систем в StarUML: учебное пособие. Казань: Казанский федеральный университет, 2013. 104 с.
4. Липаев В.В. Программная инженерия сложных заказных программных продуктов: учебное пособие. М.: МАКС ПРЕСС, 2014. 312 с.
5. Карпович Е.Е. Методы тестирования и отладки программного обеспечения. М.: Изд. Дом НИТУ «МИСиС», 2020. 136 с.
6. Правильно пишем тест-кейсы. Памятка начинающему специалисту по тестированию. [Электронный ресурс]. URL: <https://victorz.ru/202001101079> (дата обращения: 07.07.2020).

УДК 004.4'23

**АНАЛИЗ ДОСТОИНСТВ И НЕДОСТАТКОВ КВАНТОВЫХ ЭМУЛЯТОРОВ
НА ПРИМЕРЕ ВЗАИМОДЕЙСТВИЯ С ПОЛЬЗОВАТЕЛЕМ****Третьяк М.А., Щекатурин А.Е., Пилипенко И.А., Кравченко В.О., Черкесова Л.В.***ФГБОУ ВПО «Донской государственный технический университет», Ростов-на-Дону,
e-mail: reception@donstu.ru*

В данной статье были рассмотрены наиболее популярные квантовые эмуляторы, использующиеся для квантовых вычислений и позволяющие эмулировать процесс работы квантового компьютера. Была проведена работа по изучению квантовых эмуляторов, выявлению и описанию их индивидуальных особенностей, составлению рекомендаций для более удобного начала работы с ними, а также описанию их достоинств и недостатков. Задача эмуляторов состоит в том, чтобы предоставлять пользователю возможность отладки и запуска программ или схем, предназначенных для работы на квантовом компьютере. Чтобы решить эту задачу, нужно эмулировать логические квантовые вентили, которые являются базовым элементом квантового компьютера. Они преобразуют входящие состояния кубитов на выходные по определенному закону, и в отличие от многих классических логических вентилях всегда являются обратимыми. В этом принципиальном моменте эмуляторы одинаковы, но есть различия в интерфейсах, полноте документации и т.д. Именно взаимодействию с пользователем будет посвящена эта статья. Ввиду того что работа с реальными квантовыми компьютерами доступна лишь узкому кругу исследователей, эмуляторы просто необходимы для проверки гипотез или алгоритмов.

Ключевые слова: кубит, эмулятор, гейт, квантовый компьютер, квантовое программирование**ANALYSIS OF THE ADVANTAGES AND DISADVANTAGES OF QUANTUM
EMULATORS ON THE EXAMPLE OF INTERACTION WITH THE USER****Tretyak M.A., Schekaturin A.E., Pilipenko I.A., Kravchenko V.O., Cherkesova L.V.***Don State Technical University, Rostov-on-Don, e-mail: reception@donstu.ru*

This article examined the most popular quantum emulators used for quantum computing and allowing to emulate the process of a quantum computer. Work was done to study quantum emulators, identify and describe their individual characteristics, make recommendations for a more convenient start to work with them, as well as describe their advantages and disadvantages. The task of emulators is to provide the user with the ability to debug and run programs or circuits designed to work on a quantum computer. To solve this problem, it is necessary to emulate logical quantum gates, which are the basic element of a quantum computer. They transform the incoming qubit states to the weekend ones according to a certain law, and unlike many classical logic gates, they are always reversible. At this crucial moment, emulators are the same, but there are differences in interfaces, completeness of documentation, etc. This article will be devoted to user interaction. Due to the fact that working with real quantum computers is available only to a narrow circle of researchers, emulators are simply necessary to test hypotheses or algorithms.

Keywords: qubit, emulator, gate, quantum computer, quantum programming

Современного человека компьютерами уже не удивишь. Они стали чем-то обыденным, давно интегрировались в жизнь, с их помощью можно выполнить широкий ряд задач, реализация которых еще в прошлом веке считалась бы чем-то научно-фантастическим. Рост производительности достигается за счет увеличения количества транзисторов на кристалле процессора, что является достаточно дорогостоящим способом [1]. И несмотря на бурное развитие классических компьютеров, уже сейчас существуют ряд задач, которые они не в состоянии решить [2]. Они обычно требуют экспоненциального роста вычислений относительно входных данных. С рядом подобных задач может справиться квантовый компьютер, чья производительность многократно превосходит возможности классических компьютеров. Это связано с тем, что в процессоре цифрового компьютера

все вычисления строятся на основе битов, способных находиться в двух состояниях: 1 либо 0. То есть вся работа сводится к анализу огромного количества данных на предмет соответствия заданным условиям. В основу же квантового компьютера положены кубиты. Кубитом называют квантовые биты, особенностью которых является возможность находиться в состоянии 1 или 0, а также одновременного нахождения в состоянии 1 и 0. Такое состояние называется суперпозицией. Возможности квантового компьютера превосходят возможности классического, так как нет необходимости искать нужный ответ среди множества [3]. В этом случае ответ выбирается из уже имеющихся вариантов с определенной долей вероятности соответствия [4; 5].

Существует такое понятие, как «квантовое превосходство» – это потенциальная способность квантовых вычислительных

устройств решать проблемы, которые классические компьютеры практически не могут, или им потребуется для этого слишком много времени [6]. В той же статье BBC [6] был дан критерий наступления квантового превосходства, по мнению авторов, это – порог вычислительной мощности в 50 кубитов. И хотя квантовое превосходство уже было достигнуто [6], до промышленного производства таких процессоров ещё далеко. Но прикоснуться к будущему IT-сферы возможно уже сегодня, благодаря эмуляторам квантовых компьютеров или облачным платформам, которые взаимодействуют с квантовым компьютером. Различие между эмулятором и облачной платформой заключается в том, что облачная платформа передаёт код программы существующему квантовому компьютеру, после чего передаёт результат обратно пользователю, в то время как эмулятор, пользуясь ресурсами цифрового компьютера, воспроизводит работу идеального квантового компьютера с данной ему программой.

В этой статье мы рассмотрим достоинства и недостатки нескольких эмуляторов, которые, на наш взгляд, лучше всего подходят для обучения квантовому программированию, а именно Quantum Development Kit [7], Quantum Computing Playground [8], jQuantum [9], QuEST [10], Quantum Programming Studio [11], Q-Kit [12]. Однако разнообразие квантовых эмуляторов гораздо больше, и их список представлен на информационном портале Quantiki [13]. В качестве примера работы эмулятора взят алгоритм Шора для числа 15.

1. *Quantum Development Kit* (QDK) был предложен компанией Microsoft в качестве решения, содержащего огромный инструментарий, необходимый для быстрого изучения основ квантового программирования. Этому способствует собственный язык Q#, разработанный компанией Microsoft специально для реализации квантовых вычислений. Microsoft QDK ориентирован в первую очередь на разработчиков, проявляющих интерес к квантовым вычислениям и желающих изучать квантовое программирование, даже если они не являются специалистами по квантовой физике.

Примечательной особенностью Quantum Development Kit является предоставленная разработчикам возможность вести проект в среде разработки, что весьма удобно для создания больших проектов. QDK интегрирован в среду разработки Microsoft Visual Studio (VS), что станет существенным плюсом для тех, кто уже с ней работал и знаком с процессом построения проекта в ней. На данный момент с помощью QDK можно эмулировать

30 логических кубитов для написания программ на цифровом компьютере. QDK поддерживает написание программ с использованием нескольких языков программирования, таких как Python, Q# или C#. При этом разработка программ может происходить как в VS или VS Code, так и в Notebook Jupyter, используя ядро Q#. Однако QDK не имеет графической оболочки, что может вызвать неудобство для людей, не имеющих опыта программирования. Но взамен графической оболочки и основной документации по QDK Microsoft предлагает сборник учебных пособий Quantum Katas [14], в котором представлены элементы квантовых вычислений, а также руководство по программированию на Q# [15; 16], что делает этот вариант наиболее удобным для новичков.

Подходы к разработке квантовых программ для QDK:

1. Разработка с помощью Python. Данный пакет позволяет использовать функции, написанные на языке квантового программирования Q#, программируя при этом на Python.

2. Разработка с помощью Notebook Jupyter. Jupyter предлагает вести разработку квантовых программ из браузера, создавая так называемые заметки, инструкции и другие составляющие квантовой программы. При этом код находится в записной книжке, состоящей из специальных ячеек, запуск которых компилирует код написанный на Q#, и выводит результат операции. Jupyter предоставляет возможность компиляции и моделирования методов, использующихся в Q#.

3. Разработка в Visual Studio. Visual Studio предлагает разработчику вести проект в уже знакомой среде разработки, предоставляя функцию подсветки синтаксиса, для большего удобства написания приложений, а также расширения для Q# VS Code.

На рис. 1 представлен пример кода, написанный на языке программирования Q# в среде разработки Microsoft Visual Studio.

Результат работы квантовой программы выводится в командную строку. На рис. 2 представлен пример такого вывода.

Достоинства Quantum Development Kit:

- QDK позволяет реализовывать сложные квантовые алгоритмы наравне с написанием программ для классических цифровых компьютеров;
- профилирование;
- открытый исходный код;
- есть возможность оценки и контроля вычислительных ресурсов;
- QDK предоставляет возможность отладки программы, что редко наблюдается в других эмуляторах;

```

8  function SetBitValue(reg: Int, bit: Int, value: Bool): Int {
9      if(value) {return reg ||| (1 <<< bit);}
10     else {return reg &&& ~(1 <<< bit);}
11 }
12 operation HelloQ () : Unit {
13     mutable c = 0;
14     using(qubits = Qubit[7]) {
15         H(qubits[0]);
16         H(qubits[1]);
17         H(qubits[2]);X(qubits[6]);
18         CNOT(qubits[0], qubits[5]);
19         CNOT(qubits[0], qubits[4]);
20         X(qubits[4]);
21         CCNOT(qubits[4], qubits[1], qubits[6]);
22         CNOT(qubits[6], qubits[4]);
23         CCNOT(qubits[5], qubits[1], qubits[3]);
24         H(qubits[2]);
25         CNOT(qubits[3], qubits[5]);
26         Controlled Rz([qubits[1]], (PI() / 2.0, qubits[2]));
27         H(qubits[1]);
28         Controlled Rz([qubits[0]], (PI() / 4.0, qubits[2]));
29         Controlled Rz([qubits[0]], (PI() / 2.0, qubits[1]));
30         H(qubits[0]);
31         SWAP(qubits[2], qubits[0]);
32         set c = SetBitValue(c, 2, ResultAsBool(M(qubits[2])));
33         set c = SetBitValue(c, 1, ResultAsBool(M(qubits[1])));
34         set c = SetBitValue(c, 0, ResultAsBool(M(qubits[0])));
35         ResetAll(qubits); }
36     let answer = IntAsString(c);
37     Message(answer);
38 }

```

Рис. 1. Реализация алгоритма Шора в эмуляторе Microsoft QDK

```

4
C:\Users\jfisto\source\repos\QSharpApplication3
cc 6536) завершил работу с кодом 0.

```

Рис. 2. Результат алгоритма Шора в эмуляторе QDK

- возможность написания модульных тестов для тестирования квантовых программ;
- можно вычислить идентификатор кубита, поместив точку останова в код;
- есть возможность работы в облаке Azure с поддержкой до 40 кубитов [17];
- широкий набор примеров реализации алгоритмов на Q# при помощи QDK;
- большой набор библиотек квантового вычисления с открытым исходным кодом.

Недостатки Quantum Development Kit:

- программа, написанная на Q#, не имеет возможности самоанализа;
- нет инструментов для изображения схемы;
- вычисления в облаке Azure не доступны в России;
- возможны ошибки производительности при работе с Jupyter.

2. *Quantum Computing Playground* (QCP), созданный WebGL Chrome в экспериментальных целях, имеет свой собствен-

ный язык сценариев и обладает удобным графическим веб-интерфейсом IDE. При этом Quantum Computing Playground способен имитировать квантовые регистры до 22 кубитов. Для начала работы с эмулятором QCP разработчики предлагают изучить основы синтаксиса языка программирования QScript, а также встроенные команды и математические функции, которые приведены в образцах программ. При этом эмулятор обладает руководством по написанию кода и работы с некоторыми гейтами, а также описанием алгоритмов Шора и Гровера.

Quantum Computing Playground предоставляет возможность использовать отладчик, который позволяет отслеживать поведение написанной программы, а также возможность сохранения кода по URL-адресу. Данный эмулятор вряд ли подойдет человеку без опыта программирования ввиду недостаточного объема обучающего ма-

териала. На рис. 3 представлена реализация алгоритма Шора в веб-эмуляторе QCP.

В эмуляторе реализован графический способ наблюдения за состояниями кубитов в 2D или 3D, где точки описывают суперпозицию кубитов, а их цвет или высота полос описывают амплитуду и фазу заданной суперпозиции. На рис. 4 представлен результат работы квантового алгоритма Шора.

Достоинства Quantum Computing Palyground:

– открытый исходный код [18];

– поддержка отладки кода программы, написанного на собственном языке программирования QScript;

– кроссплатформенность.

Недостатки Quantum Computing Palyground:

– при наличии большого объема кода или использования большого числа искусственных кубитов замедлит работу шейдеров программы;

– высокое потребление оперативной и графической памяти.

```

2 proc FindFactors N
3   x = 0
4   if N < 15
5     Print "Invalid number!"
6     Breakpoint
7   endif
8   width = QMath.getWidth(N)
9   twidth = 2 * width + 3
10  for x; (QMath.gcd(N, x) > 1) || (x < 2); x
11    x = Math.floor(Math.random() * 10000) % N
12  endfor
13  Print "Random seed: " + x
14  for i = 0; i < twidth; i++
15    Hadamard i
16  endfor
17  ExpModN x, N, twidth
18  for i = 0; i < width; i++
19    MeasureBit twidth + i
20  endfor
21  InvQFT 0, twidth
22  for i = 0; i < twidth / 2; i++
23    Swap i, twidth - i - 1
24  endfor
25  for trycnt = 100; trycnt >= 0; trycnt--
26    Measure
27    c = measured_value
28    if c == 0
29      Print "Measured zero, try again."
30      continue
31    endif
32    q = 1 << width
33    Print "Measured " + c + " (" + c / q + ")"
34    tmp = QMath.fracApprox(c, q, width)
35    c = tmp[0];
36    q = tmp[1];
37    Print "fractional approximation is " + c + "/" + q
38    if (q % 2 == 1) && (2 * q < (1 << width))
39      Print "Odd denominator, trying to expand by 2."
40      q *= 2
41    endif
42    if q % 2 == 1
43      Print "Odd period, try again."
44      continue
45    endif
46    Print "Possible period is " + q
47    a = QMath.ipow(x, q / 2) + 1 % N
48    b = QMath.ipow(x, q / 2) - 1 % N
49    a = QMath.gcd(N, a)
50    b = QMath.gcd(N, b)
51    if a > b
52      factor = a
53    else
54      factor = b
55    endif
56    if (factor < N) && (factor > 1)
57      Display "<h2>Success: " + factor + " * " + N / factor
58      Breakpoint
59    else
60      Print "Unable to determine factors, try again."
61      continue
62    endif
63  endfor
64 endproc
65 VectorSize 16
66 FindFactors 15

```

Рис. 3. Реализация алгоритма Шора в Quantum Computing Playground



Рис. 4. Результат алгоритма Шора в эмуляторе *Quantum Computing Palyground*

3. *JQuantum* – это специализированная программа, взаимодействующая с виртуальной Java-машиной, предназначенная для моделирования процессов, запускаемых только на квантовом компьютере. Проще говоря – для создания программ и визуализации результатов работы алгоритмов, написанных для квантового компьютера. Основной задачей *JQuantum* является обеспечение прозрачности работы программного кода квантового компьютера. *JQuantum* помогает визуализировать код, что очень удобно для людей, которые не знакомы с низкоуровневыми языками программирования квантового компьютера, такими как OpenQASM для IBM и т.п. При этом сам эмулятор может использовать максимум 15 кубитов при создании схемы.

Одной из приятных особенностей в *JQuantum* является возможность пользователя работать как с графическим эмулятором, так и с вычислительным кодом библиотек квантового программирования java, способных выполнять математические операции, которые необходимы для более точного квантового результата, что делает *JQuantum* удобным для пользователей с опытом программирования и без.

Для того чтобы работать с квантовыми вычислениями в среде разработки от компании JetBrains IntelliJ IDEA, необходимо импортировать библиотеку quantum, в которой хранится множество функций и гейтов квантового программирования. Данная особенность будет удобна для программирования квантового компьютера на высокоуровневом языке программирования Java или

стремительно набирающем популярность языке Kotlin, который полностью совместим с Java.

На панели управления (представленной на рис. 5) слева расположен инструмент инициализации регистров, в котором нам предлагается указать их начальные состояния, а также их количество. Там же расположены команды контроля выполнения квантовой программы. Можно запускать схему пошагово, производя отладку на каждом этапе, при этом в окне вывода результатов видны промежуточные значения выполнения схемы. Можно вернуться к предыдущему шагу, запустить всю схему целиком с окончательным выводом результата или же инициализировать схему заново, для нового запуска квантовой программы, – очень вариативно.

На панели проектирования схемы представлен набор основных гейтов, использующихся при написании квантовых программ, а также инструменты редактирования схемы. Рабочая область представляет собой набор инициализированных кубитов, на которых располагаются гейты. Окно вывода позволяет пользователю следить за состоянием регистров x и y , выводя в них результаты выполнения квантовой программы в процессе ее работы. На рис. 5 представлен пример квантовой схемы алгоритма Шора в эмуляторе *JQuantum*.

На рис. 6 представлен результат работы схемы алгоритма Шора в эмуляторе *JQuantum*. Результат работы схемы выводится в регистре y в двоичной и целочисленной записи числа.

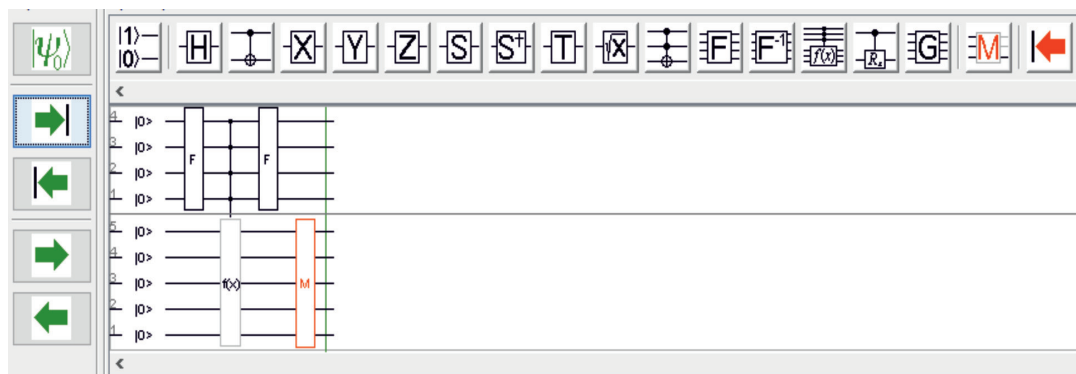


Рис. 5. Реализация схемы алгоритма Шора в эмуляторе JQuantum

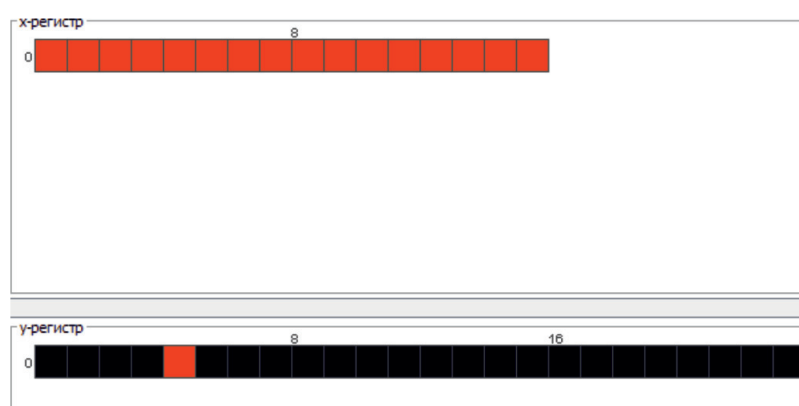


Рис. 6. Результат работы алгоритма Шора в эмуляторе JQuantum

Достоинства JQuantum:

- исходный код [19] находится в открытом доступе;
- кроссплатформенность. JQuantum может стабильно работать с версии Java 5, которая находится в свободном доступе на сайте Oracle;
- эмулятор можно запустить как апплет из браузера или предварительно установив исполняемый файл jar на рабочий стол;
- повторяемость экспериментов;
- чрезвычайная надежность вычислений;
- масштабируемость.

Недостатки JQuantum:

- данная библиотека не может работать в экстремальном масштабе;
- эмулятор не решает проблем, связанных с оптимизацией времени работы программы, сравнимого с теоретическими квантовыми компьютерами;
- для стабильного использования эмулятора необходимо установить Java 5+;
- для изменения схемы необходима полная или частичная инициализация гейтов схемы;

- не полный набор гейтов, требующий создания недостающих из уже имеющихся;
- нет возможности запуска программ на реальном квантовом компьютере.

4. *QuEST* – это первый GPU-симулятор универсальных квантовых цепей с открытым исходным кодом, гибридом OpenMP и MPI. Созданный как библиотека C, он разработан таким образом, что пользовательский код может быть легко развернут на любой платформе от ноутбука до суперкомпьютера. QuEST способен моделировать общие квантовые схемы общих однокубитных и многокубитных управляемых вентилей в чистых и смешанных состояниях, представленных в виде векторов состояний и матриц плотности, и в присутствии декогеренции [20]. На рис. 7.1 и 7.2 представлена реализация алгоритма Шора, написанного на языке программирования C++.

Достоинства QuEST:

- низкое потребление оперативной памяти на небольшом количестве кубитов [20];
- открытый исходный код;

- кроссплатформенность;
- распределение вычислений на несколько ядер процессора;
- возможность перенести вычисления на графический процессор.

Недостатки QuEST:

- нет инструментов для графического представления схемы;

– вся документация представлена только на английском языке.

5. *Quantum Programming Studio (QPS)* – это графический пользовательский веб-интерфейс для разработки квантовых алгоритмов и выполнения их на симуляторах или на реальных квантовых компьютерах, разработанный группой Quantastica в 2019 г.

```

7  void swap(Qureg qubits, const int q1, const int q2) {
8      controlledNot(qubits, q1, q2);
9      controlledNot(qubits, q2, q1);
10     controlledNot(qubits, q1, q2);
11 }
12 void ccx(Qureg qubits, const int q1, const int q2, const int q3) {
13     hadamard(qubits, q3);
14     controlledNot(qubits, q2, q3);
15     phaseShift(qubits, q3, -M_PI / 4);
16     controlledNot(qubits, q1, q3);
17     tGate(qubits, q3);
18     controlledNot(qubits, q2, q3);
19     phaseShift(qubits, q3, -M_PI / 4);
20     controlledNot(qubits, q1, q3);
21     tGate(qubits, q2);
22     tGate(qubits, q3);
23     hadamard(qubits, q3);
24     controlledNot(qubits, q1, q2);
25     tGate(qubits, q1);
26     phaseShift(qubits, q2, -M_PI / 4);
27     controlledNot(qubits, q2, q3);
28 }

```

Рис. 7.1. Реализация алгоритма Шора в эмуляторе QuEST

```

29 int main(int argc, char* argv[]) {
30     QuESTEnv env = createQuESTEnv();
31     Qureg qubits = createQureg(7, env);
32     int measured[7];
33     hadamard(qubits, 0);
34     hadamard(qubits, 1);
35     hadamard(qubits, 2);
36     paulix(qubits, 6);
37     controlledNot(qubits, 0, 5);
38     controlledNot(qubits, 0, 4);
39     paulix(qubits, 4);
40     ccx(qubits, 4, 1, 6);
41     controlledNot(qubits, 6, 4);
42     ccx(qubits, 5, 1, 3);
43     hadamard(qubits, 2);
44     controlledNot(qubits, 3, 5);
45     controlledPhaseShift(qubits, 1, 2, M_PI / 2);
46     hadamard(qubits, 1);
47     controlledPhaseShift(qubits, 0, 2, M_PI / 4);
48     controlledPhaseShift(qubits, 0, 1, M_PI / 2);
49     hadamard(qubits, 0);
50     swap(qubits, 2, 0);
51     measured[2] = measure(qubits, 2);
52     measured[1] = measure(qubits, 1);
53     measured[0] = measure(qubits, 0);
54     destroyQureg(qubits, env);
55     destroyQuESTEnv(env);
56     return 0;
57 }

```

Рис. 7.2. Реализация алгоритма Шора в эмуляторе QuEST

```

4
-----
Process exited after 1.586 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

```

Рис. 8. Результат работы алгоритма Шора в эмуляторе QuEST

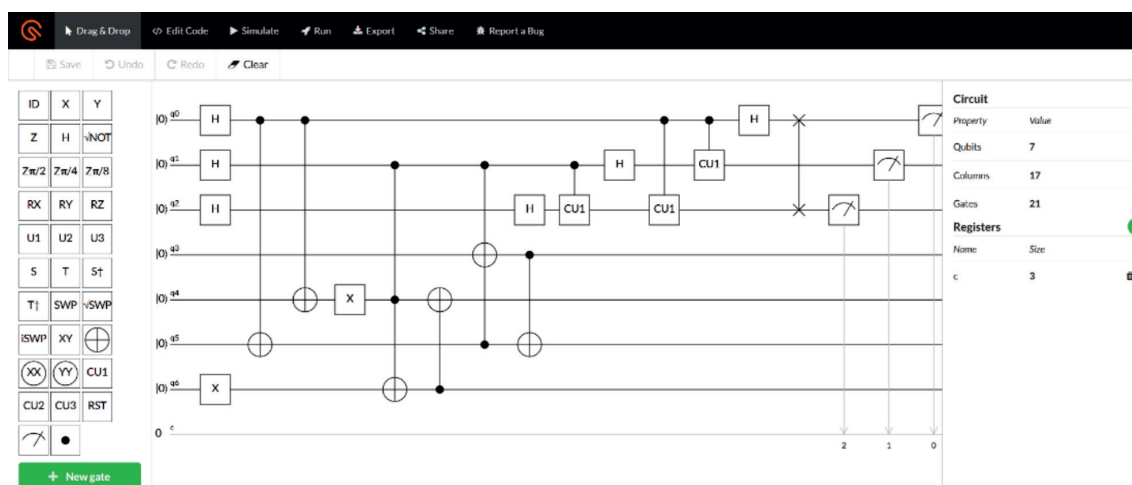


Рис. 9. Реализация схемы алгоритма Шора в эмуляторе QPS

Основной особенностью является кроссплатформенность, заключающаяся в возможности запускать программы в различных эмуляторах или существующих на момент написания статьи квантовых компьютерах. Помимо этого, в QPS существует возможность экспорта кода в наиболее популярные языки квантового программирования.

Данный эмулятор предназначен как для специалистов, так и для людей, которые только начинают постигать азы квантового программирования.

На рис. 9 представлен пример алгоритма Шора в эмуляторе Quantum Programming Studio.

Схема слегка отличается от традиционной в связи с тем, что некоторые необходимые алгоритмы были заранее реализованы. Результат выводится в похожем наглядном виде, представленном на рис. 10.

Достоинства квантового эмулятора QPS:

- QPS является веб-приложением и, как следствие, доступен для любого устройства с выходом в Интернет;
- кроссплатформенность;
- возможность экспорта кода на другие языки;
- наглядное построение схем;
- возможность запустить программу на квантовом компьютере;
- открытый исходный код.

Classical registers

Register	Bin	Hex	Dec
c	100	4h	4

Local state

Qubit	Measured	Probability of 1	θ °deg	φ °deg	Bloch
q0	0	0	0	0	
q1	0	0.5	90	0	
q2	1	0.5	90	0	
q3	1	0.25	60	0	
q4	1	0.75	120	0	
q5	0	0.25	60	0	
q6	1	0.75	120	0	

Рис. 10. Результат схемы алгоритма Шора в эмуляторе QPS

Недостатки квантового эмулятора QPS:

- QPS является веб-приложением, и, как следствие, нет возможности автономной работы;
- запуск схемы с 24 кубитами и больше приведёт к падению браузера;

– сравнительно небольшое количество заранее реализованных алгоритмов.

6. *Quantum-Kit (Q-Kit)* – это графический симулятор квантовых цепей. Он позволяет строить и проектировать квантовые схемы, визуализируя влияние операций с квантовыми гейтами в виде распределений квантовых состояний или на блоховскую сферу.

Q-Kit – бесценный инструмент в интуитивном понимании квантовых цепей, таких как квантовая телепортация, факторизация Шора или алгоритмы поиска Гровера. Это позволяет экспериментировать с различными квантовыми алгоритмами путем моделирования кубитов для изучения сложных квантовых систем на компьютере, а не в лаборатории. Это также позволяет тестировать квантовое программное обеспечение на эмуляторе перед запуском на квантовом оборудовании. Q-Kit имеет приложения не только для исследования, разработки и про-

верки квантовых алгоритмов, но и для интуитивного понимания квантовых схем, как высокоуровневого предшественника квантового программирования.

Данный эмулятор хорошо подойдет для тех, кто только начал разбираться с квантовым программированием, ввиду простого интерфейса и наглядности вычислений. На рис. 11 представлена реализация алгоритма Шора в среде эмулятора Q-Kit.

На рис. 12 представлен результат работы схемы алгоритма Шора.

Достоинства квантового эмулятора Q-Kit:

- удобный интерфейс;
- относительная быстрота работы схем [21];
- простое построение схем;
- кроссплатформенность;
- большое количество встроенных алгоритмов и гейтов;
- возможность создавать собственные гейты.



Рис. 11. Реализация схемы алгоритма Шора в эмуляторе Q-Kit

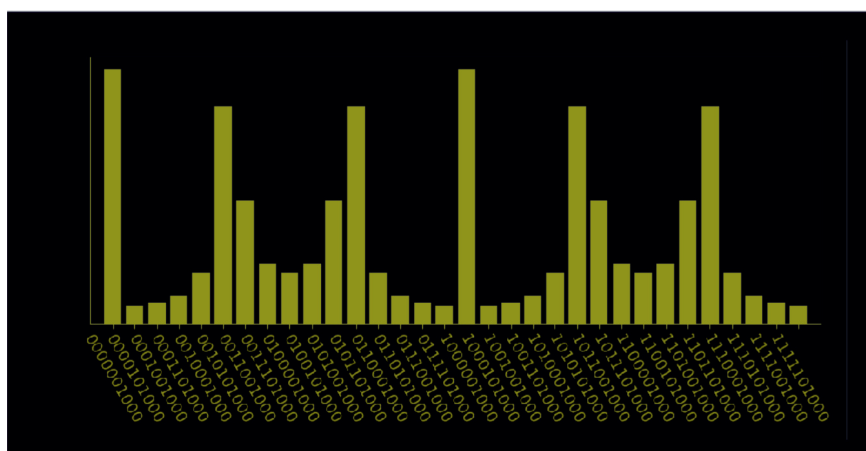


Рис. 12. Результат работы схемы алгоритма Шора в эмуляторе Q-Kit

Сравнительная таблица платформ

	QDK	QCP	JQuantum	QuEST	QPS	Quantum-Kit
Поддерживаемые платформы	Windows, macOS, Linux	Web	Windows	Windows, macOS, Linux	Web	Windows, macOS, Linux
Исходный код	(+)	(+)	(+)	(+)	(+)	(-)
Графическое приложение или библиотека	Библиотека Q#	Web-приложение	Графическое приложение	Библиотека для C++	Web-приложение	Графическое приложение
Количество логических кубитов	32	22	15	29	24	30
Количество реализованных гейтов	17	21	16	29	30	21
Возможность создать свои гейты	(+)	(+)	(-)	(+)	(+)	(+)
Наличие внутреннего языка программирования	(+)	(+)	(+)	(+)	(+)	(-)
Документация	(+)	(-)	(+)	(+)	(+)	(-)

Недостатки квантового эмулятора Q-Kit:

- ограничение в 20 кубит;
- присутствие визуальных багов;
- отсутствует встроенный язык.

На сегодняшний день имеется большое количество разнообразных эмуляторов квантовых компьютеров, которые могут стать хорошим инструментом в руках человека, исследующего квантовые вычисления. На просторах Интернета хранится огромное число методических пособий, руководств и документов, описывающих принципы работы с квантовыми эмуляторами [22].

В таблице приведены обобщенные результаты данного исследования.

Выводы

В данной статье был рассмотрен ряд квантовых эмуляторов, которые хорошо подходят для начала работы с квантовыми вычислениями. При этом их всех объединяет то, что начало работы с данными эмуляторами не вызовет у пользователя каких-либо затруднений. Все рассмотренные эмуляторы обладают удобным интерфейсом, который также способствует повышению продуктивности пользователя при работе с кодом. Для большей части эмуляторов есть актуальная, структурированная и понятная документация, изучение которой поможет освоить квантовое программирование и написать первую квантовую программу. При этом эмуляторы поддерживают возможность отладки программы, что позволяет оценить корректность работы кода. В данной статье были выбраны и проанализированы наиболее подходящие эмуляторы для начала работы с квантовыми вычислениями. Невозможно сказать, что какой-то из

них лучше или хуже. Можно лишь рассмотреть их особенности, которые могут быть удобны на тех или иных уровнях квантового программирования. Также возможно разделить эмуляторы на следующие группы для удобства:

– для тех, кто только начинает изучать квантовые вычисления, хорошо подойдут QPS, JQuantum, QCP, с примерами квантовых алгоритмов;

– для тех, кто знаком с программированием на разных языках, QDK, QuEST и JQuantum – отличный выбор для начала и продолжения работы с квантовыми вычислениями;

– для изучения поведения схем отлично подойдут Q-kit, JQuantum, QPS;

– а QPS – хороший инструмент для изучения квантовых языков программирования в общем виде.

Таким образом, квантовые вычисления теперь доступны не только узкой прослойке исследователей, трудящихся в передовых учреждениях с квантовыми компьютерами в их физическом виде. Доступ к вышеперечисленным эмуляторам так прост, что начать заниматься квантовым программированием под силу даже студенту. Осталось только выбрать подходящий эмулятор из списка.

Список литературы

1. Intel. Amazing Performance and Responsiveness. [Electronic resource]. URL: <https://www.intel.ru/content/www/ru/ru/homepage.html> (date of access: 27.06.2020).
2. Dunne P.E. An annotated list of selected NP-complete problems. Dept. of Computer Science, University of Liverpool. 2008. [Electronic resource]. URL: https://cgi.csc.liv.ac.uk/~ped/teachadmin/COMP202/annotated_np.html (date of access: 27.06.2020).

3. Georgescu M., Ashhab S., Franco Nori. «Quantum SimulationI». Cornell University. 2013. [Electronic resource]. URL: <https://arxiv.org/abs/1308.6253> (date of access: 27.06.2020).
4. Душкин Р.В. Квантовые вычисления и функциональное программирование. М.: ДМК-Пресс, 2015, 232 с.
5. Berkeley of University of California, «Quantum Mechanics and Quantum Computation». [Electronic resource]. URL: <https://www.edx.org/course/quantum-mechanics-and-quantum-computation#VOugFrDF9sA> (date of access: 27.06.2020).
6. National Academies of Sciences, Engineering, and Medicine 2018. Quantum Computing: Progress and Prospects. Washington, DC: The National Academies Press. DOI: 10.17226/25196.
7. Quantum Development Kit. [Electronic resource]. URL: <https://www.microsoft.com/en-us/quantum/development-kit> (date of access: 27.06.2020).
8. Quantum Computing Playground. [Electronic resource]. URL: <http://www.quantumplayground.net> (date of access: 27.06.2020).
9. Andreas de Vries A Quantum Computer Simulator. [Electronic resource]. URL: <http://jqquantum.sourceforge.net/> (date of access: 27.06.2020).
10. QuEst. Quantum Exact Simulation Toolkit. [Electronic resource]. URL: <https://quest.qtechtheory.org/> (date of access: 27.06.2020).
11. Quantum Programming Studio. [Electronic resource]. URL: <https://quantum-circuit.com> (date of access: 27.06.2020).
12. Quantum-Kit. [Electronic resource]. URL: <https://sites.google.com/view/quantum-kit> (date of access: 27.06.2020).
13. Quantiki. List of QC simulators. [Electronic resource]. URL: <https://quantiki.org/wiki/list-qc-simulators> (date of access: 27.06.2020).
14. GitHub. Quantum Ktas. [Electronic resource]. URL: <https://github.com/Microsoft/QuantumKatas/> (date of access: 27.06.2020).
15. GitHub. Microsoft Quantum Development Kit Samples. [Electronic resource]. URL: <https://github.com/microsoft/Quantum> (date of access: 27.06.2020).
16. Microsoft documentation. Microsoft Quantum Development Kit. [Electronic resource]. URL: <https://docs.microsoft.com/en-us/quantum> (date of access: 27.06.2020).
17. Microsoft Azure Documentation. Azure Quantum. [Electronic resource]. URL: <https://azure.microsoft.com/ru-ru/services/quantum/?cdn=disable> (date of access: 27.06.2020).
18. GitHub. Quantum-Computing-Playground. [Electronic resource]. URL: <https://github.com/gwroblew/Quantum-Computing-Playground> (date of access: 27.06.2020).
19. GitHub. JQuantum. [Electronic resource]. URL: <https://github.com/chasenorman/JQuantum> (date of access: 27.06.2020).
20. Tyson Jones «QuEST and High Performance Simulation of Quantum Computers». 2018. [Electronic resource]. URL: <https://arxiv.org/pdf/1802.08032.pdf> (date of access: 27.06.2020).
21. Archana Tankasala «Q-Kit: SimulatingShor's Factorization of 24-Bit Number on Desktop». [Electronic resource]. URL: <https://arxiv.org/ftp/arxiv/papers/1908/1908.07187.pdf> (date of access: 27.06.2020).
22. Quantum Computing Report. Education. [Electronic resource]. URL: <https://quantumcomputingreport.com/resources/education/> (date of access: 27.06.2020).

УДК 004.021

ИССЛЕДОВАНИЕ МОДЕЛЕЙ СО СКРЫТЫМ РАССТОЯНИЕМ ДЛЯ ВНЕДРЕНИЯ СУЩНОСТЕЙ И ИХ ОТНОШЕНИЙ В ГРАФ ЗНАНИЙ

Хлопенкова А.Ю., Белов Ю.С.

Московский государственный технический университет имени Н.Э. Баумана, филиал, Калуга,
e-mail: iu4-kf@mail.ru

В данной статье рассматриваются алгоритмы моделирования графа знаний, основанные на моделях со скрытым расстоянием. Демонстрируется представление внедряемых сущностей в векторном пространстве. Описывается роль использования триплетов, содержащих сущности и отношения, для построения графа. Для каждого из алгоритмов приводится функция вычисления обучаемой модели. Модель TransE представляет собой первоначальную реализацию с использованием триплетов (h, r, t) для представления отношений в виде трансляций в векторном пространстве и является базовой для всех остальных. В моделях TransH и TransR для реализации графа знаний подчеркивается необходимость использования гиперплоскостей. Так, например, отношение моделируется как вектор на собственной гиперплоскости. На примере модели TransD демонстрируется использование матрицы динамического отображения для внедрения сущностей. При обучении модели внедрению именованных символьных объектов (сущностей или отношений) TransD может учитывать их разнообразие. В модели KG2E приводится пример с использованием распределения Гаусса. В качестве меры расстояния используются расстояние Кульбака – Лейблера и оценочная вероятность. В заключение делаются выводы по каждой из представленных моделей. Подчеркиваются преимущества каждой из описанных моделей.

Ключевые слова: граф знаний, модели со скрытым расстоянием, триплеты, сущности, отношения, внедрение

RESEARCH OF LATENT DISTANCE MODELS FOR EMBEDDING OF ENTITIES AND THEIR RELATIONSHIPS IN THE KNOWLEDGE GRAPH

Khlopenkova A.Yu., Belov Yu.S.

Bauman Moscow State Technical University, Kaluga branch, Kaluga, e-mail: iu4-kf@mail.ru

The article discusses knowledge graph modeling algorithms based on latent distance models. The representation of embedded entities in vector space is demonstrated. The role of using triplets containing entities and relationships for graph construction is described. For each of the algorithms, the calculation function for the training model is given. The TransE model is an initial implementation which use triplets (h, r, t) to represent relationships in the form of translations in a vector space and it is basic for all others. TransH and TransR models emphasize the need for hyperplanes to implement the knowledge graph. So, for example, a relation is modeled as a vector on its own hyperplane. Using the TransD model as an example, the use of a dynamic mapping matrix for implementing entities is demonstrated. During the learning embeddings of named symbol objects (entities or relations), TransD could consider the diversity of them both. The KG2E model provides an example using the Gaussian distribution. The Kullback – Leibler distance and the estimated probability are used as a distance measure. Finally, draws conclusions for each of the presented models. The advantages of each described model are highlighted.

Keywords: knowledge graph, latent distance models, triplets, entities, relationships, embedding

Граф знаний – это мультиреляционный граф, состоящий из сущностей, которые могут быть представлены как в виде связки узлов и отношений, так и в виде разнотипных ребер. В качестве экземпляра ребра может быть представлена тройка параметров (голова, отношение, хвост), она обозначается как (h, r, t) от англ. (head, relation, tail).

Наиболее важные варианты использования графов знаний в объединении с ИИ включают в себя [1]:

- интуитивно понятный поиск с использованием естественного языка;
- обнаружение соответствующего контента и информации в структурированных или неструктурированных данных;
- надежный контент и управление данными;
- прогнозирование операционного риска; и т.п.

Цель исследования: рассмотреть модели со скрытым расстоянием, используемые для реализации графа знаний. Сравнить обучаемые модели и выделить особенности каждой из них.

Объект исследования

Одними из алгоритмов реализации графа знания являются модели со скрытым расстоянием. Данные модели используют функции рейтинговой оценки, основанные на расстоянии между сущностями, для моделирования тройки графа знаний. Взаимодействие сущностей и отношений осуществляется через их скрытое пространственное представление.

TransE. TransE (от англ. Translating Embeddings for Modeling Multi-relational Data – Транслирование внедряемых сущностей в векторном пространстве для моде-

лирования мультиреляционных данных) – это модель, основанная на ЕВМ (от англ. Energy-Based Model), которая представляет отношения в виде трансляций в векторном пространстве. В частности, предполагается, что если триплет (h, r, t) имеет место, то внедрение сущности хвоста « t » должно быть близко к внедрению головной

сущности « h » плюс некоторый вектор, который зависит от отношения « r ». В TransE и сущности, и отношения являются векторами в одном и том же пространстве. Чтобы изучить такие внедрения, следует на обучающей выборке минимизировать критерий ранжирования, основанный на границе Υ [2]:

$$\mathcal{L} = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'_{(h,r,t)}} [\gamma + d(h+r, t) - d(h' + r', t')]_+,$$

$$S'_{(h,r,t)} = \{(h', r, t) | h' \in E\} \cup \{(h, r, t') | t' \in E\},$$

$$d(h+r, t) = \|h\|_2^2 + \|r\|_2^2 + \|t\|_2^2 - 2(h^T t + r^T (t - h)),$$

$$\|h\|_2^2 = \|t\|_2^2 = 1,$$

где $[\]_+$ – представляет собой только положительную часть,
 Υ – граница, разделяющая положительные и отрицательные тройки,
 d – функция расстояния,
 S – множество положительных триплетов (h, r, t) ,
 S' – множество отрицательных триплетов,
 $\mathcal{L}$ – обучаемая модель,
 E – множество сущностей, $h, t \in E$.

TransH. TransH (от англ. Knowledge Graph Embedding by Translating on Hyperplanes – внедрение сущностей графа знаний путем транслирования на гиперплоскости) следует общему принципу TransE. Однако, по сравнению с ним, он вводит конкретные для отношений гиперплоскости. Сущности представлены в виде векторов, как

и в TransE, однако отношение моделируется как вектор на собственной гиперплоскости с вектором нормали (рис. 1). Затем сущности проецируются на гиперплоскость отношения для расчета потерь. Чтобы стимулировать различие между хорошими триплетами и неправильными триплетами, используется следующий предельный рейтинг потери [3]:

$$\mathcal{L} = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'_{(h,r,t)}} [\gamma + f_r(h, t) - f_r(h', t')]_+,$$

$$f_r(h, t) = (h - w_r^T h w_r)_2 + d_r - (t - w_r^T t w_r)_2^2,$$

где w_r – гиперплоскость для отношения r ,
 f_r – функция оценки.

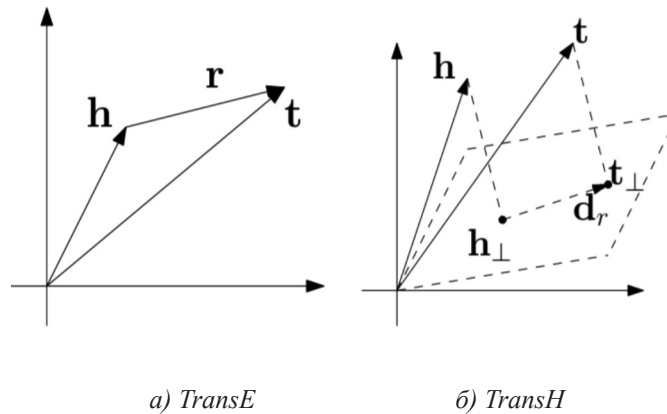


Рис. 1. Иллюстрация алгоритмов моделирования графа знаний

TransR. TransR (от англ. Relation Embeddings for Knowledge Graph Completion – внедрение отношений для заполнения графа знаний) очень похож на TransH, с той лишь разницей, что вместо того, чтобы иметь одну гиперплоскость отношения, он вводит множество гиперплоскостей (рис. 2). Сущности являются векторами в пространстве сущностей, и каждое отношение является вектором в определенном пространстве отношения. Для расчета потерь сущности проецируются в отношении конкретного пространства с использованием матрицы проекции. Для каждой тройки (h, r, t) сущности в пространстве сущностей сначала проецируются в пространство r -отношений h_r и t_r с помощью операции M_r , а затем приводят к уравнению вида $h_r + r \approx t_r$ [4].

$$\mathcal{L} = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'_{(h,r,t)}} \max(0, \gamma + f_r(h, t) - f_r(h', t')),$$

$$f_r(h, t) = h_r + r - t_{r2},$$

$$h_r = h M_r, t_r = t M_r,$$

TransD. TransD (от англ. Knowledge Graph Embedding via Dynamic Mapping Ma-

trix – внедрение сущностей графа знаний с помощью матрицы динамического отображения) – улучшенная версия TransR. Для каждого триплета (h, r, t) используются две матрицы отображения $M_{rh}, M_{rt} \in R^{m \times n}$ для проецирования сущностей из пространства сущностей в пространство отношений. M_{rh}, M_{rt} сопоставляют матрицы h, t соответственно (рис. 3) [5].

$$\mathcal{L} = \sum_{\xi \in S} \sum_{\xi' \in S'} [\gamma + f_r(\xi') - f_r(\xi)]_+,$$

$$\xi = (h, r, t), \xi' = (h', r, t'),$$

$$f_r(h, t) = -h_{\perp} + r - t_{\perp 2},$$

$$h_{\perp} = M_{rh} h, t_{\perp} = M_{rt} t,$$

$$M_{rh} = r_p h_{ip} + I^{m \times n},$$

$$M_{rt} = r_p t_{ip} + I^{m \times n},$$

где r – вектор проекции, M_{rh}, M_{rt} – матрицы динамического отображения, $I^{m \times n}$ – единичная матрица размером $m \times n$.

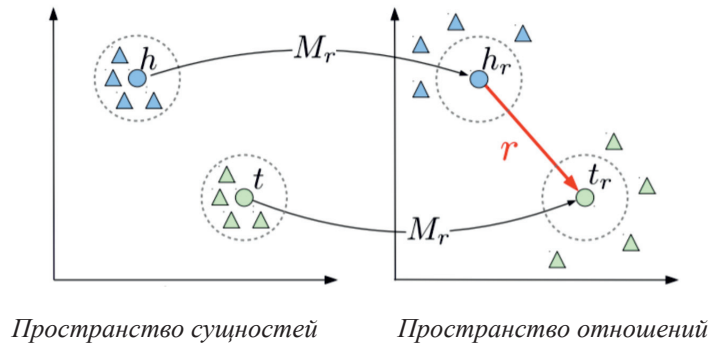


Рис. 2. Иллюстрация алгоритма TransR

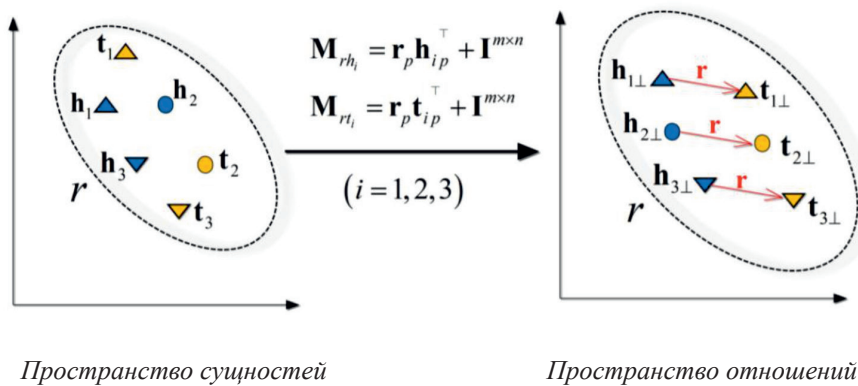


Рис. 3. Иллюстрация алгоритма TransD

TransM. TransM (от англ. Transition-based Knowledge Graph Embedding with Relational Mapping Properties – внедрение отношений для построения графа знаний на основе переходов со свойствами реляционного отображения) помогает устранить отсутствие гибкости алгоритма в TransE, когда дело доходит до сопоставления свойств триплетов. На рис. 4 продемонстрирована разница между алгоритмами. Используется структура графа знаний посредством предварительного вычисления индивиду-

ального веса для каждого тестового триплета в соответствии с его свойством реляционного отображения. Простой способ смоделировать структуру – связать каждый тестовый триплет с весом, который представляет степень отображения. Согласно наблюдениям, свойство отображения триплета во многом зависит от его отношений внутри графа [6].

$$\mathcal{L} = \min \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'_{(h,r,t)}} [\gamma + f_r(h,t) - f_r(h',t')]_+.$$

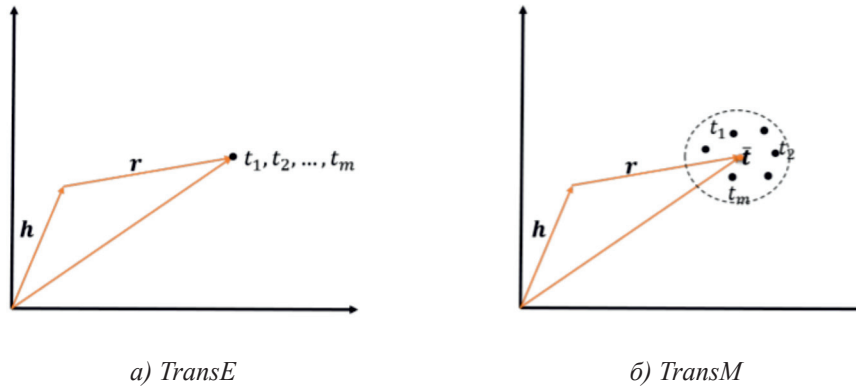


Рис. 4. Различия между алгоритмами при моделировании экземпляров отношения один ко многим

KG2E. KG2E (от англ. Knowledge Graphs with Gaussian Embedding – графы знаний с внедрением распределения Гаусса) вместо того, чтобы представлять сущности и отношения в качестве детерминированных точек в векторном пространстве, моделирует сущности и отношения (h, r, t) , используя случайные величины, полученные из многомерного Гауссовского распределения. Затем KG2E оценивает события, используя отношение трансляций, оценивая расстояние между двумя распределениями r и $t-h$. KG2E предоставляет на выбор две меры расстояния (расстояние Кульбака – Лейблера и оценочная вероятность) [7].

$$\mathcal{L} = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'_{(h,r,t)}} [\gamma + \mathcal{E}(h,r,t) - \mathcal{E}(h',r',t')]_+.$$

Расстояние Кульбака – Лейблера

$$\begin{aligned} \mathcal{E}(h,r,t) &= \mathcal{E}(P_e P_r) = D_{KL}(P_e P_r) = \int_{x \in R^{k_e}} \mathcal{N}(x; \mu_r, \Sigma_r) \log \frac{\mathcal{N}(x; \mu_e, \Sigma_e)}{\mathcal{N}(x; \mu_r, \Sigma_r)} dx = \\ &= \frac{1}{2} \left\{ \text{tr} \left(\Sigma_r^{-1} \Sigma_e \right) + (\mu_r - \mu_e)^T \Sigma_r^{-1} (\mu_r - \mu_e) - \log \frac{\det(\Sigma_e)}{\det(\Sigma_r)} - k_e \right\}. \end{aligned}$$

Оценочная вероятность

$$\begin{aligned} \mathcal{E}(h,r,t) &= \log \mathcal{E}(P_e P_r) = \log \mathcal{N}(0; \mu_e - \mu_r, \Sigma_e + \Sigma_r) = \\ &= \frac{1}{2} \left\{ (\mu_e - \mu_r)^T (\Sigma_e + \Sigma_r)^{-1} (\mu_e - \mu_r) + \log \det(\Sigma_e + \Sigma_r) + k_e \log(2\pi) \right\}, \end{aligned}$$

$$\mathcal{E}(P_e P_r) = \int_{x \in R^{k_e}} \mathcal{N}(x; \mu_e, \Sigma_e) \mathcal{N}(x; \mu_r, \Sigma_r) dx = \mathcal{N}(0; \mu_e - \mu_r, \Sigma_e + \Sigma_r),$$

где $tr(\Sigma_r)$ – трассировка ковариационной матрицы,

$\mathcal{E}$ – множество сущностей триплета,

D_{KL} – расстояние Кульбака – Лейблера,

Σ_e, Σ_r – ковариационные матрицы,

μ – вектор средних,

$\mathcal{N}(\mu, \Sigma)$ – многомерное Гауссово распределение (с диагональной ковариацией для эффективности вычислений),

P_e, P_r – распределение вероятностей и отношений соответственно,

k_e – размер сущности в скрытом векторном пространстве,

$\det$ – определитель матрицы.

Заключение

В заключение можно выделить основные характеристики описанных алгоритмов реализации графа знаний. При максимуме внимания на минимальной параметризации модели был разработан алгоритм TransE, чтобы в первую очередь представлять иерархические отношения. Он является хорошо масштабируемой моделью. TransH преодолевает недостатки TransE, связанные с рефлексивными отношениями «один ко многим / многие к одному / многие ко многим», при этом наследуя его эффективность. Обширные эксперименты классификации триплетов и извлечения реляционных сущностей показывают, что TransH приносит многообещающие улучшения в TransE. TransR предлагает внедрения посредством трансляции между проецируемыми сущностями. TransD имеет меньшую сложность и большую гибкость, чем TransR / CTransR. Обширные эксперименты показывают, что TransD превосходит TransE, TransH и TransR / CTransR по двум задачам, а именно классификации триплетов и прогнозированию ссылок. TransM – превосходная модель, которая не только идеальна для представления иерархических и нерефлексивных характеристик, но также гибкая для адаптации различных свойств отображения триплетов знаний. Результаты обширных экспериментов с несколькими эталонными

наборами данных доказывают, что модель может достичь более высокой производительности без ущерба для эффективности. KG2E представляет собой новый метод для представлений сущностей и отношений. Обширные эксперименты по прогнозированию ссылок и классификации триплетов с несколькими наборами эталонных данных (включая WordNet и Freebase) демонстрируют, что предлагаемый метод превосходит аналоговые методы.

Список литературы

1. Хлопенкова А.Ю., Белов Ю.С. Методы обработки естественного языка в виртуальных голосовых помощниках // E-Scio Электронное периодическое издание «E-Scio.ru» – Эл № ФС77-66730. [Электронный ресурс]. URL: <http://e-scio.ru/wp-content/uploads/2019/11/Хлопенкова-А.-Ю.-Белов-Ю.-С..pdf> (дата обращения: 01.07.2020).
2. Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. Advances in Neural Information Processing Systems 26. 2013. [Electronic resource]. URL: <http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf> (date of access: 01.07.2020).
3. Zhen Wang, Jianwen Zhang, Jianlin Feng, Zheng Chen. Knowledge Graph Embedding by Translating on Hyperplanes. AAAI Publications, Twenty-Eighth AAAI Conference on Artificial Intelligence. 2014. [Electronic resource]. URL: <https://persagen.com/files/misc/wang2014knowledge.pdf> (date of access: 01.07.2020).
4. Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, Xuan Zhu. Learning Entity and Relation Embeddings for Knowledge Graph Completion. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015. [Electronic resource]. URL: http://nlp.csai.tsinghua.edu.cn/~lyk/publications/aaai2015_transr.pdf (date of access: 01.07.2020).
5. Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, Jun Zhao. Knowledge Graph Embedding via Dynamic Mapping Matrix. Association for Computational Linguistics. 2015. DOI 10.3115/v1/P15-1067. [Electronic resource]. URL: <https://www.aclweb.org/anthology/P15-1067.pdf> (date of access: 01.07.2020).
6. Miao Fan, Qiang Zhou, Emily Chang, Thomas Fang Zheng. Transition-based Knowledge Graph Embedding with Relational Mapping Properties. Department of Linguistics, Chulalongkorn University. 2014. [Electronic resource]. URL: <https://www.aclweb.org/anthology/Y14-1039.pdf> (date of access: 01.07.2020).
7. Shizhu He, Kang Liu, Guoliang Ji and Jun Zhao. Learning to Represent Knowledge Graphs with Gaussian Embedding. Conference: the 24th ACM International. 2015. DOI: 10.1145/2806416.2806502. [Electronic resource]. URL: <http://www.nlpr.ia.ac.cn/cip/~liukang/liukangPageFile/Learning%20to%20Represent%20Knowledge%20Graphs%20with%20Gaussian%20Embedding.pdf> (date of access: 01.07.2020).

УДК 004.6

СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ОБЪЕКТОВ В БАЗЕ ДАННЫХ ORACLE

Журавлев А.А., Аксенов К.А.

*Уральский федеральный университет им. Б.Н. Ельцина, ИРИТ РмФ, Екатеринбург,
e-mail: SanyaProgrammer2503@gmail.com, k.a.aksenov@urfu.ru*

Базы данных играют огромную роль в работе разработчика приложений. Они необходимы для хранения и анализа различной информации. Отличительной чертой базы данных от других хранилищ типа архивов является то, что информация в ней обязательно подвергается обработке электронными носителями с последующей систематизацией, которая подчинена определенным правилам. Основными задачами базы данных являются: обеспечение хранения необходимой информации, возможность получения конкретной информации, а также обеспечение целостности базы данных. Наиболее интересными случаями применения базы данных являются хранение и использование объектов. В данной статье представлено создание и использование объектов в базе данных Oracle на примере геометрической фигуры (треугольника). В статье использован эмпирический метод исследования, поскольку в качестве основного источника результата взят анализ над данными (объектами). В ходе написания статьи получены следующие результаты: с помощью базы данных Oracle построена новая таблица с полем-объектом, для хранения треугольников; создана отдельная (вторая) таблица со строками-объектами и заполнена объектами из первой таблицы; создан столбец-ссылка на объекты дополнительной таблицы; сделаны соответствующие выводы о проделанной работе.

Ключевые слова: объекты, базы данных, создание, использование, Oracle

CREATING AND USING OBJECTS IN ORACLE DATABASE

Zhuravlev A.A., Aksenov K.A.

*Ural Federal University named after B.N. Yeltsin, IRIT RmF, Yekaterinburg,
e-mail: SanyaProgrammer2503@gmail.com, k.a.aksenov@urfu.ru*

Databases play a huge role in the work of the application developer. It is necessary for storing and analyzing different information. A distinguishing feature of the database from other repositories such as archives is that the information in it is necessarily processed by electronic media with subsequent systematization, which is subject to certain rules. The main objectives of the database are: ensuring the storage of necessary information, the ability to obtain specific information, as well as ensuring the integrity of the database. The most interesting database application is storing objects. This article presents the creation and use of objects in an Oracle database using the example of a geometric figure (triangle). The article uses the empirical research method, since the analysis of data (objects) is taken as the main source of the result. During the writing of the article, the following results were obtained: using the Oracle database, a new table with an object field was built for storing triangles; a separate (second) table was created with rows-objects and filled with objects from the first table; created a column link to the objects of the additional table; appropriate conclusions were made about the work done.

Keywords: objects, database, creating, using, Oracle

Базы данных играют огромную роль в работе разработчика приложений. Они необходимы для хранения и анализа различной информации [1–3].

Наиболее интересными случаями применения базы данных являются хранение и использование объектов. В данной статье будет представлено создание и использование объектов в базе данных Oracle на примере геометрической фигуры (треугольника). Для данного объекта будут выполнены следующие операции:

- построение таблицы для хранения типа Треугольник;
- создание дополнительной таблицы для хранения таблицы, хранящей объекты типа Треугольник;
- создание столбца-ссылки на объекты дополнительной таблицы.

Цель данной статьи – создать базу данных, хранящую информацию о рассматриваемом типе данных (треугольнике), провести работу с созданной БД в соответствии

с заданием, представленным в разделе «Описание проектируемой базы данных».

Материалы и методы исследования

В качестве материала исследования выступает база данных Oracle.

В статье используется эмпирический метод исследования, поскольку основной источник результатов – анализ.

Результаты исследования и их обсуждение

Описание проектируемой базы данных

Треугольник в трехмерном пространстве с осями X, Y, Z, заданный координатами вершин. Для вершины (точки в пространстве) определяется объектный тип с тремя числовыми атрибутами – координатами. Для треугольников задается свой тип с тремя атрибутами – вершинами и методом, вычисляющим наибольшее расстояние точки, находящейся на треугольнике от плоскости X-O-Y.

Задание

1. Построить в БД новую таблицу с полем-объектом, для хранения геометрических фигур. Столбцы таблицы содержат: идентификатор фигуры – целое (первичный ключ), название фигуры – строка до 30 символов, объект – геометрическая фигура из варианта задания. Загрузить 3-4 строки таблицы данными. С помощью запросов вывести атрибуты и результаты выполнения методов для каждого объекта таблицы.

2. Создать отдельную (вторую) таблицу со строками-объектами и заполнить ее объектами из первой таблицы. Вывести атрибуты и результаты выполнения методов для каждого объекта созданной таблицы.

3. В первую таблицу добавить столбец-ссылку на объекты второй таблицы и занести в его некоторые поля ссылку на соответствующий объект второй таблицы. Пользуясь ссылками, вывести значение любого атрибута и результат выполнения метода для соответствующего объекта второй таблицы.

Создание типов Точка и Треугольник

Для работы с описанной базой данных необходимо создание двух типов объекта: Точка и Треугольник. Ниже представлено создание типа Точка (POINT, хранит координаты) и Треугольник (TRIANGLE, хранит вершины типа Точка), имеющего метод `max_distance_from_plane` (максимальное расстояние от плоскости XOY) [4–6].

```
--Создание типа Точка
CREATE OR REPLACE TYPE POINT AS OBJECT (
  x FLOAT,
  y FLOAT,
  z FLOAT
);

--Создание типа Треугольник
CREATE OR REPLACE TYPE TRIANGLE AS OBJECT (
  A POINT,
  B POINT,
  C POINT,
  --Функция для определения максимального расстояния от треугольника до
  --плоскости XOY
  MEMBER FUNCTION max_distance_from_plane(t TRIANGLE) RETURN FLOAT
);

/
CREATE OR REPLACE TYPE BODY TRIANGLE AS MEMBER FUNCTION max_distance_from_plane(t TRIANGLE) RETURN FLOAT IS
BEGIN
  IF (ABS(t.A.z) >= ABS(t.B.z) AND ABS(t.A.z) >= ABS(t.C.z)) THEN
    RETURN ABS(t.A.z);
  ELSE IF (ABS(t.B.z) >= ABS(t.A.z) AND ABS(t.B.z) >= ABS(t.C.z)) THEN
    RETURN ABS(t.B.z);
  ELSE
    RETURN ABS(t.C.z);
  END IF;
END IF;
END;

END;
```

Наибольшим расстоянием от плоскости X-O-Y до треугольника является вершина треугольника с наибольшим по модулю значением координаты z.

*Использование объектов**Построение таблицы для хранения геометрических фигур*

Для хранения данных о треугольниках создана таблица Figures. Данная таблица содержит следующие атрибуты:

- id (ID треугольника) имеет тип Number (число), является первичным ключом таблицы;
 - name (название треугольника) имеет тип VARCHAR2 (строка);
 - object (хранит данные о вершинах треугольника) имеет тип Triangle (треугольник).
- Ниже представлено создание таблицы Figures (с помощью команды CREATE TABLE), хранящей данные о геометрических фигурах [7–9].

```
--Создание таблицы Фигуры
CREATE TABLE FIGURES (
  id NUMBER(10) NOT NULL,
  name VARCHAR2 (30) NOT NULL,
  object TRIANGLE NOT NULL,
  CONSTRAINT pk_id PRIMARY KEY (id)
);
```

Вставка в таблицу Figures 3 строк с помощью команды INSERT INTO представлена ниже.

```
--Вставка строк в таблицу
INSERT INTO FIGURES(id, name, object)
SELECT 1, 'Треугольник № 1', TRIANGLE(POINT(1, 1, 1), POINT(1, 3, 1), POINT(1, 2, 3)) FROM DUAL
UNION ALL
SELECT 2, 'Треугольник № 2', TRIANGLE(POINT(2, 2, 2), POINT(2, 6, 2), POINT(2, 4, 6)) FROM DUAL
UNION ALL
SELECT 3, 'Треугольник № 3', TRIANGLE(POINT(3, 3, 3), POINT(3, 9, 3), POINT(3, 6, 9)) FROM DUAL;
```

Выведем значения атрибутов и вычисленное для каждого треугольника максимальное расстояние до плоскости XOY.

```
SELECT
  f.id "ID", f.name "Название фигуры",
  f.object.A "Вершина A", f.object.B "Вершина B", f.object.C "Вершина C",
  f.object.max_distance_from_plane(f.object) "Максимальное расстояние от плоскости XOY"
FROM FIGURES f;
```

Результат запроса изображен на рис. 1.

Название фигуры	ID	Вершина A	Вершина B	Вершина C	Максимальное расстояние от плоскости XOY
Треугольник № 1	1	[SYSTEM. POINT]	[SYSTEM. POINT]	[SYSTEM. POINT]	3
Треугольник № 2	2	[SYSTEM. POINT]	[SYSTEM. POINT]	[SYSTEM. POINT]	6
Треугольник № 3	3	[SYSTEM. POINT]	[SYSTEM. POINT]	[SYSTEM. POINT]	9

Рис. 1. Результат запроса

Создание дополнительной таблицы

Создание (команда CREATE TABLE) дополнительной таблицы NewFigures (данная таблица имеет такие же атрибуты (id, name и object)), содержащей объекты из первой таблицы, представлено ниже [10].

```
--Создание дополнительной таблицы
CREATE TABLE NEWFIGURES OF TRIANGLE;
```

Заполнение (команда INSERT INTO) таблицы NewFigures выполнено с помощью кода, представленного ниже.

```
--Заполнение таблицы
INSERT INTO NEWFIGURES SELECT f.object FROM FIGURES f;
SELECT
  f.A "Вершина A", f.B "Вершина B", f.C "Вершина C",
  f.max_distance_from_plane (TRIANGLE(f.A, f.B, f.C)) "Максимальное расстояние от плоскости XOY"
FROM NEWFIGURES f;
```

Данные созданной таблицы представлены на рис. 2.

	Вершина А	Вершина В	Вершина С	Максимальное расстояние от плоскости XOY
1	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	3
2	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	6
3	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	9

Рис. 2. Данные таблицы

Создание столбца-ссылки

Добавление столбца-ссылки на объекты дополнительной таблицы (с помощью команда ALTER TABLE и UPDATE) представлено ниже [11].

--Добавление ссылки на дополнительный столбец

```
ALTER TABLE FIGURES ADD (
    newfigures_ref REF TRIANGLE
);
```

--Обновление таблицы

```
UPDATE FIGURES f SET f.newfigures_ref = (
    SELECT REF(newf) FROM NEWFIGURES newf
    WHERE f.object.A.x = newf.A.x AND f.object.A.y = newf.A.y AND f.object.A.z = newf.A.z
    AND f.object.B.x = newf.B.x AND f.object.B.y = newf.B.y AND f.object.B.z = newf.B.z
    AND f.object.C.x = newf.C.x AND f.object.C.y = newf.C.y AND f.object.C.z = newf.C.z
);
```

Запрос данных через функцию deref() представлен ниже.

```
SELECT
    f.id "ID", f.name "Название фигуры",
    deref(f.newfigures_ref).A "Вершина А",
    deref(f.newfigures_ref).B "Вершина В",
    deref(f.newfigures_ref).C "Вершина С",
    deref(f.newfigures_ref).max_distance_from_plane(TRIANGLE(
    deref(f.newfigures_ref).A, deref(f.newfigures_ref).B, deref(f.newfigures_ref).C))
    "Максимальное расстояние от плоскости XOY"
FROM FIGURES f;
```

Результат запроса изображен на рис. 3.

ID	Название фигуры	Вершина А	Вершина В	Вершина С	Максимальное расстояние от плоскости XOY
1	1 Треугольник № 1	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	3
2	2 Треугольник № 2	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	6
3	3 Треугольник № 3	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	9

Рис. 3. Результат запроса

В данной статье представлено создание и использование объектов в базе данных Oracle. В качестве хранимого объекта выступает треугольник. Для работы с базой данных, в первую очередь, были созданы типы данных Точка (хранит координаты) и Треугольник (хранит вершины треугольника и метод, определяющий максимальное расстояние от плоскости X-O-Y до треугольника).

На основе созданных типов данных получены следующие результаты.

- Определена таблица Figures (содержит атрибуты id (ID треугольника, также

является первичным ключом таблицы), name (название треугольника) и object (содержит данные о вершинах треугольника)), хранящая данные о треугольниках.

- Создана таблица NewFigures, хранящая объекты из первой таблицы.

- Определен столбец-ссылка на объекты таблицы NewFigures. Выполнен запрос данных через функции. deref().

Заключение

Основное отличие объектов от обычных данных заключается в том, что объекты содержат в себе несколько полей. Однако

создание и использование объектов практически не отличается от создания и использования обычной информации в базах данных, поскольку в одном и другом случае используются одинаковые команды (такие как CREATE, ALTER и пр.), что и представлено в данной статье на примере типа данных Треугольник.

Список литературы

1. Медведев Ю.С., Пиотровский Д.Л. Мониторинг производительности базы данных Oracle // Вестник АГУ. 2018. № 4. С. 210–213.
2. Объекты базы данных Oracle. [Электронный ресурс]. URL: https://life-prog.ru/2_34406_ob-ekti-bazi-dannih-Oracle.html (дата обращения: 29.07.2020).
3. Объекты в Oracle. [Электронный ресурс]. URL: <http://citforum.ru/database/oracle/prz/prosto.shtml> (дата обращения: 29.07.2020).
4. Бьюли Алан. Изучаем SQL. СПб., М.: «Символ-плюс», 2016. 309 с.
5. Аллен Дж. Тейлор. SQL для чайников. М.: «Диалектика», 2014. 416 с.
6. Бен Форта. SQL за 10 минут. М.: «Вильямс», 2014. 288 с.
7. Рон Хардман, Майкл МакЛафлин. Oracle Database PL/SQL. Рекомендации эксперта. М.: «Лори», 2014. 450 с.
8. Линн Бейли. Изучаем SQL. СПб.: «Питер», 2012. 592 с.
9. Билл Карвин. Программирование баз данных SQL. Типичные ошибки и их устранение. М.: «Рид Групп», 2012. 336 с.
10. Том Кайт. Издательство: Эффективное проектирование приложений Oracle. М.: «Лори», 2006. 657 с.
11. Дж. Льюис. Oracle. Основы стоимостной оптимизации. СПб.: Питер, 2006. 528 с.

УДК 627.21:656.615

ОЦЕНКА МИРОВОГО ОПЫТА ОСНАЩЕНИЯ И ЭКСПЛУАТАЦИИ ЭКСПОРТНЫХ ПЕРЕГРУЗОЧНЫХ КОМПЛЕКСОВ ПО ПЕРЕВАЛКЕ УГЛЯ В МОРСКИХ ПОРТАХ

Мазуренко О.И.*ФГБОУ ВО «Государственный университет морского и речного флота им. адм. С.О. Макарова»,
Санкт-Петербург, e-mail: mazurenko.olga@gmail.com*

Настоящая статья посвящена анализу современного состояния угольной промышленности и мировых портовых мощностей по перевалке угля. Содержится оценка роли угля в мировой промышленности, а также его доля в структуре потребления энергетических ресурсов, кратко оценены актуальные мировые запасы угля с привязкой по странам. Учитываются доли стран мира в процессе формирования угольного рынка, а также влияние внутренних экономических тенденций стран и регионов на динамику его изменения. В статье приведён экспортно-импортный баланс современного мирового рынка угля, сформулирован перечень основных стран – экспортеров и импортеров данного энергетического ресурса по состоянию на 2018–2019 гг. Далее в статье рассмотрены основные типы портовых перегрузочных комплексов, используемых для перевалки угля в мировых портах, сформулированы тенденции их развития, выявлены основные преимущества и недостатки. Выполнен анализ технического оснащения и технологических процессов для каждого типа портовых перегрузочных комплексов. Сформирован и ранжирован перечень крупнейших мировых портовых угольных терминалов по состоянию на 2019 г. с привязкой к странам, дана краткая характеристика каждого из указанных терминалов.

Ключевые слова: морские порты, угольные терминалы, перегрузочные комплексы

WORLD EXPERIENCE OF EQUIPPING AND OPERATING OF COAL EXPORT TRANSSHIPMENT COMPLEXES

Mazurenko O.I.*Admiral Makarov State University of Maritime and Inland Shipping, St. Petersburg,
e-mail: mazurenko.olga@gmail.com*

This article is devoted to the analysis of the current state of the coal industry and world port coal transshipment capacities. It contains an assessment of the role of coal in world industry, as well as its share in the structure of energy resources consumption, current global coal reserves are briefly estimated. It takes into account the shares of the countries of the world in the process of market formation, as well as the influence of internal economic trends of countries and regions on the dynamics of its change. The article presents the export-import balance of the modern coal market, formulates a list of the main countries of exporters and importers of this energy resource as of 2019. The article considers the main types of transshipment facilities used for transshipment of coal in seaports, formulates their development trends. The analysis of technical equipment and technological processes for each type of transshipment complexes is carried out. A list of the largest coal terminals has been compiled as of 2019 with reference to countries.

Keywords: seaports, coal terminals, transshipment complexes

Несмотря на значительные изменения в экономике XXI в., уголь продолжает играть в ней важную роль. Более половины добытого угля используется для производства электричества и тепла. При этом в странах Западной Европы доля используемого угля в промышленности и в производстве тепла снизилась практически в два раза, тогда как в странах Азии, наоборот, увеличилась на 70%. Вследствие этого вопросы организации перевозок угля между странами-производителями и странами-потребителями не утратили своей актуальности.

Цель исследования: выявление основных направлений мировых грузопотоков угля по состоянию на 2019 г., а также обоснование необходимости применения современных транспортно-технологических схем, основанных на машинах непрерывно-

го действия, для обработки постоянно возрастающих объемов перемещаемого в экспортно-импортном сообщении груза угля.

Материалы и методы исследования

С использованием методов анализа и синтеза в исследовании проводится выявление основных поставщиков и покупателей на мировом рынке угля. На базе анализа статей, пособий для вузов и интернет-источников в исследовании также проводится синтез требований к современным транспортно-технологическим схемам для обработки груза угля.

Результаты исследования и их обсуждение

Общемировое потребление угля по сравнению с уровнем 1970-х гг. увеличи-

лось на 1,7%, при этом в развивающихся странах Азии его потребление для производства тепловой энергии, электричества или для промышленного производства выросла практически на 200%, в частности за счет увеличения доли Индии и Китая. Однако в странах Западной Европы и США доля угля, используемого для производства тепла и электричества, значительно уменьшилась, что видно на рис. 1.

В одних развитых странах отказ от использования угля лёг в основу национальной энергетической политики. К таким странам относятся Великобритания, Канада, Швейцария, Новая Зеландия. В других странах, таких как Австралия, Индия, Китай, США, Германия и Россия, из-за своей доступности и распространённости уголь всё ещё является важной частью энергетики и промышленности.

Общепринятой в мире является концепция отрицательного влияния на экологию от использования угля в энергетике и промышленности. Вследствие этого на мировой рынок угля оказывается постоянное давление, которое вызывает ритмичные циклы его падения и последующего роста. При этом в одних регионах мира, в которые входит Западная Европа и США, цикл падения продолжается с 2010 г., тогда как в странах Юго-Восточной Азии, наоборот, наблюдается цикл небольшого роста.

Основным потребителем угля на мировом рынке все еще остается КНР, хотя, как было сказано выше, и на данном рынке продолжается небольшое падение спроса. Ожидается постепенное снижение доли использования угля в промышленности и энергетике КНР до 20%. При этом должна

увеличиться доля использования возобновляемых источников энергии, а также природного газа [1].

По состоянию на 2019 г. доля угля в балансе энергетических ресурсов составляет 29%. При этом данный вид энергетических ресурсов занимает второе место после нефти. По всем существующим сценариям доля угля будет продолжать падать, пока не достигнет показателя в 20% к 2045 г.

Мировые запасы угля по регионам представлены на рис. 2. Лидером по объему доказанных запасов угля являются США, где доказанные запасы угля составляют 250 млрд т, или 24,5% общемировых. На втором месте по доказанным запасам находится Россия, где разведанные запасы составляют 160 млрд т, или 15,2% мировых. В тройке лидеров также находится Австралия с доказанными разведанными запасами в 147 млрд т.

В Азии лидером по разведанным запасам является Китай, где сосредоточено 13,2% мировых запасов угля, или 138 млрд т, далее следуют Индия с запасами 100 млрд т, Индонезия с запасами 37 млрд т и Турция с запасами 11 млрд т. Значительные запасы угля также сконцентрированы в Германии, в Польше, в Украине и в Казахстане.

При этом по соотношению запасов угля и его добычи абсолютными лидерами являются Российская Федерация и страны Содружества независимых государств. Прогнозируемое истощение запасов при сохранении уровня добычи ожидается не ранее чем через 390 лет. При этом истощение запасов угля в Соединенных Штатах ожидается ранее — через 300 лет, тогда как в азиатских странах угля хватит всего на 80 лет.

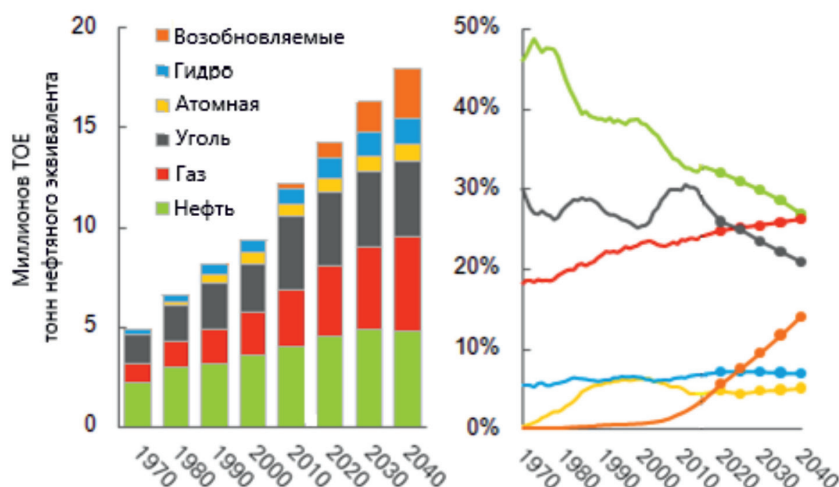


Рис. 1. Доля видов топлива в мировой энергетике (статистика и прогноз)



Рис. 2. Мировые запасы угля по регионам, % (2019)

Таким образом, можно сделать вывод, что запасы угля в мире являются значительными, а лидерами по добыче и экспорту данного полезного ископаемого на протяжении ближайших лет будут являться страны СНГ, Россия, США, Австралия, Бразилия, Индонезия и Турция.

Что касается добычи угля, то в 2018 г. мировой уровень добычи вырос по отношению к 2017 г. на 4,3%. Рост за период с 2007 по 2017 г. составил 1,3%. Уровень добычи угля в 2017 г. составил 3755 млн т, в 2018 г. – 3916 млн т, а в 2019 г. составил 3899 млн т.

КНР остаётся основным производителем энергетического угля в мире. В 2018 г. добыча составила 1 828 млн т в год, или 46,7% от общемирового объема. При этом этот показатель выше уровня 2017 г. на 4,7%.

Второе место занимают США с объемом 364,5 млн т, что на 2% меньше уровня 2017 г. и на 4% меньше уровня 2007 г. Доля США на общемировом рынке составляет почти 10%. На третьем месте по уровню добычи находится Индонезия, где в 2018 г. было добыто 323 млн т угля, что на 19% выше уровня 2017 г. и на 8% выше уровня 2007 г. Четвёртое место занимает Индия с уровнем добычи 308 млн т угля в год, что составляет 7,8% рынка. В 2018 г. в Индии было добыто на 7,5% больше угля, чем в 2017 г., и на 3% больше, чем в 2007 г. На четвёртом месте находится Австралия с уровнем добычи 301 млн т, которая контролирует 7,7% рынка угля. Всего в Азии и Тихоокеанском бассейне добывается сегодня 2853 млн т угля в год, что составляет 73% от всего добытого объема.

На шестом месте находится Россия с уровнем добычи в 2018 г. 220 млн т, что составляет 6% рынка [2]. При этом уровень добычи вырос на 7% по сравнению с 2017 г. и на 3,7% по сравнению с 2007 г. Также мировыми лидерами по добыче угля являются: Колумбия (с объемом 58 млн т в год), Польша (с объемом 48 млн т в год), Казахстан

(с объемом 50 млн т в год), Южная Африка (с объемом 145 млн т в год), Монголия (с объемом 35 млн т в год) и Канада (с объемом 29 млн т в год). Экспорт всех видов угля в мире в 2018 г. вырос на 60% по сравнению с уровнем 2017 г. и составил 858,8 млн т. Ведущим экспортером угля на сегодняшний день остается Австралия с объемом 249 млн т в год, что составляет примерно 29% от общемирового экспортного уровня. При этом Австралия нарастила в 2018 г. уровень экспорта на 7,4% по сравнению с уровнем 2017 г. и на целых 4,3% по сравнению с уровнем 2007 г. На втором месте по уровню экспорта находится Индонезия с долей рынка 25,7%, или 220 млн т в год. Экспорт угля из Индонезии увеличился почти на 15% по сравнению с уровнем 2017 г. Россия находится на третьей позиции мирового угольного экспортного рынка. Всего в 2018 г. Россией было направлено на экспорт 136 млн т угля, что составляет 15% рынка. Рост объемов составил 13% по сравнению с уровнем 2017 г. и 6% по сравнению с уровнем 2007 г. В десятку лидеров по объемам экспорта угля также входят США (было экспортировано 66 млн т угля), Колумбия (46 млн т), Южная Африка (50 млн т), Монголия (23 млн т). Большую часть экспорта этих стран составляет энергетический уголь. Крупнейшие импортеры угля – страны Западной Европы с уровнем 150 млн т по итогам 2018 г., что на 7% больше уровня 2017 г., но на 5% меньше уровня 2007 г.

Далее следует Китай (146 млн т в год), Индия (141 млн т в год), Япония (120 млн т в год), Южная Корея (92 млн т в год). Значительную долю на рынке импорта угля занимают Канада, Мексика, Соединенные Штаты, страны Африки и Тихоокеанского бассейна. Основными поставщиками угля в страны Азии являются Индонезия, Австралия, Россия и ЮАР. Основные поставщики угля страны Западной Европы – Россия, Австралия, США и Колумбия. Таким образом, мировой рынок угля

в 2018–2019 гг. находился в относительно стабильном и благоприятном состоянии. Однако какие результаты рынок покажет в 2020 г., пока неизвестно.

По состоянию на 2018 г. общий объем экспортно-импортных перевозок угля в мире составил 858 млн т. При этом 90% перевозок угля между странами осуществляется с использованием морского транспорта.

В этой связи перед морскими портами, как пунктами перевалки между видами транспорта, стоит амбициозная задача – справляться с постоянно возрастающими объемами на перевалке угля, при этом не теряя в качестве и скорости грузовых работ. Именно поэтому перед транспортной наукой и машиностроением стоит стратегическая задача создания и имплементации работы новых мощных технических средств, а также организации работы и внедрение технологических схем, рассчитанных на выполнение грузовых работ в портах с интенсивностью 5–10 тыс. т/час.

Для организации грузовых работ по перевалке угля в морских портах сегодня используются два основных типа перегрузочных комплексов:

- универсальные (использующие крановую/грейферную и малую механизацию);
- специализированные (использующие конвейеры, стакеры/реклаймеры, вагоноопрокидыватели и судопогрузочные машины).

Универсальные перегрузочные комплексы исторически являются более распространенными. Ведь с помощью портальных кранов и крановых захватов – ковшей грейферов можно обрабатывать достаточно широкую номенклатуру навалочных грузов. Основным достоинством использования крановой (грейферной) технологии является её обратимость, т.е. возможность осуществлять как экспортные, так и импортные операции (погрузку/разгрузку судов). Данный тип перегрузочных комплексов также характеризуется относительно низ-

кой себестоимостью и сравнительно высокими показателями производительности. В состав механизации универсальных портовых перегрузочных комплексов, использующих крановую технологию, включают следующие машины:

- трюмной механизации;
- причальной механизации;
- тыловой механизации;
- вагонной механизации.

В зависимости от объема перевалки на причалах универсального терминала может использоваться 1 или 2 линии портальных кранов, как показано на рис. 3. Установка двух линий портальных кранов позволяет увеличивать вместимость складов, при этом тыловые краны дают возможность причальным кранам сконцентрироваться исключительно на обработке судов. Для еще большего увеличения объемов тыловых вкладов в данной технологии используются многочисленная малая механизация – бульдозеры [3].

Значительную долю работы в подобных перегрузочных комплексах выполняют вспомогательные машины, которые выполняют функции по восстановлению и поддержанию сыпучести угля, это показано на рис. 4.

Важную роль выполняют рыхлительные и заборные устройства, системы локализации и обеспыливания, а также малая механизация по зачистке трюмов и кузовов вагонов. Грузоподъемность портальных кранов на универсальных перегрузочных комплексах ограничивают 10 тоннами. Это связано с тем, что при разгрузке железнодорожных вагонов использование кранов грузоподъемностью > 10 т (12–16 т) приводит к сильному повреждению кузовов железнодорожных вагонов, тогда как применение кранов и грейферов с более низкой грузоподъемностью ведёт к значительному сокращению производительности работы всего перегрузочного комплекса [4].

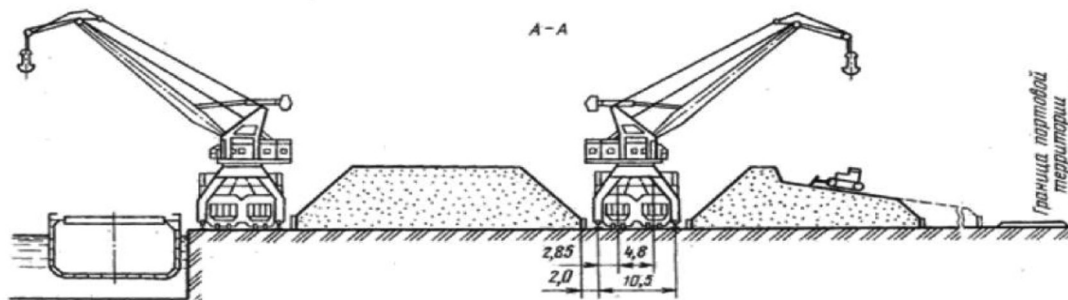


Рис. 3. Крановая схема механизации угольного терминала с двумя линиями кранов и использованием тыловых бульдозеров

Общими лимитирующими факторами в работе универсальных перегрузочных комплексов являются:

- скорость и интенсивность работы портовых кранов;
- грузоподъемность кранов и вместительность ковшей грейферов;
- ограничения по количеству кранов на причальной линии с учетом рабочей зоны каждого крана.

Ограничен и дедейт судов, которые возможно обрабатывать на универсальном передаточном комплексе: < 25000 т. Универсальные перегрузочные комплексы, использующие крановую (грейферную) технологию, значительно уступают специализированным комплексам по производительности. На них невозможно осуществлять автоматическую очистку и дробление фракции угля, применять активное пылеподавление. Эти недостатки не только значительно ухудшают качество выполнения портовых перегрузочных работ с углем, но и наносят значительный вред экологии. Работа порталных кранов с грейферными захватами в портах сильно изнашивает железнодорожный подвижной состав, выполняющий практически 100% сухопутных перевозок угля, так как стенки полувагонов сильно повреждаются от ударов вследствие такого рода разгрузки. Относительно низкое качество разгрузочных работ с применением грейферных захватов требует привлечения многочисленной дополнительной техники – подгребающих бульдозеров и ковшовых экскаваторов для полной разгрузки и очистки железнодорожных вагонов с углем [5]. Таким образом, использование портовых перегрузочных комплексов с порталными кранами и грейферами оправдано при годовом грузообороте до 1,5–3 млн т.

Однако в связи с постоянно возрастающими объемами морских перевозок угля во всем мире подобные универсальные перегрузочные комплексы постепенно уходят в прошлое и заменяются специализированными высокотехнологичными угольными терминалами [6]. Специализированные угольные терминалы рассчитаны на грузооборот более 3 млн т/год и основаны на использовании машин непрерывного действия.

В качестве устройства для разгрузки железнодорожных вагонов используются роторные вагоноопрокидыватели (3), для загрузки морских судов – конвейерные судопогрузочные машины (8), для работы со штабелями груза на складе, загрузки склада с железной дороги и отгрузки с него груза на судопогрузочные машины используют конвейерные стакеры и реклаймеры (2). Подобная схема механизации обеспечивает

возможности перегрузки угля по различным технологическим вариантам. По прямому технологическому варианту уголь из роторного вагоноопрокидывателя (3) через приемный бункер (4) поступает на терминальную сеть ленточных конвейеров (1, 5, 9, 11), проходит через узлы пробоотборника, дробилки и очистки (6), а затем судопогрузочными машинами подается в трюмы судна-балкера. При работе по технологическому варианту «склад – судно» уголь посредством реклаймера (2) подается на складской ленточный конвейер (10), а затем по системе конвейеров (1, 7, 9) подается на судопогрузочные машины (8). При работе по технологическому варианту «вагон – склад» уголь от роторного вагоноопрокидывателя (3) через приемный бункер (4) по сети ленточных конвейеров (5, 10, 11) поступает на стакер (2), который формирует складской штабель угля. Данная схема приведена на рис. 5.

На специализированных перегрузочных комплексах для обработки угля чаще всего применяются вагоноопрокидыватели роторного типа, которые обеспечивают необходимую высокую производительность, надежность и степень автоматизации. Роторные вагоноопрокидыватели совершают порядка 30 циклов в час при обработке железнодорожных вагонов грузоподъемностью 60–95 тонн и 25 циклов в час при обработке железнодорожных вагонов грузоподъемностью 100–125 т. Производительность вагоноопрокидывателей в таких случаях составляет 1800 и 3100 т/ч соответственно. Основным недостатком подобного типа вагоноопрокидывателя является необходимость значительного заглубления приемного бункера (4). Реже используются боковые вагоноопрокидыватели, которые могут обеспечить 25 и 20 циклов в час при расчетной производительности 1500 и 1860 т/ч соответственно [7].

В странах с холодным климатом на угольных терминалах дополнительно устанавливаются вагоноразмораживатели – специализированные сооружения ангарного типа, оборудованные системой котельных с паровым отоплением и алюминиевыми отражающими панелями на стенах. В них размещается 20–80 вагонов, которые стоят 20–40 мин при температуре 40–90 °С. Эта операция предназначена для того, чтобы примерзший уголь отстал от стенок вагонов и мог быть выгружен в вагоноопрокидывателе. Вагоноразмораживатели часто оборудуют предварительными рыхлительными установками с 3–10 шнеками, которые разрыхляют замерзший уголь в кузовах вагонов до их разогрева. Весь цикл показан на рис. 6.

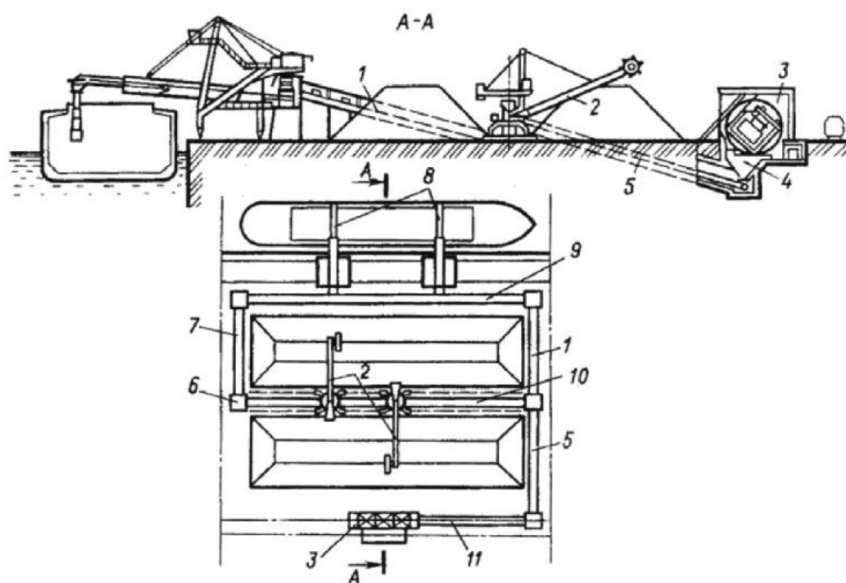


Рис. 5. Специализированный перегрузочный комплекс с приемом угля из железнодорожных вагонов через роторный вагоноопрокидыватель

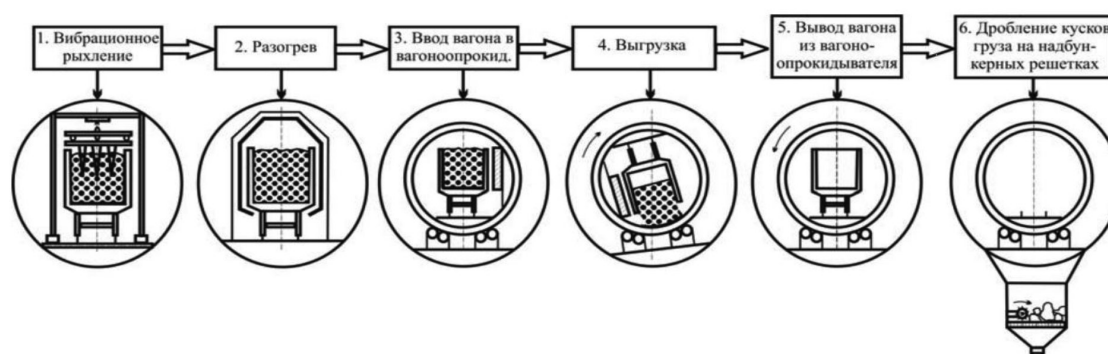


Рис. 6. Технологическая схема выгрузки вагона с углем на специализированном угольном терминале в холодное время года [8]

Вагоноопрокидыватели обычно устанавливаются в герметично закрытые здания и оборудуются дополнительными циклонными системами очистки воздуха и системами активного пылеподавления. Помимо этого, здания вагоноопрокидывателей оборудуются ветрозащитными сооружениями с автоматическими воротами, которые исключают распространение пыли, а также снижают уровень шума от выгрузки.

Безопасной транспортировке выгруженного из вагонов угля по системе магистральных транспортных конвейеров уделяют особое внимание – конвейерные системы защищают специализированной боковой

и верхней дуговой защитой от ветра, а пересыпанные станции и узлы оборудованы мощными вытяжками, воздух из которых поступает в фильтрационные системы, где проходит тщательную очистку [9].

Для обеспечения отгрузки на морские суда угля необходимого качества и необходимой фракции на конвейерной линии терминалов устанавливаются дробильно-сортировочные комплексы, которые позволяют отсеивать крупные фракции угля и затем измельчать их до нужного размера. Обычно подобные дробильно-сортировочные комплексы устанавливаются на участках транспортных конвейеров и совмеща-

ются с комплексами магнитной очистки от металла. Магнитная очистка необходима для удаления из партий груза мусора – металла с помощью переменного магнита.

Работу непосредственно на угольных складах (формирование и расформирование штабелей груза) осуществляют при помощи специализированных высокопроизводительных машин – стакеров и реклаймеров, показанных на рис. 7. Линейный стакер позволяет принимать уголь с подающего конвейера и формируется из него штабель. По сути, стакер является укладчиком груза. Реклаймер же выполняет обратную функцию – забирает уголь со штабеля для подачи на причальный конвейер и судопогрузочную машину. Существуют также комбинированные стакеры-реклаймеры, на роторных машинах которых используется реверсивный ленточный конвейер. Они могут последовательно как формировать штабель угля, так и расформировывать его. Максимальная производительность работы стакера/реклаймера составляет до 5000 т/ч, вылет стрелы – до 50 м. Конвейеры на стакерах и реклаймерах также закрывают боковой ветрозащитой, а приемные бункеры дополнительно накрывают плотными герметичными кожухами [10].

Стакеры и реклаймеры на угольных складах комбинируют с активной системой круглогодичного автоматизированного пылеподавления: в теплое время года ведется водное орошение складов из промышленных пульверизаторов, в холодное время года применяются снегогенераторы.

Для исключения вредного воздействия на окружающую среду на складах современных угольных терминалов в обязательном порядке применяют дополнительные системы аспирации и очистки воздуха, системы многоуровневой фильтрации и нанофлотации сточных и ливневых вод.

Лимитирующим фактором производительности специализированных и угольных терминалов является производительность морского грузового фронта и, в частности, судопогрузочных машин терминала [11].

На рынке существует достаточно широкий ассортимент судопогрузочных машин от различных производителей, которые отличаются как по производительности, так и по конструктивным особенностям. На специализированных угольных терминалах чаще всего используются машины с переменным вылетом, телескопическими транспортерами, а также с метательной машиной для равномерной загрузки пространства трюмов. Трюмы судов-балкеров, как правило, обрабатываются равномерно по слоям, при этом осуществляя поворот стрелы судопогрузочной машины с её постоянным поворотом вокруг своей оси и поднятием. Подобное высокотехнологичное и высокопроизводительное современное перевалочное оборудование устанавливается на специализированных терминалах для обеспечения 100 % автоматизации технологических процессов перевалки. Подобный уровень автоматизации процессов позволяет добиться впечатляющих объемов перевалки грузов [12].



Рис. 7. Комбинированный стакер-реклаймер на угольном складе

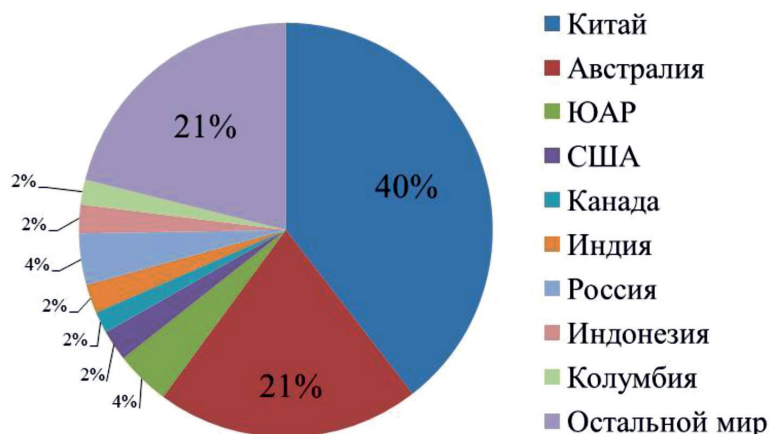


Рис. 8. Распределение портовых мощностей по перевалке угля в мире, 2018 [13]

Крупнейшие терминалы по перевалке угля (> 15 млн т/год) [15]

№ п/п	Страна	Порт	Терминал	Грузооборот в 2018 г., млн т
1	Китай	Циньхуандао	Qinhuangdao Coal Terminal Co Ltd	246,55
2	Китай	Хуанхуа	Huanghua Port Coal Terminal	208,5
3	Китай	Таншань	Caofeidian Harbour Coal Terminal	152
4	Австралия	Ньюкастл	PWCS Port Waratah Coal Terminals	107,2
5	Австралия	Гладстон	RG Tanna Coal Terminal	97,7
6	ЮАР	Ричардс Бэй	Richards Bay Coal Port	73,5
7	Китай	Тяньцзинь	Tianjin Port Holdings Co Ltd	70,92
8	Австралия	Хей Поинт	Dalrymple Bay Coal Terminal	63,01
9	Австралия	Хей Поинт	Hay Point Coal Terminal	43,4
10	США	Норфолк	Lamberts Point Coal Terminal	41,2
11	Индия	Мундра	Mundra Coal Port	38,6
12	Колумбия	Пуэрто-Болivar	Puerto Bolivar Coal Terminal	33,2
13	Канада	Ванкувер	Roberts Bank Coal Terminal	29,03
14	Австралия	Эбот Поинт	Abbot Point Coal Terminal	26,2
15	Россия	Усть-Луга	АО «Ростерминалуголь»	24,5
16	Россия	Находка	АО «Восточный Порт»	24,2
17	Индонезия	Сангкиман	Tanjung Bara Coal Terminal	23,2
18	Россия	Ванино	АО «Дальтрансуголь2»	20,1
19	Австралия	Вуллонгонг	Port Kembla Coal Terminal	15,8
20	Индонезия	Баликпапан	Balikpapan Coal Terminal	15,2

Как видно из рис. 8, подавляющее большинство наиболее крупных терминалов находятся в двух странах: в Китае и в Австралии. На их долю приходится порядка 45% общемировых мощностей. При этом 9 наиболее крупных мировых экспортеров/импортеров угля: Китай, Австралия, ЮАР, США, Канада, Индия, Россия, Индонезия и Колумбия –владеют 79% портовых мощностей по его перевалке [14]. На все остальные страны мира приходится всего 21%.

В таблице представлен перечень крупнейших мировых портов и терминалов по перевалке угля.

Выводы

Универсальные перегрузочные комплексы, используемые при перевалке угля, постепенно уходят в прошлое. Повсеместно они заменяются высокопроизводительными специализированными перевалочными комплексами, использующими новейшие

технологии для организации грузовых работ, соблюдения приемлемого уровня экологических и санитарных норм, а также обеспечения перевалки всё возрастающих объемов угля на рынке [16].

Список литературы

1. British Petroleum Statistical Review of World Energy. № 68, 2019. June. [Electronic resource]. URL: <https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2019-full-report.pdf> (date of access: 18.06.2020).
2. Плакиткин Ю.А. Угольная промышленность России на мировом рынке угля: тенденции перспективного развития // Уголь. 2016. № 7. С. 12–16.
3. Степанов А.Л. Перегрузочное оборудование портов и транспортных терминалов. Учебник для ВУЗов. СПб.: Политехника, 2013. 427 с.
4. Как отгружают уголь в морских портах. [Электронный ресурс]. URL: <https://kak-eto-sdelano.ru/kak-otgruzhayut-ugol-v-morskih-portah/> (дата обращения: 19.07.2020).
5. Экономические и экологические проблемы развития российских угольных терминалов: аналит. доклад, апр. 2018 / Институт проблем естественных монополий. М.: ИПЕМ, 2018. 81 с.
6. Ломакина Н.С. Технология перегрузочных работ на угольном терминале // Современные наукоемкие технологии. 2014. № 5. С. 134–136.
7. Ключевые параметры крупнейших специализированных угольных портовых терминалов в мире [Электронный ресурс]. URL: <https://expert.ru/ratings/klyuchevyye-parametryi-krupnejshih-spetsializirovannyih-ugolnyih-portovyih-terminalov-v-mire/> (дата обращения: 19.07.2020).
8. Инвестиционный проект «Новый угольный терминал» [Электронный ресурс]. URL: <https://en.ppt-online.org/25840> (дата обращения: 19.07.2020).
9. Без шума и пыли: мировые технологии перевалки угля [Электронный ресурс]. URL: <https://rg.ru/2017/09/05/reg-dfo/nailuchshie-mirovye-tehnologii-perevalki-uglia-uzhe-primenaiutsia-v-rossii.html> (дата обращения: 19.07.2020).
10. Как переваливают уголь: репортаж из самого большого угольного порта страны [Электронный ресурс]. URL: <https://www.popmech.ru/technologies/265442-vostochnyy-port-kak-perevalivayut-ugol/> (дата обращения: 19.07.2020).
11. Морпорт: Перспективы развития специализированных терминалов: обеспечение ритмичной переработки грузов в портах и развитие портовых мощностей [Электронный ресурс]. URL: <https://morproekt.ru/articles/prezentatsii/1017-perspektivy-razvitiya-spetsializirovannykh-terminalov-obespechenie-ritmichnoj-pererabotki-gruzov-v-portakh-i-razvitie-portovykh-moshchnostej> (дата обращения: 19.07.2020).
12. Стратегия развития морской портовой инфраструктуры России до 2030 года/ одобрена на совещании членов Морской коллегии при Правительстве Российской Федерации 28.09.12 [Электронный ресурс]. URL: <http://www.rosmorport.ru/media/File/strategy.pdf> (дата обращения: 19.07.2020).
13. Курс на специализацию: ТОП-10 крупнейших угольных терминалов мира [Электронный ресурс]. URL: https://cfts.org.ua/articles/top_10_ugolnykh_terminalov_mira_685/67489 (дата обращения: 19.07.2020).
14. Купцов Н.В. Анализ конъюнктуры и тенденций мирового грузооборота угольных грузов и прочих энергоносителей с учетом роли Российской Федерации // Научное обозрение. 2015. №19. С. 408–416.
15. Global Energy Observation: Current List of Coal Ports Transmission [Электронный ресурс]. URL: http://globalenergyobservatory.org/list.php?db=Transmission&type=Coal_Ports (дата обращения: 19.07.2020).
16. Кондратьев В.Б., Попов В.В. Глобальный рынок угля: состояние и перспективы // Горная промышленность. 2019. № 2 (144). С. 6–12.

УДК 656.612

ОСНОВНЫЕ ПРИНЦИПЫ УПРАВЛЕНИЯ РАБОТОЙ ФЛОТА МОРСКОЙ СУДОХОДНОЙ КОМПАНИИ В СОВРЕМЕННЫХ РЫНОЧНЫХ УСЛОВИЯХ

Киринос Д.А.*ФГБОУ ВО «ГУМРФ им. адм. С.О. Макарова», Санкт-Петербург, e-mail: kir.denis@yahoo.com*

В статье определено, что результативность деятельности судоходных компаний напрямую зависит от эффективности системы управления работой судов. Для понимания закономерностей отечественной научной мысли в области управления работой флота проведен ретроспективный анализ. Выяснилось, что основной задачей флота в СССР являлось обеспечение потребностей народного хозяйства посредством предоставления услуги по морской перевозке товаров промышленности. Устойчивость работы флота в плановой экономической системе обеспечивалась постоянством количества судов и пароходств, а также периодичностью возникновения и постоянным размером заявок на перевозку. После перехода к рыночной экономике советская система управления работой флота оказалась недееспособной. В рыночной экономике связи между функциональными элементами транспортной системы начали устанавливаться на основе взаимных договоров и соглашений, поэтому пароходства были вынуждены перейти к экономическому планированию и управлению на основе моделирования бизнес-процессов. Далее в статье выявлены основные проблемы, с которыми сегодня сталкиваются отечественные судоходные компании, работающие в рыночных условиях. Автором сформулировано понятие управления работой флота в рыночных условиях, определен объект управления, задачи управления и функция управления. Выяснено, что мировой опыт управления работой морского флота свидетельствует о необходимости использования автоматизации и бизнес-моделирования с применением параметризуемой входной и выходной информации. Автором предложены базовые принципы работы пароходств в рыночных условиях и доказана необходимость внедрения автоматизированной системы, выполняющей аналитическую работу и необходимые вычисления. Установлено, что одним из наиболее эффективных инструментов для создания, апробации и тестирования подобной электронной автоматизированной системы является имитационное моделирование.

Ключевые слова: морской флот, управление предприятием, судоходство

BASIC PRINCIPLES OF FLEET MANAGEMENT OF A MARITIME SHIPPING COMPANY IN MODERN MARKET CONDITIONS

Kirnosov D.A.*Admiral Makarov State University of Maritime and Inland Shipping,
St. Petersburg, e-mail: kir.denis@yahoo.com*

It is determined in the article that the performance of shipping companies directly depends on the effectiveness of the ship management system. To understand the patterns of Russian scientific thought in the field of fleet management, a retrospective analysis was carried out. It turned out that the main task of the fleet in the USSR was to meet the needs of the national economy by providing services for the sea transportation of industrial goods. The stability of the fleet in the planned economy of the system was ensured by the constancy of the number of ships and shipping companies, as well as the frequency of occurrence and the constant size of requests for transportation. After the transition to a market economy, the Soviet fleet management system turned out to be incompetent. In a market economy, connections between the functional elements of the transport system began to be established on the basis of mutual contracts and agreements, so shipping companies were forced to switch to economic planning and management based on modeling business processes. Further, the article identifies the main problems faced by domestic shipping companies operating in market conditions today. The author has formulated the concept of fleet management in market conditions, defined the management object, management tasks and management function. It was found that the world experience in managing the work of the marine fleet indicates the need to use automation and business modeling with the use of parameterized input and output information. The author proposes the basic principles of the operation of shipping companies in market conditions and proves the need for the introduction of an automated system that performs analytical work and the necessary calculations. It was found that one of the most effective tools for creating, approbation and testing of such an electronic automated system is simulation.

Keywords: sea fleet, business management, shipping

Общая эффективность деятельности морского флота напрямую зависит от качества работы отдельных функциональных единиц – судоходных компаний в разных странах мира. В свою очередь, результативность деятельности судоходных компаний непосредственно зависит от наличия налаженной и четко выстроенной системы планирования, организации и управления

работой транспортных элементов – судов различного класса и вместимости.

Оперативное управление работой морского флота является сложным процессом, в основе которого лежат элементы работы систем массового обслуживания, элементы бизнес-моделирования, логистики и транспортной диспетчеризации. В условиях рыночной экономики в судоходных компаниях

организованное и налаженное управление работой флота позволяет добиться максимальных показателей прибыли и обеспечить баланс доходов и затрат. На сегодняшний день публикуется небольшое количество научных работ, в которых рассматриваются вопросы управления работой флота.

Этот факт позволяет сделать вывод о необходимости проведения дополнительных научных исследований в данной сфере, в особенности в рамках морского судоходства в Российской Федерации.

В отечественной транспортной науке основные подходы и принципы управления морским флотом были сформированы в середине XX века для условий плановой экономики. После распада СССР, перехода на рыночную экономику и образования частных судоходных компаний появилась необходимость в ревизии принципов управления флотом. Соответственно, перед исследованием была поставлена следующая цель: на базе имеющихся научных работ в данной области и практического опыта частных пароходств сформировать основные принципы управления работой морского флота, использование которых позволит добиваться поставленных в компаниях экономических результатов в рыночных условиях. При этом автором было принято решение также рассмотреть возможность внедрения сформированных принципов в основу автоматизированных электронных управленческих систем.

Материалы и методы исследования

В основу исследования положен анализ материалов и публикаций по теме управления работой морского флота, изданных в отечественных печатных изданиях в период с 2013 по 2019 год. Также использованы материалы о практической деятельности 26 российских судоходных компаний и результаты эксплуатации их тоннажа в период с 2015 по 2019 год. Предлагаемые автором принципы и выводы синтезированы на основе указанных материалов.

Результаты исследования и их обсуждение

Для понимания закономерностей и более точной оценки современного состояния отечественной научной и практической мысли в области организации оперативного управления работой морского флота автором, прежде всего, был проведен ретроспективный анализ. По результатам ретроспективного анализа установлено следующее.

Подход к управлению работой морского флота в Советском Союзе был окончательно сформирован к середине XX века.

Основной задачей морского флота в социалистической России являлось обеспечение потребностей народного хозяйства посредством предоставления услуги по морской перевозке товаров промышленности. Обеспечение данной потребности достигалось за счет целенаправленного и точного распределения ресурса – вместимости определенных судов, работающих на установленных маршрутах перевозок, для выполнения заявок морских портов. В рамках советской хозяйственной деятельности выполнимость данной задачи достигалась за счет обеспечения согласованной деятельности грузоотправителей/грузополучателей, портов и пароходств.

Советская система управления работой флота строилось на базе 4 основных элементов: рейсового планирования и управления, планирования и управления в рамках навигационного периода, месячного планирования и управления, а также декадного планирования и управления. Данный подход представляет собой упрощение и разбиение поставленной советской экономикой транспортной задачи на небольшие элементы с учетом возрастания их сложности [1].

Устойчивость работы морского флота в подобной системе обеспечивалась благодаря постоянству и предсказуемости характеристик и количества работающих судов и пароходств, а также благодаря периодичности возникновения и постоянным размерам заявок портов на перевозку. В подобных условиях не требовалось строгое согласование всех элементов в рамках транспортной системы, а необходимость в автоматизации оперативного управления практически отсутствовала за счет возможности осуществлять управление работой флота и портов в ручном режиме.

В результате этого задача организации управления работой флота в автоматизированном режиме в советское время так и осталась нерешенной. Не существовало и электронной информационной системы, позволяющей собирать информацию о местоположении судов пароходств в определенные моменты времени. Сказалась сложность и дороговизна внедрения автоматизированного управления работой флота в существовавшие механизмы деятельности пароходств.

Таким образом, отсутствие в советской науке и судоходной практике необходимых механизмов автоматизации оперативного управления работой морского флота было вызвано ограничениями управленческих подходов к выстраиванию транспортной отрасли, а также уровнем научно-технического прогресса того времени.

Тем не менее к концу 80-х годов XX века были сформированы основные научные принципы, позволяющие повысить точность планирования и управления флотом, но из-за отсутствия необходимости обеспечения быстрой и согласованной работы большого количества элементов транспортной системы эта задача так и не получила своего окончательного решения [1].

Резкие изменения произошли с распадом Советского Союза. К 1995 году советская система управления и планирования работы флота оказалось абсолютно недееспособной в связи с переходом к рыночной экономике, распадом устоявшихся транспортных связей и переходом парокходств и морских портов в частные руки. Связи между грузоотправителями и грузополучателями, морскими портами и судоходными компаниями оказались полностью разрушенными, а прежняя система, действовавшая на базе четырех элементов планирования и управления, оказалось не применима.

В складывающейся отечественной рыночной экономике связи между функциональными элементами транспортной системы начали устанавливаться на основе взаимных договоров и соглашений. Потеря возможности осуществлять централизованное управление негативно повлияла и на возможности совершенствования работы морского флота.

Распад СССР позволил отечественным морским парокходствам выйти на международный уровень, где они столкнулись с необходимостью решения достаточно сложных задач и вынуждены были перейти от долгосрочного планирования к экономическому планированию и оперативному управлению на основе моделирования бизнес-процессов. Работа с флотом стала строиться на основе бизнес-планов, корректировка которых производится ежедневно на протяжении всего навигационного периода. Возникновение необходимости обеспечения быстрой и согласованной работы большого количества судов вызвало постепенное уменьшение количества работающего флота и усложнило задачу выбора направлений и грузов для перевозок [2].

Отечественные морские парокходства в основном распались на значительное количество мелких компаний, имеющих в своем распоряжении < 10 судов каждая, которые принципиально отказались от долгосрочного планирования и перешли к оперативному рейсовому управлению. Оставшиеся крупные компании (с количеством судов > 10 единиц) испытывали значительные сложности с перестроением системы управления флотом. Решением накопившихся проблем

для них стало разделение имеющегося парка судов на группы в зависимости от типа-размера и внедрение отдельного локального управления каждой группой.

Таким образом, в начале 2000-х годов уровень и качество управления работой флота вернулись к уровню 20-х годов XX века. Это вызвало резкое снижение доходов и постепенное списание имеющегося тоннажа на металлолом, а затем и уход большинства отечественных судоходных компаний с международного транспортного рынка. Необходимость значительных конструктивных изменений системы управления флотом потребовала от отечественных судовладельцев использования международного опыта.

Далее автором выявлены основные проблемы, с которыми сталкиваются отечественные судоходные компании, работающие в рыночных условиях. К ним относятся:

- динамично изменяющиеся условия рынка, которые выражаются как в появлении новых товаропотоков, так и в изменении его направлений и условий уже имеющих. На появление нового грузопотока судоходная компания должна реагировать путём поиска и предложения новых судов для его обслуживания (как из имеющегося собственного резерва, так и путем дополнительного фрахтования на стороне). В условиях динамично изменяющегося рынка и его сложной конъюнктуры в относительно выигрышных условиях находятся крупные судовладельцы, для которых уровень ставок играет меньшую роль, а резерв судов у них больше;

- выбор судна определенного типа-размера для обслуживания конкретного грузопотока. Судовладельцам необходимо подстраиваться под требования нормативных документов стран, между которыми осуществляются морские перевозки. При этом возраст судна и его характеристики должны позволять отработать чартер без проведения текущего и планового ремонта. В крупных парокходствах подобный выбор осуществляется только после проведения значительного объема аналитической работы и вычислений, которые целесообразно выполнять в автоматизированном режиме;

- необходимость постоянно учитывать входящую информацию касательно погодных условий на маршрутах движения судов компании, а также необходимость оценивать политическую и экономическую ситуации в странах, где находятся порт отправления и назначения. Также судоходным компаниям необходимо следить за вступлением в силу нормативных и правовых актов ИМО в области морского судоходства;

– необходимость учитывать изменения и отклонения исполнения чартеров от планов, особенно в случае работы нескольких судов одного типоразмера на конкретном плече для перевозки одного типа груза. У пароходств возникает необходимость осуществлять индивидуальное оперативное управление рейсами каждого судна из числа работающих, в зависимости от степени выполнения плана перевозок другими судами на данном маршруте.

Далее автором сформулировано понятие управления работой флота.

Управление работой морского флота – это вид предпринимательской деятельности, направленный на координацию функционирования морских судов, находящихся в собственности или в чартере у судоходной компании, направленный на обеспечение удовлетворения потребности клиентов в услуге морской перевозки грузов и/или пассажиров и выполняемый с целью извлечения прибыли.

В данной парадигме объектом управления является персонал пароходства и суда, находящиеся в собственности или в чартере у судоходной компании. Функция управления состоит во всестороннем обеспечении процесса извлечения прибыли от удовлетворения потребности рынка в услуге морской перевозки грузов и/или пассажиров посредством максимально эффективного использования трудовых и технических ресурсов. Достижение цели возможно только посредством выстраивания иерархической структуры, имеющей функционирующие горизонтальные и вертикальные связи.

Автором сформулированы основные задачи управления работой морского флота.

1. Расстановка флота компании на разных маршрутах работы с расчетом на максимальное использование его провозной способности и с осуществлением краткосрочного и долгосрочного общего планирования его работы.

2. Индивидуальное планирование работы каждого судна на каждом конкретном маршруте перевозок, рутинное формирование индивидуальных рейсовых заданий судов.

3. Составление общего и индивидуального расписания работы судов компании, осуществление смены экипажей, проведение текущего и планового ремонта.

4. Ежедневный мониторинг работы судов флота для целей минимизации простоев.

5. Проведение оценки результативности работы флота компании, контроль исполнения плана перевозок.

6. Повышение эффективности работы судов в чартерах путем поиска вариантов

снижения эксплуатационных затрат (достижение максимальной эффективности работы каждого конкретного суда с учетом обеспечения безопасности мореплавания, условий работы моряков и сохранности окружающей среды).

7. Составление единой тарифной сетки, формулировка правил преференций, определение возможных скидок и надбавок.

8. Ведение деловой переписки с капитанами судов флота, организация стабильной работы с брокерскими и агентскими компаниями, терминалами морских портов, грузотправителями и грузополучателями.

9. Проведение маркетинговых исследований на рынке международных морских перевозок, поиск клиентов и вариантов выхода на иные национальные и международные рынки.

10. Оценка международного положительного опыта работы других пароходств, его обобщение и внедрение в деятельность компании.

Автором установлено, что высокая степень адекватности управления и планирования может достигаться путем применения различных приемов. В частности, путем использования так называемых коэффициентов резерва – $k_{рез}$. Однако высокая динамика транспортных процессов зачастую порождает сложные математические зависимости, которые практически невозможно точно спрогнозировать и реализовать на практике.

Мировой опыт управления работой морского флота свидетельствует о необходимости использования автоматизации и бизнес-моделирования с применением параметризуемой входной и выходной информации [3]. По мнению автора, входная информация, необходимая для оперативного управления работой морским флотом, должна содержать:

1) направления и размеры грузопотоков с разбивкой по периодам навигации, которые потенциально можно обслужить имеющимся флотом (с учетом грузопотоков, на которые можно оперативно переключить освободившиеся суда);

2) сведения по имеющимся в распоряжении судоходной компании морским судам, включая транспортно-технологические характеристики судов по каждому типоразмеру (габаритные параметры, марка и модель двигателя, расход топлива на ходу, на стоянке, при грузовых операциях, грузовые устройства и помещения, необходимый экипаж, сведения о корпусе и графике ремонта, обслуживания агрегатов);

3) сведения о географических условиях судоходства в каждом из обслуживаемых морских бассейнов, в том числе сведения

о глубинах, фарватерах, судовых ходах, ограничениях скорости движения судов и т.д.;

4) данные об эксплуатационных расходах по каждому типоразмеру судна, нормативные расходы на экипаж, запасы, питьевую и техническую воду, провизию;

5) сведения о стоимости различных видов топлива и горюче-смазочных материалов на всех участках движения судов с разбивкой по бассейнам и портам, а также сведения о размере и порядке выплат портовых/канальных сборов, агентских и брокерских вознаграждений;

6) данные о стоимости страхования судов и грузов, а также оценки стоимости планового текущего ремонта по каждому типоразмеру судна с разбивкой по бассейнам и портам.

Выходная информация, необходимая для оперативного управления работой морским флотом, должна быть пригодной для использования руководством и менеджерами компании для имплементации в работе [3] и, по мнению автора, должна содержать:

1) предлагаемую расстановку флота на участках работы с определением последовательности каждого рейса, позволяющую удовлетворить все уже поступившие и потенциальные заявки на исполнение услуги морской перевозки;

2) планируемые эксплуатационные показатели работы каждого судна и всего флота в целом на заданный период работы;

3) планируемые эксплуатационные затраты на содержание флота в целом с разбивкой по типоразмерам судов и бассейнам работы;

4) план по груженым рейсам с указанием последовательности грузоотправителей и грузополучателей, портов отправления и портов назначения, видов груза и размеров партии, а также с указанием доходов от каждой перевозки;

5) предварительное расписание и интервалы движения судов из пунктов отправления в пункты назначения;

6) планируемое расположение судов на конец навигационного периода;

7) комплексные экономические результаты работы флота: перечень суммарных доходов и расходов, сведения о производственных затратах/непроизводительных расходах, рентабельность деятельности судовой компании, маржа, прибыль с расчетом на тонно-километры.

После формирования состава входной и выходной информации, необходимой для принятия решений в рамках управления работой флота, автором предлагаются базовые принципы, которым должна соответ-

ствовать система оперативного управления работой флота в судоходных компаниях, работающих в рыночных условиях:

– Система управления работой флота должна соответствовать принципу оптимальности в части достижения максимальных целевых показателей путём эффективного использования имеющегося в распоряжении компании флота.

– Оперативное управление флотом должно строиться на принципе непрерывности и согласованности действий руководства и штата компании на всех уровнях управления.

– Менеджмент работы флота должен строиться на принципе достоверности и конкретности, когда итоговые эксплуатационные и экономические результаты выражаются в заранее определенных индикаторах и показателях.

– Система управления работой флота должна опираться на составление календарного графика работы флота (в т.ч. наклонных графиков работы судов) персонально по каждому судну.

– Оперативное управление флотом требует ежедневной сверки и корректировки графика работы флота в зависимости от изменения экономической и эксплуатационной обстановки.

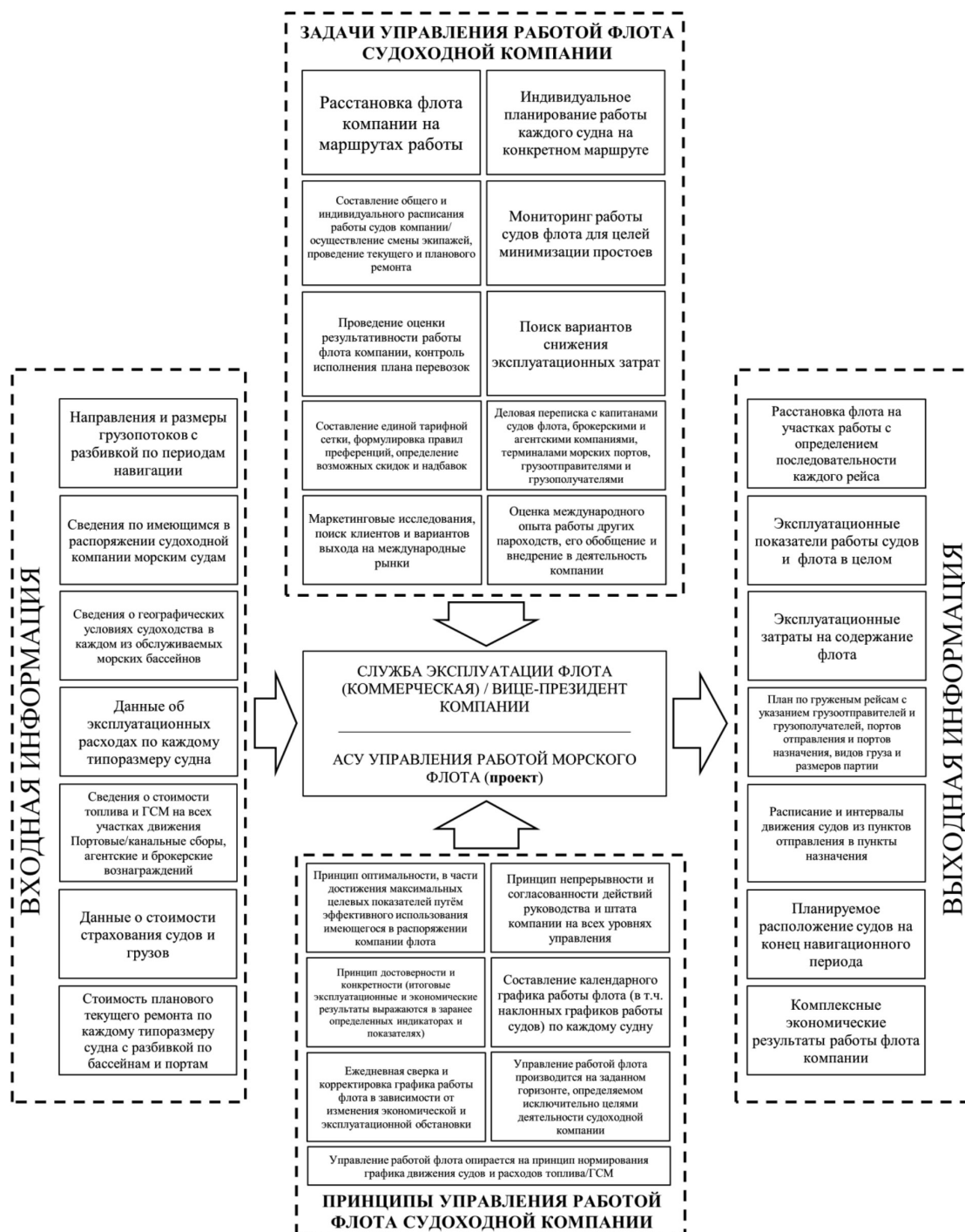
– Управление и планирование работы флота должно производиться на строго заданном горизонте, определяемом исключительно целями деятельности каждой конкретной судоходной компании.

– Управление работой флота должно опираться на принцип нормирования графика движения судов и расходов топлива/ГСМ для каждого рейса.

Имплементация указанных базовых принципов управления работой флота судоходной компании является трудозатратным и дорогостоящим процессом. Наиболее эффективно для судоходной компании их внедрение в рамках создания собственной электронной автоматизированной системы, выполняющей всю требуемую аналитическую работу и вычисления и позволяющей получать выходную информацию, пригодную для принятия рутинных операционных решений [4].

Пример внедрения предлагаемых принципов в структуру управления работой флота судоходной компании приведен на рисунке.

Автором установлено, что на сегодняшний день одним из наиболее эффективных инструментов для создания, апробации и тестирования подобной электронной автоматизированной системы является имитационное моделирование [5].



Внедрение предлагаемых принципов в структуру управления работой флота судоходной компании

В частности, агентное моделирование позволяет создавать модели рынка международного морского судоходства, устанавливать необходимое количество судоходных компаний (агентов), которые оперируют на рынке в каждый конкретный момент времени, анализировать динамику изменения

фрахтовых ставок в различных сегментах перевозок, а также устанавливать спрос и предложение на услуги международных морских перевозок.

В агентной модели также можно решать задачи по оптимизации расхода топлива и режима движения судов на каждом кон-

кратном участке перевозок, подбирать интервалы для планового и текущего ремонта с учётом характеристик флота компании [6].

Выводы

Отечественная область управления работой морского флота нуждается в научно обоснованных принципах повышения эффективности работы судов и штата компаний, которые можно реализовать на практике с учетом научно-технического прогресса современности. Повышению эффективности работы флота и персонала препятствует не столько отсутствие технических возможностей, сколько сложность для персонала в работе с огромным объемом информации, которую целесообразно выполнять с помощью автоматизированных информационных систем.

Создание подобных систем возможно с помощью таких доказавших свою эффективность инструментов, как имитационное агентное моделирование, которое возьмет на себя весь объем аналитической работы и математических вычислений

и позволит получать выходную информацию, понятную и пригодную для принятия обоснованных рутинных управленческих решений.

Список литературы

1. Зачесов В.П., Платов Ю.И. Текущее и оперативное планирование работы флота // Морские вести России. 2017. № 11. С. 52–55.
2. Арестова Ю.А. Организация оперативного управления работой флота судоходной компании // Сборник научных трудов «Sworld». 2013. № 3. С. 92–95.
3. Замятина Л.В. Проблемы судового менеджмента и оперативного управления работой флота // Научные проблемы транспорта Сибири и Дальнего Востока. 2014. № 1. С. 82–85.
4. Китов А.Г. Чуплыгин Г.Н. Проблемы оперативного управления и планирования работы флота в современных судоходных компаниях // Транспортное дело России. 2017. № 4. С. 86–87.
5. Громов С.В., Гусев Д.Е., Корьев В.Ю. Система поддержки принятия решений в судоходной компании в процессе оперативного управления работой флота // Речной транспорт (XXI ВЕК). 2016. № 4 (80). С. 58–62.
6. Китов А.Г. Лисин А.А. Чуплыгин Г.Н. Актуальные направления развития оперативного управления работой флота транспортное дело России // Морские вести России. 2017. № 3. С. 27–28.

УДК 551.46.09

ИССЛЕДОВАНИЕ И РАЗРАБОТКА ЭКСПЕРИМЕНТАЛЬНОГО ОБРАЗЦА АКУСТИЧЕСКОГО ИЗМЕРИТЕЛЯ СКОРОСТИ И НАПРАВЛЕНИЯ ТЕЧЕНИЙ

Лискин В.А., Тихонова Н.Ф.

Институт океанологии им. П.П. Ширшова РАН, Москва, e-mail: nrk@ocean.ru

Среди методов измерений скорости потока отметим гидродинамические преобразователи скорости течения, в качестве которых широкое распространение получили роторы Савониуса, винты Архимеда, импеллеры и т.д. Их распространенность обусловлена простой и технологичной конструкцией, линейной зависимостью между скоростью потока и угловой частотой вращения в рабочем диапазоне скоростей. Из акустических методов измерения скорости потока можно выделить два типа: основанные на эффекте Доплера и измерении времени распространения звука на известной базе. Метод измерения скорости потока по времени распространения звука вдоль известной базы основан на использовании зависимости времени распространения ультразвука между излучателем и приемником от скорости движения водных масс на его пути. Для однозначного определения направления течения в состав измерительного блока должны входить компас и акселерометр. Основные погрешности измерения скорости потока и скорости звука связаны с ошибками в измерении времени распространения и изменениями базы от температуры. Необходимый диапазон измерения времени распространения определяется в основном изменением скорости звука в воде, который составляет в среднем от 1400 до 1600 м/с или приблизительно 1500 м/с $\pm$ 7%. Из всех приведенных акустических методов измерения потока в настоящее время широко используются доплеровский и фазовый. Основным элементом ультразвуковой системы измерительного модуля скорости и направления течения являются пьезоэлектрические (пьезокерамические) датчики. Датчики изготавливаются на основе пьезоэлементов в виде тонких пластин пьезокерамики с продольной поляризацией. В ходе первоочередных работ по созданию ультразвукового измерителя скорости потока были проведены испытания макетов датчиков на предмет исследования их чувствительности.

Ключевые слова: акустические, скорость, поток, направление, течения, погрешности, измерения

RESEARCH AND DEVELOPMENT OF EXPERIMENTAL SAMPLE OF ACOUSTIC SPEED METER AND DIRECTION OF FLOW

Liskin V.A., Tikhonova N.F.

Shirshov Institute of Oceanology, Russian Academy of Science, Moscow, e-mail: nrk@ocean.ru

Among the methods for measuring the flow velocity, we note hydrodynamic flow velocity transducers, which are widely used: Savonius rotors, Archimedes screws, impellers, etc. Their prevalence is due to a simple and technologically advanced construction, a linear relationship between the flow rate and the angular frequency of rotation in the operating speed range. Two types of acoustic methods for measuring the flow velocity can be distinguished: those based on the Doppler effect and measuring the propagation time of sound on a known basis. The method of measuring the flow velocity by the propagation time of sound along a known base is based on the use of the dependence of the propagation time of ultrasound between the emitter and receiver on the speed of movement of water masses along its path. To unambiguously determine the direction of flow, the compass and accelerometer should be included in the composition of the measuring unit. The main errors in measuring the flow velocity and sound velocity are associated with errors in measuring the propagation time and changes in the base from temperature. The required measurement range of the propagation time is determined mainly by the change in the speed of sound in water, which averages from 1400 to 1600 m/s or approximately 1500 m/s $\pm$ 7%. Of all the acoustic methods for measuring flow, Doppler and phase are currently widely used. The main element of the ultrasonic system of the measuring module of speed and direction of flow are piezoelectric (piezoceramic) sensors. Sensors are made on the basis of piezoelectric elements in the form of thin plates of piezoelectric ceramics with longitudinal polarization. In the course of the priority work on the creation of an ultrasonic flow velocity meter, we tested the prototype sensors for the study of their sensitivity.

Keywords: acoustic, speed, flow, direction, currents, errors, measurements

Среди методов измерений скорости потока отметим гидродинамические преобразователи скорости течения, в качестве которых широкое распространение получили роторы Савониуса, винты Архимеда, импеллеры и т.д. Их распространенность обусловлена простой и технологичной конструкцией, линейной зависимостью между скоростью потока и угловой частотой вращения в рабочем диапазоне скоростей. К недостаткам этих преобразователей скорости течения можно отнести высокую нелинейность при малых скоростях потока,

наличие вращающихся частей и достаточно большую инерционность. С другой стороны, отметим, что измерение скорости течения проводящей жидкости возможно на основе использования метода электромагнитной индукции, при котором возможно появление ЭДС во время движения электропроводящей жидкости в магнитном поле. При этом возникающая ЭДС соответствует величине векторного произведения напряженности магнитного поля и скорости движения жидкости. При проектировании магнитогидродинамических преобразовате-

лей необходимо уделять большое внимание конструкции самого преобразователя и особенно материалам электродов. К основным недостаткам можно отнести открытые электроды в электролите (морской воде), что может привести к изменению показаний со временем.

Акустические методы

Из акустических методов измерения скорости потока можно выделить два типа: основанные на эффекте Доплера и измерении времени распространения звука на известной базе. Метод, основанный на эффекте Доплера, заключается в том, что если излучить в наклонном направлении к движущемуся слою воды, содержащему взвешенные частицы, акустические колебания и затем принять рассеянные водным слоем сигналы, то частота принятых колебаний будет отличаться от частоты излученных колебаний на величину, зависящую от скорости потока. Преимущество этого способа состоит в том, что с помощью одного прибора можно измерять скорость потока по толщине некоторого столба воды. К недостаткам можно отнести сложность приемо-передающей части, обязательное наличие взвешенных частиц, а также этот способ измерения потока может иметь большую погрешность, связанную с изменением скорости звука. Метод измерения скорости потока по времени распространения звука вдоль известной базы основан на использовании зависимости времени распространения ультразвука между излучателем и приемником, от скорости движения водных масс на его пути. Методы и технические средства гидрофизических измерений их использующие рассмотрены в литературных источниках [1–3]. Ниже рассмотрим некоторые методы измерения скорости звука в водной толще детально.

Метод по времени распространения. Рассмотрим следующую схему (рис. 1).

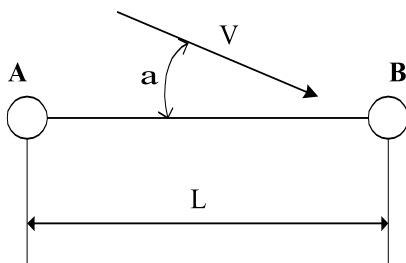


Рис. 1. Схема, поясняющая метод измерения скорости потока по времени распространения.

A и B – обратимые приемо-излучатели;
L – расстояние между ними (в дальнейшем база);
V – скорость потока; α – угол между направлением скорости течения и базой;
C – скорость звука

Для упрощения последующих математических выражений под V_L будет подразумеваться проекция скорости течения на базу, т.е. $V \cdot \cos(\alpha)$. Время распространения звука от излучателя к приемнику в этом случае описывается следующим выражением:

$$\tau_{ab} = \frac{L}{C + V_L} \quad \text{– при распространении}$$

от A к B,

$$\tau_{ba} = \frac{L}{C - V_L} \quad \text{– при распространении}$$

от B к A.

Если считать, что скорость потока за время этих измерений не менялась, то простыми преобразованиями получаем

$$C = \frac{1}{2} L \left(\frac{1}{\tau_{ab}} + \frac{1}{\tau_{ba}} \right), \quad (1)$$

$$V_L = \frac{1}{2} L \left(\frac{1}{\tau_{ab}} - \frac{1}{\tau_{ba}} \right). \quad (2)$$

Для измерения времени распространения звука вдоль базы можно рассмотреть несколько приемов: частотно-импульсный, время-импульсный и фазовый.

Частотно-импульсный метод. Рассмотрим схему на рис. 2.

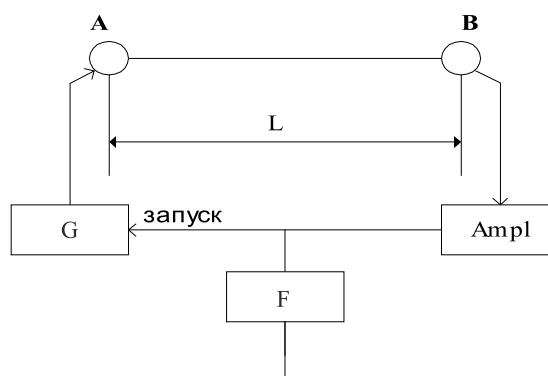


Рис. 2. Схема, поясняющая частотно-импульсный метод измерения скорости потока. A – излучатель, B – приемник; L – расстояние между излучателем и приемником (в дальнейшем база); G – генератор коротких одиночных импульсов с внешним запуском; Ampl – усилитель; F – частотомер

В этом замкнутом электронно-акустическом тракте частота повторения импульсов определяется временем распространения звука от A к B. Основным преимуществом этого метода является большой динамический диапазон измерения скорости потока. При небольшом усложнении схемы можно измерять скорость распространения в обоих направлениях и таким образом производить вычисления по формулам (1) и (2).

При этом способе надо учитывать фазовые и частотные характеристики приемо-передатчиков и электронного тракта во всем используемом диапазоне частот, что сказывается на линейности преобразователя. Дополнительные сложности может вызвать паразитная амплитудная модуляция.

Время-импульсный метод. Рассмотрим схему на рис. 3.

В этом случае непосредственно измеряется задержка распространения от приемника к излучателю. По своим особенностям практически не отличается от частотно-импульсного метода, только измерения происходят при одинаковой частоте импульсов. Основным недостатком является погрешность измерения от крутизны фронтов импульсов, снимаемых с приёмника.

Фазовый метод. Рассмотрим схему на рис. 4.

Ниже представлена блок-схема фазового измерителя потока (рис. 5).

Под управлением микропроцессора переключателем SW осуществляется выбор направления излучения-приёма. Для исключения неоднозначности фазы, связанной с большим диапазоном возможного изменения скорости звука, измерения могут производиться на нескольких частотах в диапазоне от 100 кГц до 2 МГц, в состав измерительного блока должен быть введён измеритель скорости звука. Скорость звука может быть также получена косвенными методами по результатам измерения солёности (электропроводности), давления и температуры. В приемнике происходит предварительное усиление сигнала с пьезокерамического датчика и формирование импульсов необходимых для работы фазового детектора.

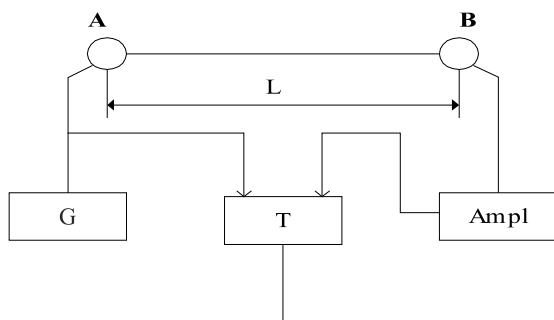


Рис. 3. Схема, поясняющая время-импульсный метод измерения скорости потока.
A – излучатель; B – приемник; L – расстояние между излучателем и приемником (в дальнейшем база); G – генератор коротких импульсов; Ampl – усилитель; T – измеритель периода

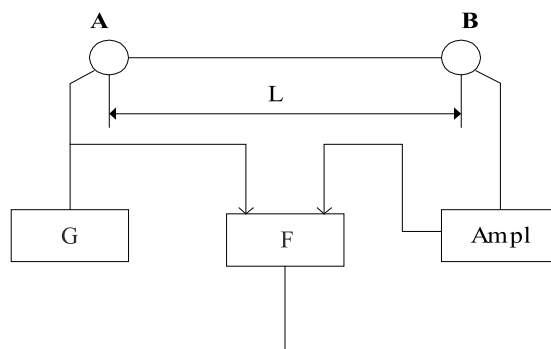


Рис. 4. Блок-схема, поясняющая фазовый метод измерения скорости потока.
A – излучатель; B – приемник; L – расстояние между излучателем и приемником (в дальнейшем база); G – генератор сигнала; Ampl – усилитель; F – фазовый детектор

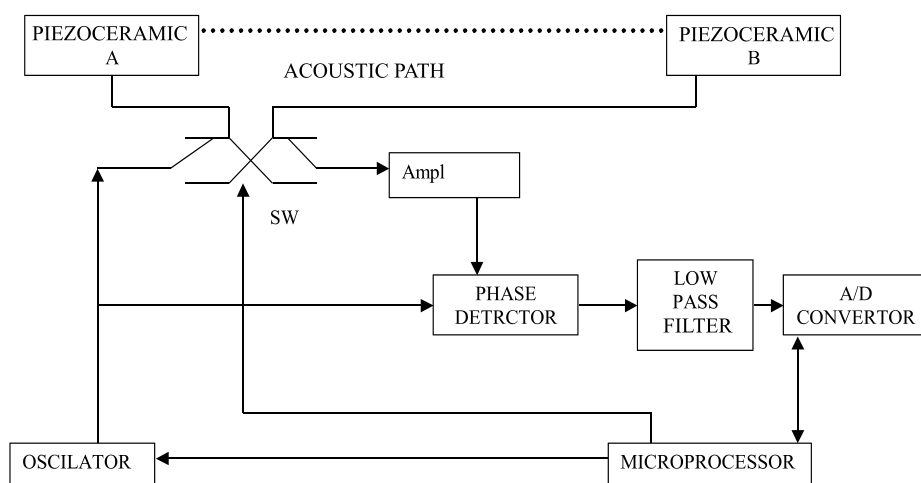


Рис. 5. Блок-схема фазового измерителя скорости потока

В этом случае фазу принятого сигнала можно выразить так (по одному направлению):

$$\varphi_{ab} = \varphi_{ia} + \varphi_{uab} + \varphi_{rb} + \varphi_{rec},$$

где φ_{ia} – фаза между напряжением, приложенным к излучателю, и акустическим давлением;

φ_{rb} – фаза между акустическим давлением и напряжением на выходе пьезоприёмника;

φ_{rec} – фазовый сдвиг в предварительном усилителе;

φ_{uab} – набег фазы за время распространения по воде;

φ_{ab} – разность фаз на входе фазового детектора.

Фазовые сдвиги в приёмнике, излучателе и предварительном усилителе не зависят от параметров среды, и их можно учесть на этапе калибровки. Разность фаз на входе фазового детектора определяется временем распространения сигнала вдоль базы, в случае если время распространения известно с точностью периода частоты, на которой ведутся измерения, то φ_{ab} однозначно определяет время распространения сигнала.

Для исключения дополнительных погрешностей, связанных с нелинейностью фазового детектора на краях диапазона измерения и постоянным смещением, в канале прямого сигнала может применяться устройство сдвига фазы на угол 0–90–180–270 градусов, контролируемое микропроцессором [1–3].

Дополнительные датчики

Для однозначного определения направления течения в состав измерительного блока должен входить компас и акселерометр. В настоящее время наиболее подходящим для данной задачи является модуль (например, HMR3300 фирмы HONEYWELL), который позволяет измерять направление с точностью не хуже 1 градуса. В состав данного модуля также входит акселерометр с диапазоном ± 60 градусов и точностью 1 градус. Также возможно применение набора (микросхем HMC1055 той же фирмы), что позволит сократить размеры модуля и энергопотребление. Для повышения точности измерения потока в состав прибора целесообразно ввести датчик температуры для компенсации температурных погрешностей, связанных с изменениями базы от температуры.

Основные погрешности измерений

Основные погрешности измерения скорости потока и скорости звука связаны с ошибками в измерении времени распространения и изменениями базы от температуры. Необходимый диапазон измерения времени

распространения определяется в основном изменением скорости звука в воде, который составляет в среднем от 1400 до 1600 м/с или приблизительно 1500 м/с $\pm 7\%$. Скорость самого потока составляет менее 1% от скорости звука. Если положить базу равной 100 мм, необходимый диапазон измерения времени распространения составит приблизительно от 60 нс до 70 нс или 65 ± 5 нс. Для достижения точности измерения потока около 10 мм/с необходимо измерять время с погрешностью не более 0,5 нс. В случае время-импульсного или частотного методов достижение такой точности затруднительно, а при фазовом методе разрешение по времени в течение одного периода несущей частоты определяется погрешностью фазового детектора на частоте, при которой происходит измерение. Если считать разрешение фазового детектора менее 0,1 градуса, то на частоте 2 МГц это будет соответствовать разрешению лучше 0,5 нс. При частотном диапазоне от 100 кГц до 2 МГц вопрос о неоднозначности фазы практически снимается, если производить измерения на нескольких частотах. Так наибольшая возможная разность времени распространения сигнала при базе 0,1 м составит около 10 мкс, что соответствует частоте 100 кГц. Изменение базы от температуры зависит от типа применяемых материалов и конструкции для крепления пьезоэлементов. Для оценки можно рассмотреть случай, когда база составляет приблизительно 100 мм и материал имеет температурный коэффициент расширения $\lambda = 11 \cdot 10^{-6} (1/^\circ\text{C})$ (сталь нержавеющая). При изменении температуры на 50 $^\circ\text{C}$ изменение длины составит 0,055 мм или около 0,06%, что приведёт к погрешности измерения потока около 7%. В случае применения стеклопластика с температурным коэффициентом расширения около $\lambda = 20 \cdot 10^{-6} (1/^\circ\text{C})$ погрешность измерения потока будет близка к 15%. Для повышения точности можно использовать введение поправок на температуру [4, 5].

Обоснование выбора методов

Из всех приведенных выше акустических методов измерения потока в настоящее время широко используются доплеровский и фазовый. Измерители потока, основанные на эффекте Доплера, выпускаются многими фирмами, но в состав этих измерителей входит сложная приёмо-передающая акустическая система. Для повышения чувствительности к мелким взвешенным частицам используют частоты порядка 1 МГц, что приводит к значительным мощностям на излучение для работы в достаточно большом диапазоне расстояний. Фазовые измерители скорости потока имеют значительно более простые

приёмо-передающие узлы и так же широко выпускаются, но мощность излучения значительно ниже по сравнению с доплеровскими измерителями. При выборе метода измерения потока в основном внимание уделялось простоте конструкции датчика и низкому энергопотреблению, что важно в случае использования датчика скорости потока в составе автономной станции. Эти датчики могут применяться при любом способе, основанном на измерении времени распространения. Достаточно простая конструкция электронной части получается при фазовом методе, так как в этом случае нет необходимости компенсировать паразитную амплитудную модуляцию, а фазо-частотные характеристики приёмно-передающего тракта могут быть калиброваны на фиксированных частотах.

Исследования характеристик пьезодатчиков

Основным элементом ультразвуковой системы измерительного модуля скорости и направления течения являются пьезоэлектрические (пьезокерамические) датчики. Датчики изготавливаются на основе пьезоэлементов в виде тонких пластин пьезокерамики с продольной поляризацией. В ходе первоочередных работ по созданию ультразвукового измерителя скорости потока были проведены испытания макетов датчиков на предмет исследования их чувствительности. При проведении лабораторных испытаний главной задачей была отработка технологий герметизации пьезодатчиков, как и выбор образцов пьезодатчиков. Был выполнен подбор высокочастотных пьезодатчиков с резонансными частотами от 350 кГц до 800 кГц. Все протестированные датчики позволяли работать в требуемом частотном диапазоне более одной декады. Для герметизации пьезоэлементов применялись силиконовые герметики. Лучшие результаты показали датчики, где пьезокерамика жёстко прикреплялась на подложку по всей плоскости. Амплитуда напряжения на излучателе была постоянной и составляла 7 В. Для проведения измерений был разработан предусилитель на дискретных элементах, так как на таких частотах интегральные усилители имели большой потребляемый ток: около 10 мА, тогда как разработанный усилитель потреблял 2,7 мА [6, 7].

Заключение

Из акустических методов измерения скорости потока можно выделить два типа: основанные на эффекте Доплера и измерении времени распространения звука на известной базе. Метод измерения скорости потока по времени распространения звука вдоль

известной базы основан на использовании зависимости времени распространения ультразвука между излучателем и приемником, от скорости движения водных масс на его пути. Для однозначного определения направления течения в состав измерительного блока должны входить компас и акселерометр. Необходимый диапазон измерения времени распространения определяется в основном изменением скорости звука в воде, который составляет в среднем от 1400 до 1600 м/с или приблизительно 1500 м/с $\pm$ 7%. Отметим, из приведенных выше акустических методов измерения потока в настоящее время широко используются доплеровский и фазовый. Важнейшим элементом ультразвуковой системы измерительного модуля скорости и направления течения являются пьезоэлектрические (пьезокерамические) датчики. Датчики изготавливаются на основе пьезоэлементов в виде тонких пластин пьезокерамики с продольной поляризацией. В ходе первоочередных работ по созданию ультразвукового измерителя скорости потока были проведены испытания макетов датчиков на предмет исследования их чувствительности. Основные погрешности измерения скорости потока и скорости звука связаны с ошибками в измерении времени распространения и изменениями базы от температуры.

Работа выполнена в рамках государственного задания ИО РАН на 2020 год (тема № 0149-2020-0011) при поддержке РФФИ (проект № 20-05-00384_А и проект № 18-05-60070 «Арктика»).

Список литературы

1. Паршуков В.Н. Средства гидроакустического и гидрологического обеспечения. // Подводные технологии и средства освоения Мирового океана. М.: Издательский дом «Оружие и технологии», 2011. С. 337–347.
2. Римский-Корсаков Н.А., Флинт М.В., Поярков С.Г., Анисимов И.М., Белевитнев Я.И., Пронин А.А., Тронза С.Н. Развитие технологии комплексных инструментальных подводных наблюдений применительно к экосистемам Российской Арктики // Океанология. 2019. Т. 59. № 4. С. 679–683. DOI: 10.1134/S0001437019040143.
3. Толстошеев А.П., Лунев Е.Г., Мотыжев С.В. Высокоточный измеритель скорости звука автономного дрейфующего буя // Современные методы и средства океанологических исследований: матер. XVI Всерос. научно-технической конференции. М.: АПР, 2019. Т. 2. С. 60–64.
4. Носов А.В. Особенности метода термокомпенсации временной погрешности для автономных регистраторов сигналов // Океанологические исследования. 2017. Т. 45. № 1. С. 163–170. DOI: 10.29006/1564-2291.JOR-2017.45(1).12.
5. Островский А.Г., Волков С.В., Зацепин А.Г., Кочетов О.Ю., Швоев Д.А., Ольшанский В.М. Автономный привязной профилограф «винчи» для морского шельфа, озер и водохранилищ // Современные методы и средства океанологических исследований (МСОИ-2019): материалы XVI Всероссийской научно-технической конференции. 2019. Т. 2. С. 174–177.
6. Римский-Корсаков Н.А., Сычев В.А. Акустический профилограф с дифференциальным выходным приемным устройством // Океанология. 1998. Т. 38. № 2. С. 315–317.
7. Губин Н.А., Ченский Д.А., Ченский А.Г. Лабораторный стенд для калибровки измерителей скорости звука // Современные методы и средства океанологических исследований (МСОИ-2015): материалы XIV Международной научно-технической конференции. 2015. Т. 1. С. 120–123.