

*Журнал «Научное обозрение. Технические науки»
зарегистрирован Федеральной службой
по надзору в сфере связи, информационных
технологий и массовых коммуникаций.
Свидетельство ПИ № ФС77-57440
выдано 27.03.2014.
ISSN 2500-0799*

**Двухлетний импакт-фактор РИНЦ – 0,695
Пятилетний импакт-фактор РИНЦ – 0,319**

*Учредитель, издатель и редакция:
ООО НИЦ «Академия Естествознания»,*

*Почтовый адрес: 101000, г. Москва, а/я 47
Адрес учредителя, издателя: 410056, г. Саратов,
ул. им. Чапаева В.И., д. 56
Адрес редакции: 410035, г. Саратов,
ул. Мамонтовой, д. 5*

**Founder, publisher and edition:
LLC SPC Academy of Natural History,**

**Post address: 101000, Moscow, p.o. box 47
Founder's, publisher's address: 410056, Saratov,
56 Chapaev V.I. str.
Editorial address: 410035, Saratov,
5 Mamontovoi str.**

*Подписано в печать 29.08.2025
Дата выхода номера 30.09.2025
Формат 60×90 1/8*

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

**Signed in print 29.08.2025
Release date 30.09.2025
Format 60×90 8.1**

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

*Технический редактор Доронкина Е.Н.
Корректор Галенкина Е.С., Дудкина Н.А.*

*Распространяется по свободной цене
Тираж 1000 экз. Заказ НО 2025/4
Подписной индекс в электронном каталоге
«Почта России»: ПА518
© 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, Mendeleev, 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. Shnurovovozova**)

НАУЧНОЕ ОБОЗРЕНИЕ • ТЕХНИЧЕСКИЕ НАУКИ

SCIENTIFIC REVIEW • TECHNICAL SCIENCES

www.science-education.ru

2025 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

СТАТЬИ

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ОБУЧЕНИЯ НА БАЗЕ AUTOCAD С ИСПОЛЬЗОВАНИЕМ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ

Винтерголлер И.Г., Мурых Е.Л., Синкевич Н.Н. 5

МЕТОДИКА ПРИМЕНЕНИЯ ПСЕВДОПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ

Лелейкин С.С. 11

СРАВНЕНИЕ ПОДХОДОВ К ВЕБ-РАЗРАБОТКЕ: ПРЕИМУЩЕСТВА И ОГРАНИЧЕНИЯ NO-CODE И ТРАДИЦИОННОГО ПРОГРАММИРОВАНИЯ

Пепельшев Д.И., Кулагина Е.С. 17

РАЗРАБОТКА МНОГОФУНКЦИОНАЛЬНОЙ ПЛАТФОРМЫ ДЛЯ АГРЕГИРОВАНИЯ НОВОСТНЫХ СТАТЕЙ

Кольева Н.С., Хасимьянова И.В., Кольев И.Н., Гонцова А.О. 22

НАУЧНЫЙ ОБЗОР

ПРИМЕНЕНИЕ КОЖУРЫ ГРАНАТА В ПИЩЕВОЙ ПРОМЫШЛЕННОСТИ

Хамзина Е.И. 28

CONTENTS

Technical sciences

ARTICLES

DEVELOPMENT OF AN INTELLIGENT LEARNING SYSTEM BASED ON AUTOCAD USING MODERN TECHNOLOGIES

Vintergoller I.G., Murykh E.L., Sinkevich N.N. 5

METHODOLOGY OF APPLICATION OF PSEUDO-PARALLEL ALGORITHMS

Leleykin S.S. 11

COMPARISON OF APPROACHES TO WEB DEVELOPMENT: ADVANTAGES AND LIMITATIONS OF NO-CODE AND TRADITIONAL PROGRAMMING

Pepelyshev D.I., Kulagina E.S. 17

DEVELOPMENT OF A MULTIFUNCTIONAL PLATFORM FOR AGGREGATION OF NEWS ARTICLES

Koleva N.S., Khasimyanova I.V., Kolev I.N., Gontsova A.O. 22

REVIEW

APPLICATION OF POMEGRANATE PEEL IN THE FOOD INDUSTRY

Khamzina E.I. 28

СТАТЬИ

УДК 004.896

**РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ
ОБУЧЕНИЯ НА БАЗЕ AUTOCAD С ИСПОЛЬЗОВАНИЕМ
СОВРЕМЕННЫХ ТЕХНОЛОГИЙ****Винтерголлер И.Г., Мuryх Е.Л., Синкевич Н.Н.***НАО «Карагандинский технический университет имени Абылкаса Сагинова»,
Караганда, e-mail: i.wintergoller@ktu.edu.kz*

В статье представлена разработка интеллектуальной системы обучения для систем автоматизированного проектирования на базе программного комплекса AutoCAD, направленная на повышение эффективности подготовки специалистов в области проектирования. Интеллектуальная система обучения использует современные технологии, такие как Python, pyautocad и PostgreSQL, для интеграции с AutoCAD и создания базы знаний, основанной на положениях национальных стандартов Российской Федерации. Описаны методы интеграции системы с AutoCAD, управления базой знаний и обеспечения интерактивного обучения. Система автоматизирует проверку чертежей, предоставляет адаптивную обратную связь. Программная реализация включает модули для создания и управления блоками, обработки отношений и правил, базами знаний, заданиями и обратной связью, реализованные через SQLAlchemy для работы с PostgreSQL. Тестирование проводилось на 60 студентах в рамках изучения дисциплины «Компьютерная графика» и показало сокращение времени выполнения заданий в среднем на 25 % и снижение ошибок на 45 % по сравнению с традиционными методами. Система обладает гибкостью и масштабируемостью, но ограничена зависимостью от локальной установки AutoCAD. Перспективы развития включают интеграцию машинного обучения, облачных технологий и естественно-языкового интерфейса.

Ключевые слова: интеллектуальная система обучения, системы автоматизированного проектирования, AutoCAD, база знаний, ГОСТы, интерактивное обучение, адаптивность

**DEVELOPMENT OF AN INTELLIGENT LEARNING SYSTEM
BASED ON AUTOCAD USING MODERN TECHNOLOGIES****Vintergoller I.G., Murykh E.L., Sinkevich N.N.***Abylkas Saginov Karaganda Technical University, Karaganda,
e-mail: i.wintergoller@ktu.edu.kz*

The article presents the development of an intelligent learning system for computer-aided design systems based on the AutoCAD software package, aimed at improving the effectiveness of training specialists in the field of design. The intelligent learning system uses modern technologies such as Python, pyautocad and PostgreSQL to integrate with AutoCAD and create a knowledge base based on the provisions of the national standards of the Russian Federation. The methods of integrating the system with AutoCAD, knowledge base management, and interactive learning are described. The system automates the inspection of drawings and provides adaptive feedback. The software implementation includes modules for creating and managing blocks, processing relationships and rules, knowledge bases, tasks, and feedback implemented through SQLAlchemy to work with PostgreSQL. Testing was conducted on 60 students as part of the study of the discipline "Computer Graphics" and showed a reduction in task completion time by an average of 25 % and a reduction in errors by 45 % compared to traditional methods. The system has flexibility and scalability, but is limited by its dependence on a local AutoCAD installation. Development prospects include the integration of machine learning, cloud technologies, and a natural language interface.

Keywords: intelligent learning system, computer-aided design systems, AutoCAD, knowledge base, GOST, interactive learning, adaptability

Введение

Системы автоматизированного проектирования (САПР), такие как AutoCAD, играют ключевую роль в инженерной практике, обеспечивая проектирование сложных объектов в различных отраслях, от машиностроения до архитектуры [1–3]. Однако освоение AutoCAD представляет значительные трудности из-за его функциональной сложности и необходимости соблюдения национальных стандартов, таких как ГОСТы. Традиционные методы обучения, основан-

ные на лекциях, учебниках и практических занятиях, не успевают за темпами обновления технологий и не обеспечивают индивидуального подхода к обучаемым [1, 4].

Интеллектуальные системы обучения (ИОС), использующие принципы искусственного интеллекта (ИИ), предлагают решение этих проблем [5–7]. ИОС способны имитировать роль преподавателя, адаптировать учебный процесс к уровню знаний пользователя и предоставлять обратную связь в реальном времени [4, 6]. Интегра-

ция базы знаний, основанной на ГОСТах, позволяет автоматизировать проверку соответствия чертежей стандартам, что особенно важно для подготовки специалистов в странах с жесткими нормативными требованиями, таких как Россия, Казахстан [1, 5].

Разработка ИОС для AutoCAD с использованием современных технологий, таких как Python и PostgreSQL, представляет собой актуальную задачу [6, 8, 9]. Python обеспечивает гибкость и широкую поддержку библиотек, включая `pyautocad` для взаимодействия с AutoCAD, а PostgreSQL гарантирует надежное хранение и масштабируемость базы знаний. Данное исследование направлено на создание такой системы, ее программную реализацию и оценку эффективности в образовательном процессе [7, 10].

Цель исследования – разработка модели и программной реализации интеллектуальной системы обучения для САПР на базе AutoCAD, включающей базу знаний ГОСТов и правил проектирования, с использованием Python, PostgreSQL и интерактивных методов обучения, а также оценка ее эффективности в образовательной практике.

Материалы и методы исследования

Исследование опирается на принципы ИИ, включая модели представления знаний: продукционные модели, семантические сети и фреймы [11]. Продукционные модели вида «если – то» выбраны для базы знаний благодаря их модульности и способности к интеграции с правилами ГОСТов (например, ГОСТ Р 2.104-2023). ИОС классифицируется как экспертно-тренирующая

система, формирующая навыки проектирования и предоставляющая адаптивную обратную связь [8, 12, 13].

ИОС включает три ключевые коллекции объектов:

1. Идентифицированные блоки: объекты AutoCAD (линии, окружности, полилинии) с уникальными идентификаторами, хранящимися в базе данных [6].

2. Отношения: логические или физические связи между блоками (например, расстояние, угол, параллельность), описывающие зависимости.

3. Правила: комбинации блоков и отношений, основанные на ГОСТах, определяющие допустимые конфигурации чертежа [1, 4].

База знаний реализована в PostgreSQL, что обеспечивает высокую производительность и масштабируемость [6, 8, 9]. Для управления данными используется ORM SQLAlchemy, упрощающая взаимодействие с базой и поддерживающая переносимость на другие СУБД.

Архитектура ИОС. Диаграмма включает четыре основных компонента (рис. 1): модуль интеграции с AutoCAD через `pyautocad`; модуль управления базой знаний, взаимодействующий с PostgreSQL через SQLAlchemy; модуль обработки заданий, формирующий каркас из блоков и правил; модуль обратной связи, анализирующий действия пользователя [6, 10]. Компоненты соединены стрелками, показывающими потоки данных.

База данных системы. Составлена ER-диаграмма с тремя таблицами (рис. 2): Blocks, Relations, Rules. Таблицы связаны через внешние ключи [8].

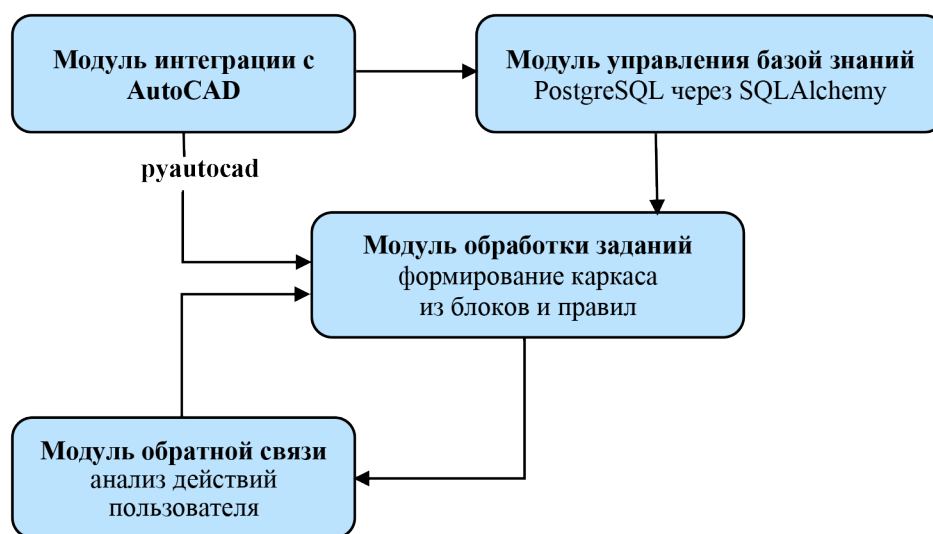


Рис. 1. Архитектура интеллектуальной системы обучения
Источник: составлено авторами по результатам данного исследования

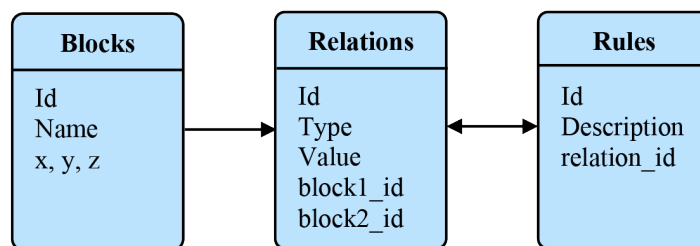


Рис. 2. ER-диаграмма

Источник: составлено авторами по результатам данного исследования

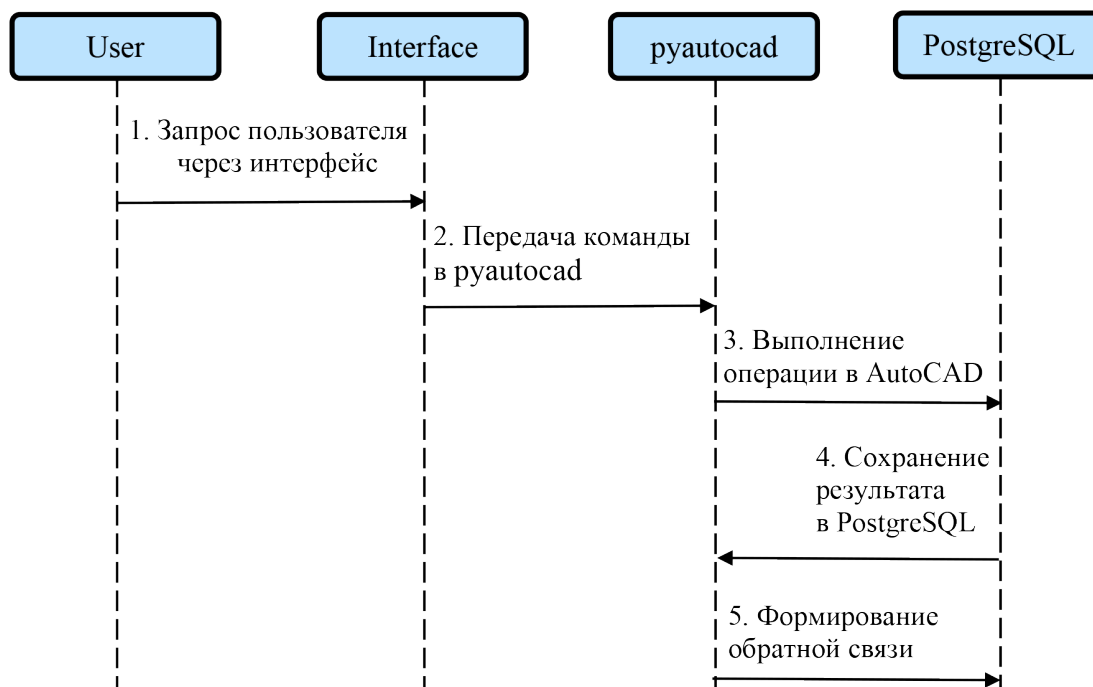


Рис. 3. UML-диаграмма процесса взаимодействия с AutoCAD

Источник: составлено авторами по результатам данного исследования

Процесс взаимодействия с AutoCAD.

На рис. 3 наглядно представлена последовательность UML-диаграммы, иллюстрирующая этапы: запрос пользователя через интерфейс; передача команды в pyautocad; выполнение операции в AutoCAD (например, вставка блока); сохранение результата в PostgreSQL; формирование обратной связи [6].

ИОС реализована на Python 3.10 с использованием библиотеки pyautocad для интеграции с AutoCAD [3]. Основные модули системы:

- модуль подключения к AutoCAD: взаимодействие с активным экземпляром AutoCAD через COM-интерфейс;
- модуль управления блоками: создание, вставка и модификация блоков в чертеже;
- модуль базы знаний: управление правилами и отношениями в PostgreSQL;

- модуль обратной связи: анализ действий обучаемого и предоставление рекомендаций;
- модуль тестирования: автоматизированная проверка чертежей на соответствие ГОСТам [14].

Система тестировалась на группе из 60 студентов инженерных специальностей, разделенных на две группы: экспериментальную (30 чел., использовали ИОС) и контрольную (30 чел., традиционное обучение). Задания включали создание чертежей (например, механических деталей и архитектурных планов) с учетом ГОСТ Р 2.104-2023 и ГОСТ Р 2.105-2019 [1]. Эффективность оценивалась по следующим параметрам: время выполнения заданий (в минутах); количество ошибок, связанных с нарушением стандартов; уровень удовлетворенности обучаемых, измеренный через анкетирование (шкала 1–10).

Подключение к AutoCAD

```
from pyautocad import Autocad, APoint
from comtypes import COMError
def connect_to_autocad(create_if_not_exists=True):
    try:
        acad = Autocad(create_if_not_exists=create_if_not_exists)
        acad.app.Visible = True
        print("Подключено к AutoCAD")
        return acad
    except COMError as e:
        raise Exception(f"Ошибка подключения к AutoCAD: {e}")
if __name__ == "__main__":
    acad = connect_to_autocad()
```

Подключение к базе данных

```
from sqlalchemy import create_engine, Column, Integer, String, Float, ForeignKey
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker, relationship
Base = declarative_base()
class Block(Base):
    __tablename__ = 'blocks'
    id = Column(Integer, primary_key=True)
    name = Column(String, nullable=False)
    x = Column(Float)
    y = Column(Float)
    z = Column(Float)
class Relation(Base):
    __tablename__ = 'relations'
    id = Column(Integer, primary_key=True)
    type = Column(String, nullable=False) # e.g., 'distance', 'angle'
    value = Column(Float)
    block1_id = Column(Integer, ForeignKey('blocks.id'))
    block2_id = Column(Integer, ForeignKey('blocks.id'))
    block1 = relationship("Block", foreign_keys=[block1_id])
    block2 = relationship("Block", foreign_keys=[block2_id])
class Rule(Base):
    __tablename__ = 'rules'
    id = Column(Integer, primary_key=True)
    description = Column(String, nullable=False) # e.g., 'ГОСТ 2.104-2023'
    relation_id = Column(Integer, ForeignKey('relations.id'))
    relation = relationship("Relation")
def connect_to_db():
    engine = create_engine('postgresql://user:password@localhost:5432/acad_db')
    Base.metadata.create_all(engine)
    Session = sessionmaker(bind=engine)
    return Session()
if __name__ == "__main__":
    session = connect_to_db()
```

Создание блока

```
def create_block(acad, block_name, points):
    model_space = acad.model
    for i in range(len(points) - 1):
        model_space.AddLine(APoint(*points[i]), APoint(*points[i + 1]))
    model_space.AddLine(APoint(*points[-1]), APoint(*points[0]))
    print(f"Фигура '{block_name}' создана в ModelSpace")
```


Вставка блока

```
def insert_block(acad, insert_point, block_name):
    doc = acad.doc
    model_space = doc.ModelSpace
    block_ref = model_space.InsertBlock(APoint(*insert_point), block_name, 1, 1, 1, 0)
    block_ref.Update()
    print(f'Блок '{block_name}' вставлен в точке {insert_point}')
    return block_ref
if __name__ == "__main__":
    acad = connect_to_autocad()
    insert_block(acad=acad, block_name="NewBlock", insert_point=(20, 30, 0))
```

Проверка соответствия ГОСТам

```
def verify_gost_compliance(session, block1_id, block2_id, relation_type, expected_value):
    relation = session.query(Relation).filter_by(
        block1_id=block1_id,
        block2_id=block2_id,
        type=relation_type
    ).first()
    if relation and abs(relation.value - expected_value) < 0.01:
        print(f'Соответствие ГОСТу для {relation_type}: проверено')
        return True
    else:
        print(f'Нарушение ГОСТа для {relation_type}: ожидалось {expected_value}, получено {relation.value}')
        return False
```

Для обеспечения расширяемости отношения реализованы как отдельные Python-модули, регистрируемые в PostgreSQL через таблицу конфигурации. Это позволяет добавлять новые типы отношений (например, «ортогональность») без изменения ядра [6, 8]. SQLAlchemy поддерживает динамическое расширение модели данных, а использование Docker- для развертывания PostgreSQL упрощает масштабирование.

Результаты исследования и их обсуждение

Разработанная ИОС успешно интегрируется с AutoCAD через ruautocad, обеспечивая управление блоками и проверку чертежей в реальном времени. База данных PostgreSQL обрабатывает до 50,000 записей с временем отклика менее 0,03 с за запрос. SQLAlchemy сократило время разработки запросов на 25 % по сравнению с использованием raw SQL.

Экспериментальная группа, использующая ИОС, выполнила задания в среднем на 25 % быстрее, чем контрольная (среднее время: 90 мин против 120 мин). Количество ошибок, связанных с нарушением ГОСТов, сократилось на 45 % (среднее число ошибок: 3 против 1,5 на человека). Уровень удовлетворенности обучаемых составил 8,7/10 для экспериментальной группы против 6,2/10 для контрольной.

Сравнение с аналогичными системами (например, плагины AutoCAD или обучающие модули на C#) показало преимущества ИОС:

- гибкость: Python и ruautocad поддерживают кроссплатформенность и интеграцию с современными библиотеками;
- масштабируемость: PostgreSQL превосходит SQLite по производительности при больших объемах данных;
- адаптивность: обратная связь в реальном времени отсутствует в большинстве плагинов.

Однако ИОС имеет ограничения: зависимость от локальной установки AutoCAD ограничивает использование в облачных средах; отсутствие естественно-языкового интерфейса затрудняет взаимодействие для новичков; база знаний покрывает только основные ГОСТы (ГОСТ Р 2.104-2023, ГОСТ Р 2.105-2023), требуя расширения.

Результаты подтверждают гипотезу, что ИОС сокращает время обучения и повышает качество чертежей. Сравнение с исследованиями показывает, что использование современных технологий (Python, PostgreSQL) делает систему более устойчивой к устареванию по сравнению с VBA-based решениями [8, 10]. Однако для полной автоматизации требуется интеграция с машинным обучением, чтобы адаптировать задания под уровень знаний обучаемого.

Перспективы развития

Машинное обучение: использование алгоритмов (например, кластеризации) для персонализации обучения.

Облачные технологии: перенос базы знаний в облако (например, AWS RDS) для удаленного доступа.

Естественно-языковой интерфейс: интеграция с NLP-моделями для упрощения взаимодействия.

Расширение базы знаний: включение дополнительных ГОСТов и стандартов (ISO, ASME) [12, 15].

Заключение

Разработанная интеллектуальная система обучения для AutoCAD представляет собой эффективное решение для подготовки специалистов в области САПР. ИОС, включающая базу знаний ГОСТов, реализована с использованием Python, ruautocad и PostgreSQL, обеспечивая высокую производительность и соответствие стандартам. Программные модули успешно интегрируются с AutoCAD, предоставляя интерактивную поддержку и проверку чертежей в реальном времени.

Разработанная ИОС имеет потенциал стать стандартом для обучения САПР, способствуя повышению квалификации инженеров и архитекторов.

Список литературы

1. Саттарова З.Ф., Ахметов Л.Г. Методика применения системы автоматизированного проектирования AutoCAD для обучения студентов политехнического колледжа // Вестник Марийского государственного университета. 2023. № 3. С. 363–368. URL: <https://cyberleninka.ru/article/n/metodika-primeneniya-sistemy-avtomatizirovannogo-proektirovaniya-autocad-dlya-obucheniya-studentov-politehnicheskogo-kolledzha> (дата обращения: 25.05.2025). DOI: 10.30914/2072-6783-2023-17-3-363-368.
2. Боровский А.В., Сачков Д.И. Методы и алгоритмы разработки САПР для проектирования автоматизированных систем управления технологическими процессами // Современные технологии. Системный анализ. Моделирование. Иркутский государственный университет путей сообщения 2016. № 3 (51). С. 119–122. URL: <https://cyberleninka.ru/article/n/metody-i-algoritmy-razrabotki-saprdlya-proektirovaniya-avtomatizirovannyh-sistem-upravleniya-tehnologicheskimi-protsessami> (дата обращения: 20.05.2025).
3. Jiang A., Sun J., et al. Automation of Design Innovation Process Based on CAD Technology and Reinforcement Learning // Computer Aided Design & Applications. 2024. Vol. 21 (S23). P. 84–99. URL: https://cad-journal.net/files/vol_21/CAD_21%28S23%29_2024_84-99.pdf (дата обращения: 24.05.2025). DOI: 10.14733/cadaps.2024.S23.84-99.
4. Волкова И.Г., Суханов А.А. Подходы к проектированию обучающих систем на основе моделей знаний // Открытое образование. 2021. Т. 25. № 4. С. 68–76. DOI: 10.21686/1818-4243-2021-4-68-76.
5. Баженова И.В., Клунникова М.М., Пак Н.И. Интеллектуальная модель оценки уровня вычислительного мышления обучающихся // Информатика и образование. 2022. Т. 37. № 4. С. 71–79. DOI: 10.32517/0234-0453-2022-37-4-71-79.
6. Chen L., Yu W., Zhang X. Adaptive Learning in CAD Education Using Intelligent Tutoring Systems // Journal of Intelligent & Fuzzy Systems. 2023. Vol. 45, Is. 2. P. 1341–1352. DOI: 10.3233/JIFS-223344.
7. Joshi M.A. Adaptive Learning through Artificial Intelligence // Int. J. Innovative Research in Science, Engineering and Technology. 2023. Vol. 4, Is. 4. P. 001–002. DOI: 10.35248/ijir-set.23.4(1).001-002.
8. Madan A.K., Kharbanda P., Yadav P., Kumar R. AI and Machine Learning Uses in CAD/CAM // International Journal of Research Publication and Reviews. 2023. Vol. 4, Is. 4. P. 4397–4401. URL: <https://ijrpr.com/uploads/V4ISSUE4/IJRPR12061.pdf> (дата обращения: 20.04.2025).
9. Bradáč V., Smolka P., Kotyrbá M., Průdek T. Design of an Intelligent Tutoring System to Create a Personalized Study Plan Using Expert Systems // Applied Sciences. 2022. Vol. 12, Is. 12. Article 6236. URL: <https://www.mdpi.com/2076-3417/12/12/6236> (дата обращения: 01.07.2025). DOI: 10.3390/app12126236.
10. Cerna P.V.G., Perez-Poch A., Penalba F.A., Torner R. Adaptive Learning Web Application to Improve CAD Learning in Engineering // Proceedings of CAD'24. 2024. P. 110–114. DOI: 10.14733/cadconfP.2024.110-114.
11. Uglev V.A. Implementation of Decision Making Mechanism in the Intelligent Tutoring System Based on the Expert Systems Module // Pattern Recognition and Image Analysis. 2024. Vol. 34, Is. 6. P. 744–750. URL: <https://link.springer.com/article/10.1134/S1054661824700615> (дата обращения: 07.07.2025). DOI: 10.1134/S1054661824700615.
12. Lu J. Artificial Intelligence in Education: Design and Evaluation of Adaptive Learning Systems // Region – Educ. Research and Reviews. 2024. Vol. 6, Is. 4. DOI: 10.32629/rrr.v6i4.2101.
13. Rus V., Brusilovsky P., Tamang L.-J., Akhuseynoglu K., Fleming S. Cross-Cutting Support of Making and Explaining Decisions in Intelligent Tutoring Systems Using Cognitive Maps // Intelligent Tutoring Systems: 18th International Conference, ITS 2022. LNCS 13284. Springer, 2022. P. 51–64. [Электронный ресурс]. URL: https://link.springer.com/chapter/10.1007/978-3-031-09680-8_4 (дата обращения: 10.07.2025).
14. Hibbi F.-Z., Abdoun O., Haimoudi E.K. Bayesian Network Modelling for Improved Knowledge Management of the Expert Model in the Intelligent Tutoring System // International Journal of Advanced Computer Science and Applications. 2022. Vol. 13, Is. 6, P. 24–33. URL: <https://thesai.org/Publications/ViewPaper?Volume=13&Issue=6&Code=IJACSA&Serial-No=24> (дата обращения: 10.07.2025). DOI: 10.14569/IJACSA.2022.0130624.
15. Chen X. et al. Automation of Design Innovation Process Based on CAD Technology and RL // CAD Journal. 2024. Vol. 21. S23. P. 84–99. URL: https://cad-journal.net/files/vol_21/CAD_21%28S23%29_2024_84-99.pdf (дата обращения: 10.07.2025).

УДК 004.021

МЕТОДИКА ПРИМЕНЕНИЯ ПСЕВДОПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ

Лелейкин С.С.

*ОАНО ВО «Московский технологический институт», Москва,
e-mail: ssoft@mail.ru*

В данной статье автор предлагает ознакомиться с методикой применения псевдопараллельных вычислительных алгоритмов при решении задач, требующих от программного обеспечения непрерывной работы в режиме реального времени в случае, когда архитектура исполняющей вычислительной машины не позволяет реализовать многопроцессность или мультипоточность. Подобные случаи часто возникают при необходимости оптимизации использования вычислительных мощностей и ограничении их выбора ввиду специфики решения конкретной задачи. Методика базируется на подходе, характеризующемся использованием алгоритма, подразумевающего наличие в программе базового цикла, контролирующего счетчики и связанные с ними триггеры, при срабатывании вызывающие задачи. В зависимости от времени, затрачиваемого на полезную нагрузку, происходит коррекция вызова триггеров, что позволяет программе контролировать и соблюдать время запуска задач. Данная методика была успешно использована автором при разработке и внедрении в промышленную среду системы термоконтроля серверных комнат, выполненной с применением таких компонентов, как микроконтроллерная сборка NodeMCU (ESP8266) и цифровой датчик температуры и влажности DHT11. Задача была выполнена с максимальной утилизацией вычислительных мощностей и рациональным использованием выделенного бюджета, что подкрепляет вышеуказанную методологию практической базой и повышает уровень потенциальной целесообразности ее внедрения и применения в промышленных средах.

Ключевые слова: параллелизм, алгоритмы, прикладная информатика, вычисления, системы реального времени

METHODOLOGY OF APPLICATION OF PSEUDO-PARALLEL ALGORITHMS

Leleykin S.S.

Moscow Technological Institute, Moscow, e-mail: ssoft@mail.ru

In this article, the author suggests getting acquainted with the methodology for using pseudo-parallel computing algorithms in solving problems that require software to work continuously in real time when the architecture of the executing computer does not allow for multiprocessing or multithreading. Such cases often arise when it is necessary to optimize the use of computing power and limit their choice due to the specifics of solving a particular problem. The technique is based on an approach characterized by the use of an algorithm that implies the presence of a basic loop in the program that controls counters and related triggers that trigger tasks. Depending on the time spent on the payload, trigger calls are corrected, which allows the program to monitor and observe the start time of tasks. This technique has been successfully used by the author in the development and implementation in the industrial environment of a thermal monitoring system for server rooms, made using components such as the NodeMCU microcontroller assembly (ESP8266) and the DHT11 digital temperature and humidity sensor. The task was completed with maximum utilization of computing power and rational use of the allocated budget, which supports the above methodology with a practical base and increases the level of potential expediency of its implementation and application in industrial environments.

Keywords: concurrency, algorithms, applied computer science, computing, real-time systems

Введение

В данной статье автор описывает методику применения псевдопараллельных вычислительных алгоритмов, позволяющую решать задачи, требующие обработки информации/вычислений/получения сигналов/изменения состояний в режиме реального времени параллельно друг с другом, но с применением электронной вычислительной машины, архитектура которой подразумевает работу в однопоточном/однопоточном режиме.

Автором были изучены и проанализированы существующие публикации по тематике «Параллельные вычисления» [1–3].

Анализ показал косвенное упоминание подобия псевдопараллельных вычислений некоторыми авторами вышеуказанных статей, но подробный разбор в статьях не приводится, что подчеркивает необходимость изучения и описания научным языком данной тематики для дальнейшего ее применения в промышленных средах.

Данная методика основывается на построении работы программы таким образом, при котором основной алгоритм – повторяющийся цикл с жестко заданным временем выполнения и набором счетчиков, количество которых равно количеству выполняемых задач.

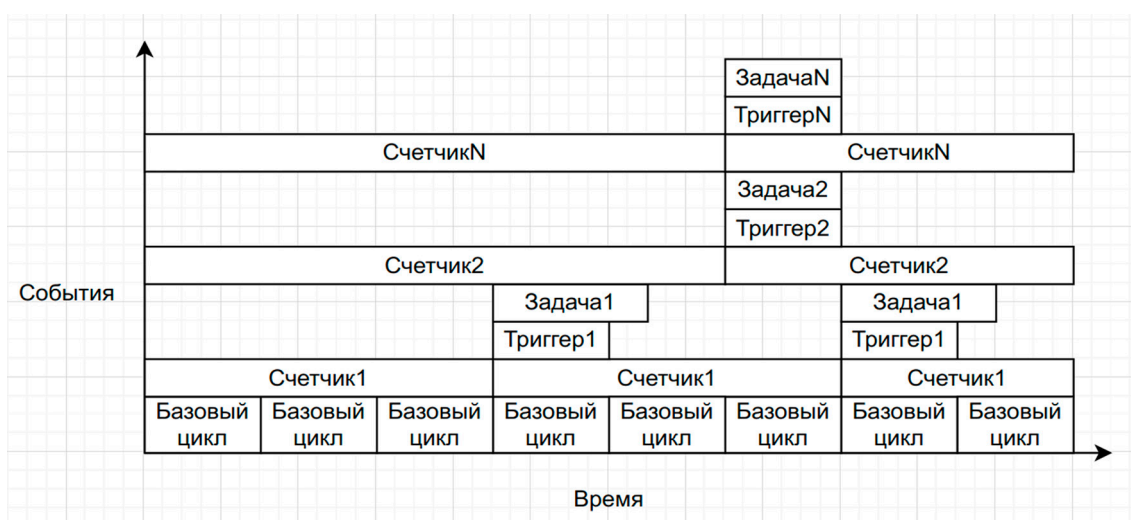


Рис. 1. Диаграмма Ганта, отражающая принцип работы псевдопараллельного алгоритма
Горизонтальная плоскость является временной шкалой, вертикальная – отражает
слои происходящих в тот или иной отрезок времени событий. Нижний слой содержит
блоки повторения базового цикла, над ними расположены слои счетчиков,
связанных с ними триггеров и выполняемых задач
Источник: составлено автором на основе анализа задачи
и последующего синтеза структуры целевого алгоритма

Внутри цикла реализованы проверки, базирующиеся на состоянии счетчиков и, при достижении ими определенных значений, вызывающие необходимые функции. Счетчики инкрементируются при каждом проходе цикла. Так же алгоритм поддерживает коррекцию в зависимости от затраченного на выполнение задач времени. Диаграмма работы алгоритма отражена на рис. 1.

Цель исследования – освоить и описать научным языком методику применения псевдопараллельных алгоритмов для последующего внедрения в промышленные контуры.

Материалы и методы исследования

Изначально исследования проводились автором на базе процесса разработки системы термоконтроля серверных комнат методом анализа, синтеза и моделирования, но довольно быстро были масштабированы и на другие случаи решения задач, связанных с автоматизацией различных процессов (внутренняя автоматизация организации).

В результате исследования было выявлено, что данный алгоритм имеет практическую ценность при построении автоматизированных вычислительных систем реального времени, требующих от вычислительных мощностей параллельного решения различного рода задач, легко применим и повторяем.

Для исследования автором применялись следующие технологии: микроконтроллерная платформа NodeMCU [4], DHT11, скриптовый язык программирования lua [5].

Для удобочитаемости, универсальности и простоты освоения в данной статье будут использованы примеры, реализованные на языке программирования Python [6] в силу его распространенности и легкости восприятия исходного кода.

Результаты исследования и их обсуждение

Результатом исследования является алгоритм, позволяющий реализовать параллельные вычисления на платформах, архитектурно не поддерживающих параллельный запуск нескольких процессорных команд (последовательная платформа).

Итак, непосредственно алгоритм:

1. Базовый цикл. Алгоритм строится вокруг основного цикла, выполняющегося бесконечно. Цикл имеет жестко заданный период выполнения. Внутри цикла находятся счетчики и логика их инкрементирования. Такой цикл в статье будет называться базовым циклом.

В интерпретации языка программирования Python [6] аналогом является следующий код:

```
import time
cyclePeriod = 0.001
counter1 = 0
```



```
counter2 = 0
counter... = 0
while(True):
    counter1 += 1
    counter2 += 1
    counter... += 1
... код с функциональной логикой ...
time.sleep(cyclePeriod)
```

В приведенном примере период равен одной миллисекунде. Счетчики через 1 с примут значение 1000, через 2 с – 2000 и т.д.

2. Задачи. Условия вызова задач определяются внутри тела базового цикла. Каждая задача – функция, на выполнение которой уходит определенное количество времени и которая занимает определенное количество процессорных тактов [7]. Вызов осуществляется при достижении счетчиком функции определенного значения (срабатывает триггер). После выполнения функции счетчик сбрасывается в значение 0.

Расчет триггера выполняется следующим способом:

$$n_{\text{триггер}} = T_{\text{функции}} / T_{\text{базовогоцикла}},$$

где $n_{\text{триггер}}$ – результирующее значение триггера, $T_{\text{функции}}$ – желаемый период запуска функций, $T_{\text{базовогоцикла}}$ – период выполнения базового цикла.

Пример, реализованный на языке программирования Python [6] в контексте кода приведенного в 1 разделе:

```
... тело цикла
# Задача 1 (1 раз в секунду)
if counter1 >= 1000:
    print('1 раз в секунду')
    counter1 = 0

# Задача 2 (2 раза в секунду)
if counter2 >= 500:
    print('2 раза в секунду')
    counter2 = 0
... тело цикла
```

При достижении первым счетчиком значения больше или равно 1000 будет выполнена первая задача, счетчик будет возвращен в значение 0. Аналогично отработает вторая задача при достижении вторым счетчиком значения 500 или больше.

3. Компенсация задержек. Глобальная корректировка счетчиков в результате возникновения задержек при выполнении задач является необходимостью в случае, когда программа должна выполнять задачи с соблюдением временных периодов. Например, первая задача – строго каждые 500 мс, вторая задача – строго каждые 1000 мс и т.д. [8]

За счет сокращения времени цикла достигается отсутствие сдвига из-за затрачиваемого на выполнение задач времени. Если необходимости в соблюдении жестких временных рамок при запуске задач нет, реализация данной функциональности не обязательна, в таком случае периоды между запусками задач будут отличаться в зависимости от количества их вызовов и времени обработки, проще говоря время старта каждой задачи будет «плавать» в зависимости от времени, затраченного на выполнение остальных задач.

Первый этап: необходимо определить время, затраченное на выполнение полезной нагрузки задачи и зафиксировать его. Надежным и автоматическим способом является снятие временной метки непосредственно перед запуском задачи и сразу после ее выполнения. Разница временных меток позволит получить время выполнения задачи:

$$T_{\text{задачи}} = T_{\text{выполнение}} - T_{\text{запуск}}$$

Второй этап: при срабатывании триггера и выполнении вышеуказанной задачи добавить ко всем счетчикам компенсационное значение, равное отношению $T_{\text{задачи}}$ к $T_{\text{базовогоцикла}}$

$$n_{\text{компенс}} = T_{\text{задачи}} / T_{\text{базовогоцикла}},$$

Простыми словами, время до срабатывания всех триггеров сократится на количество тактов, соответствующее времени, которое выполнялась запущенная задача. Таким образом, произойдет компенсация времени выполнения задачи за счет сокращения тактов до запуска триггеров всех задач.

На языке программирования Python [6] описанный выше алгоритм реализуется следующим образом:

```
...
cyclePeriod = 0.001
counter = [0, 0, 0]
# Получение времени в миллисекундах
def current_milli_time():
    return round(time.time() * 1000)
# Корректировка счетчиков
def counterAdjustment(correction, counter):
    counter = [x + correction for x in counter]
    print(f'Корректировка: {correction}')
    return counter
... внутри цикла:
# Задача 1 (1 раз в секунду)
if counter[0] >= 1000:
    preMillis = current_milli_time()
    print('1 раз в секунду')
    ... полезная нагрузка ...
    counter = counterAdjustment(current_milli_time() - preMillis, counter)
    counter[0] = 0
```

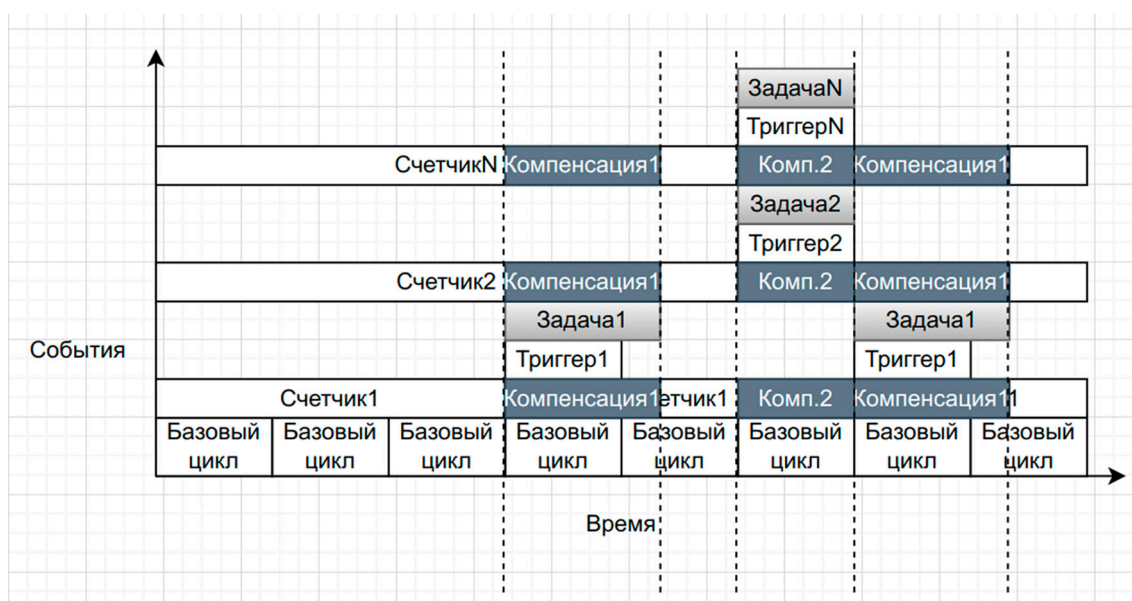


Рис. 2. Диаграмма Ганта, отражающая суть компенсации значения счетчиков при возникновении задержек во время выполнения задач. От блоков задач проведены вертикальные прерывистые линии, иллюстрирующие проекцию времени выполнения задач на счетчики
 Источник: составлено автором на основе реализованного алгоритма с интегрированной компенсацией времени выполнения задач

В приведенном примере $T_{\text{базового цикла}}$ равен 1 мс, соответственно, расчет $\Delta t_{\text{компенсация}}$ достаточно прост: количество тактов на сокращение равно количеству мс, затраченному на выполнение задачи.

Принцип работы компенсационного алгоритма проиллюстрирован на рис. 2.

4. Выполнение задач в фоновом режиме или с отложенными эффектами. При выполнении кода в теле задачи нельзя использовать функции задержки (в случае Python [6] – `time.sleep()`), так как вызов такой функции полностью прервет выполнение про-

граммы на заданный в параметрах период, что остановит базовый цикл и нарушит принцип работы алгоритма.

При необходимости ожидания какого-либо условия или отсчета заданного периода времени внутри задачи, используется подход с флагом запуска ожидания и связкой триггер-счетчик, работающих параллельно с другими задачами и срабатывающих при выполнении заданных условий.

Пример вышеописанного алгоритма, реализованный на языке программирования Python [6]:

```
... Тело цикла
# Задача 3 (запускается каждые 3 секунды, ждет 10 секунд, выполняется)
if waitTaskFlag == False and counter[2] >= 3000: # Запуск ожидания
    print('Ожидающая задача запущена')
    waitTaskFlag = True

if waitTaskFlag == True and waitCounter < 10000: # Ожидание, пока счетчик меньше 10000
    waitCounter += 1

if waitTaskFlag == True and waitCounter >= 10000: # Запуск задачи по истечении 10 секунд
    после запуска
    print('Ожидающая задача выполнена спустя 10 секунд после запуска')
    waitTaskFlag = False
    waitCounter = 0
    counter[2] = 0
... Тело цикла
```

Этапы алгоритма реализации задержки
при необходимости ее выполнения внутри задачи

Шаг	Описание	Условия выполнения
1	Запуск ожидания Поднимает флаг ожидания	Если флаг ожидания опущен и триггер базового счетчика сработал (при необходимости)
2	Ожидание Инкрементирует счетчик ожидания	Если флаг ожидания поднят и триггер счетчика ожидания не сработал
3	Запуск задачи Сброс флага, сброс базового счетчика и счетчика ожидания	Если флаг ожидания поднят и триггер счетчика ожидания сработал

Источник: составлено автором на основе подхода собственной разработки, функционирующего за счет системы, состоящей из флага запуска ожидания и связки триггер – счетчик, работающих параллельно с другими задачами и срабатывающих при выполнении заданных условий. Назначение: более структурированно представить алгоритм, описанный выше.

Алгоритм, лежащий в основе данного кода, описан более подробно в таблице.

5. Пример кода. Ниже приведен полный листинг кода программы, основанной на использовании псевдопараллельного вычислительного алгоритма.

```
import time
import math

cyclePeriod = 0.001
counter = [0, 0, 0]
waitTaskFlag = False
waitCounter = 0

# Получение времени в миллисекундах
def current_milli_time():
    return round(time.time() * 1000)

# Корректировка счетчиков
def counterAdjustment(correction, counter):
    counter = [x + correction for x in counter]
    print(f'Корректировка: {correction}')
    return counter

# Основной цикл
while(True):
    # Инкремент счетчика
    counter = [x + 1 for x in counter]

    # Задача 1 (1 раз в секунду)
    if counter[0] >= 1000:
        preMillis = current_milli_time()
        print('1 раз в секунду')

        for x in range(100000):
            temp = math.cos(999999)

        counter = counterAdjustment(current_milli_time() - preMillis, counter)
        counter[0] = 0

    # Задача 2 (2 раза в секунду)
    if counter[1] >= 500:
        preMillis = current_milli_time()
        print('2 раза в секунду')
        counter = counterAdjustment(current_milli_time() - preMillis, counter)
        counter[1] = 0
```



```
# Задача 3 (запускается каждые 3 секунды, ждет 10 секунд, выполняется)
if waitTaskFlag == False and counter[2] >= 3000: # Запуск ожидания
    print('Ожидающая задача запущена')
    waitTaskFlag = True

if waitTaskFlag == True and waitCounter < 10000: # Ожидание, пока счетчик меньше 10000
    waitCounter += 1

if waitTaskFlag == True and waitCounter >= 10000: # Запуск задачи по истечении 10 секунд
    после запуска
    print('Ожидающая задача выполнена спустя 10 секунд после запуска')
    waitTaskFlag = False
    waitCounter = 0
    counter[2] = 0

time.sleep(cyclePeriod)
```

В коде приведенной программы импортированы две библиотеки: `time` и `math`. Обе библиотеки входят в базовую поставку дистрибутива Python, применяются для работы со временем (`time`) и выполнения математических операций (`math`). Библиотеки, предназначенные для многопоточных и многоядерных вычислений [9] (например, популярные библиотеки `threads` и `multiprocessing`), в программе не использованы.

Заключение

Разработанная и описанная автором статьи методика, построенная на псевдопараллельном подходе к разработке программного обеспечения, включающая такие понятия, как базовый цикл, счетчики, триггеры, корректировка и задачи, успешно зарекомендовала себя при практическом использовании в промышленной среде и рекомендуется автором к освоению и повторению.

Новизна результатов исследования подтверждается отсутствием ранее опубликованных минимально схожих материалов, столь подробно описывающих применение подобных алгоритмов в промышленной эксплуатации.

Представленный псевдопараллельный алгоритм применим в различных встраиваемых системах, требующих использования микроконтроллерных вычислительных платформ, но также может быть применен и при разработке программного обеспечения для других видов ЭВМ, например, при невозможности использования по ряду причин библиотек, реализующих многопоточные или многоядерные вычисления.

Список литературы

1. Клименко В.И. Параллельные вычисления и создание параллельных программ // Научные труды КубГТУ. 2016. № 15. URL: <https://ntk.kubstu.ru/data/mc/0036/1298.pdf> (дата обращения: 02.06.2025).
2. Волосова А.В. Параллельные методы и алгоритмы: учебное пособие М.: МАДИ 2020. [Электронный ресурс]. URL: <https://lib.madi.ru/fel/fel1/fel20E533.pdf> (дата обращения: 02.06.2025).
3. Хардилов М.В., Эминджонов Д.Е. Обзор архитектур многопроцессорных и многопоточных систем // Научный Лидер. 2025. № 6 (207). С. 46–48. URL: https://scilead.ru/media/journal_pdf_207.pdf (дата обращения: 12.05.2025).
4. Гусев В.В., Гусев И.В., Христофоров Р.П., Домрачева Т.С. Интернет вещей и устройства, подключаемые к интернету // Аллея науки. 2018. Т. 2. № 11 (27). С. 859–865. URL: https://alley-science.ru/domains_data/files/Journal_Dec18/2%20tom%20Dekabr.pdf?x14474 (дата обращения: 03.06.2025).
5. Волков В.Д. Сравнительный анализ языков программирования на основе решения тестовой задачи сортировки данных // Вестник РГТУ. Серия «Информатика. Информационная безопасность. Математика». 2023. № 3. С. 61–70. URL: <https://www.rsuh.ru/vestnik/digest/informatika-informatsionnaya-bezopasnost-matematika/informatika-informatsionnaya-bezopasnost-matematika-3-2023.php> (дата обращения: 12.05.2025).
6. Таршхоева Ж.Т. Язык программирования Python. Библиотеки Python // Молодой ученый. 2021. № 5 (347). С. 20–21. URL: <https://moluch.ru/archive/347/78102/> (дата обращения: 12.05.2025).
7. Фролов В., Галактионов В., Санжаров В. RISC-V: стандарт, изменивший мир микропроцессоров // Открытые системы. СУБД. 2020. № 2. С. 30–34. URL: <https://www.osp.ru/os/2020/02/13055471> (дата обращения: 12.05.2025).
8. Глонина А.Б. Обобщенная модель функционирования модульных вычислительных систем реального времени для проверки допустимости конфигураций таких систем // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2017. Т. 6. № 4. С. 43–59. URL: <https://cyberleninka.ru/article/n/obobshchennaya-model-funktsionirovaniya-modulnyh-vychislitelnyh-sistem-realnogo-vremeni-dlya-proverki-dopustimosti-konfiguratsiy> (дата обращения: 12.05.2025).
9. Шавтикова Л.М., Текеев М.Б. Параллельное вычисление потоков на языке программирования Python // Тенденции развития науки и образования. 2020. № 68–1. С. 153–156. URL: <https://doicod.ru/doi/10.12-2020-46.pdf> (дата обращения: 12.05.2025).

УДК 004.4

СРАВНЕНИЕ ПОДХОДОВ К ВЕБ-РАЗРАБОТКЕ: ПРЕИМУЩЕСТВА И ОГРАНИЧЕНИЯ NO-CODE И ТРАДИЦИОННОГО ПРОГРАММИРОВАНИЯ

Пепельшев Д.И., Кулагина Е.С.

ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина», филиал, Нижний Тагил, e-mail: dmitry.pepelyshev@urfu.ru

Исследование направлено на сопоставительный анализ no-code платформ и традиционных методов программирования при разработке веб-сайтов. В ходе данной работы было проведено комплексное сравнение технологических и эксплуатационных параметров рассматриваемых подходов, что позволило выявить их преимущества и ограничения. Экспериментальная часть исследования включала сравнительный анализ платформы Tilda и традиционного стека технологий (HTML, CSS, JavaScript). Основными критериями оценки стали следующие параметры: производительность, уровень безопасности, кастомизация, масштабирование и перспективы долгосрочной поддержки. Результаты проведенного анализа показывают, что no-code решения обладают рядом существенных технологических ограничений, недостатками данных платформ являются: сниженная производительность из-за генерации избыточного кода, наличие уязвимостей в системе защиты данных, ограниченные возможности для оптимизации. На основании полученных данных можно сделать вывод о том, что no-code технологии не способны полностью заменить профессиональные методы веб-разработки, практическое применение таких платформ целесообразно ограничить созданием простых сайтов и небольших краткосрочных проектов. Для реализации сложных коммерческих веб-сайтов, требующих надежности, высокого уровня безопасности и архитектурной гибкости, традиционные подходы программирования сохраняют неоспоримые преимущества как с точки зрения качества конечного продукта, так и в аспекте долгосрочной поддержки.

Ключевые слова: no-code платформы, Tilda, традиционное программирование, сравнительный анализ, сайт

COMPARISON OF APPROACHES TO WEB DEVELOPMENT: ADVANTAGES AND LIMITATIONS OF NO-CODE AND TRADITIONAL PROGRAMMING

Pepelyshev D.I., Kulagina E.S.

Ural Federal University named after the first President of Russia B.N. Yeltsin, Nizhny Tagil branch, Nizhny Tagil, e-mail: dmitry.pepelyshev@urfu.ru

The research is aimed at a comparative analysis of no-code platforms and traditional programming methods in website development. In the course of this work, a comprehensive comparison of the technological and operational parameters of the considered approaches was carried out, which made it possible to identify their advantages and limitations. The experimental part of the study included a comparative analysis of the Tilda platform and the traditional technology stack (HTML, CSS, JavaScript). The main evaluation criteria were the following parameters: performance, security level, customization, scaling, and prospects for long-term support. The results of the analysis show that no-code solutions have a number of significant technological limitations, the disadvantages of these platforms are: reduced performance due to redundant code generation, vulnerabilities in the data protection system, and limited optimization opportunities. Based on the data obtained, it can be concluded that no-code technologies are not able to completely replace professional web development methods, and it is advisable to limit the practical use of such platforms to creating simple websites and small short-term projects. To implement complex commercial websites that require reliability, high levels of security, and architectural flexibility, traditional programming approaches retain undeniable advantages both in terms of end product quality and long-term support.

Keywords: no-code platforms, Tilda, traditional programming, comparative analysis, website

Введение

В современном мире веб-сайты используются во всех сферах человеческой деятельности, их функциональное разнообразие позволяет решать широкий спектр как профессиональных, так и бытовых задач. На данный момент сайты перестали быть только источниками информации, превратившись в многофункциональные платформы для бизнеса и коммуникации [1, с. 15].

Данная трансформация обусловила появление новых требований к характери-

стикам сайтов. Современные веб-ресурсы должны сочетать в себе адаптивность для различных устройств, интерактивные элементы управления, реализацию возможности интеграции с внешними сервисами. При этом сохраняются и базовые требования к юзабилити: скорость загрузки, интуитивность интерфейса и безопасность данных.

Современные веб-сайты, представляющие собой сложные системы взаимосвязанных страниц с единой структурой, могут создаваться разными способами. Тради-

ционный подход предполагает разработку с использованием HTML, CSS и JavaScript и требует от разработчиков наличия профессиональных знаний и навыков. В качестве альтернативы выступают онлайн-конструкторы сайтов, позволяющие быстро создавать готовые веб-сайты и предлагающие интуитивно понятные интерфейсы с готовыми шаблонами и примерами, что значительно снижает порог входа для непрофессиональных разработчиков [2, с. 55]. Выбор между подходами определяется комплексом ключевых факторов: уровень функциональности разрабатываемого продукта, бюджетные ограничения, временные рамки проекта и техническая подготовка команд разработчиков.

No-code представляет собой инновационную методологию проектирования и разработки веб-приложений, основанную на применении специализированных платформ с графическим интерфейсом [3, с. 184]. Данный подход включает использование интерфейса drag-and-drop, который обеспечивает возможность создания функциональных веб-приложений с помощью применения интерактивной технологии перемещения элементов без написания программного кода [4, с. 396].

Использование no-code платформ для создания веб-сайтов предполагает четкую структуру работы. На начальном этапе разработчик выбирает подходящие шаблоны из обширной библиотеки компонентов. Следующим шагом является этап визуальной кастомизации, на котором каждый элемент интерфейса изменяется и адаптируется под конкретные требования проекта. Завершающим этапом разработки становится наполнение созданной структуры текстовым, графическим и мультимедийным контентом [3, с. 184].

Одним из преимуществ использования no-code решений является их архитектура, значительная часть сложных процессов в которой автоматизирована. Платформа выполняет следующие backend-задачи: управление данными, организация API-взаимодействия, обеспечение базовых механизмов безопасности.

Данный подход существенно снижает порог вхождения в веб-разработку, позволяя создавать функциональные продукты специалистам без глубоких технических знаний. Однако следует отметить, что no-code решения могут иметь ограничения в плане гибкости, обеспечения безопасности и масштабируемости по сравнению с традиционными методами разработки [3, с. 184].

В данной работе была рассмотрена одна из самых популярных существующих no-

code платформ Tilda, которая представляет собой конструктор сайтов, основанный на использовании готовых блоков и визуального редактора. Tilda предоставляет разработчикам веб-сайтов готовые композиционные решения и шаблоны, требующие последующей кастомизации под конкретные запросы пользователя [5, с. 157]. Также следует отметить, что значительная часть функционала, включая специализированные блоки, становится доступной только после активации платной подписки, бесплатная версия позволяет создать один сайт с базовым набором функций, небольшим количеством страниц и ограниченными возможностями хостинга.

Для проектов, требующих нестандартных дизайнерских решений, в Tilda может использоваться Zero Block. Данный инструмент доступен в бесплатной версии конструктора сайтов, он позволяет проектировать и кастомизировать элементы с уникальным дизайном, регулировать параметры анимации и обеспечивать детализированную проработку каждого виджета интерфейса, а также предоставляет возможность настройки адаптации контента для различных устройств [6, с. 132].

Tilda является сбалансированным no-code решением, сочетающим в себе интуитивно понятный интерфейс и набор инструментов для профессиональной верстки, она отлично подходит для быстрого создания простых веб-сайтов. Ключевым преимуществом конструктора сайтов является возможность быстрого запуска визуально привлекательных сайтов без необходимости написания программного кода. Однако существующие ограничения платформы снижают ее применимость для разработки сложных высоконагруженных проектов, требующих глубокой кастомизации, высокого уровня безопасности и масштабируемости.

На данный момент создание профессиональных веб-приложений основывается на трех ключевых технологиях: HTML используется для структурирования страницы, CSS – для оформления внешнего вида, JavaScript – для реализации динамического поведения и интерактивности. Данные технологии образуют фундамент современной веб-разработки, независимо от сложности и назначения создаваемого приложения [7].

HTML (HyperText Markup Language) является стандартным языком разметки, используемым для структурирования содержимого веб-страниц. Данная технология основана на системе тегов – специальных маркеров, определяющих тип и назначение элементов документа. Посредством те-

гов осуществляется разметка компонентов веб-страницы, включая текстовые блоки, мультимедийные элементы, навигационные компоненты, структурные единицы и другие элементы [7]. Современная спецификация HTML5 позволяет расширить базовые возможности языка, дополняя их семантическими тегами и встроенной поддержкой мультимедиа и сохраняя при этом принцип разделения структуры, оформления и поведения веб-страниц.

CSS (Cascading Style Sheets) представляет собой язык описания стилей, предназначенный для визуального оформления веб-страниц. Данная технология обеспечивает возможность детальной настройки внешнего вида HTML-элементов, включая параметры типографики (шрифты, кегль, начертание), цветовые схемы (фон, текстуры), пространственное расположение компонентов (размеры, позиционирование, отступы) и другие параметры [8, с. 121].

Главной отличительной особенностью CSS является реализация концепции разделения содержания и представления веб-страниц. Данная технология позволяет создавать адаптивные интерфейсы, автоматически подстраивающиеся под различные устройства и форматы отображения. Важным преимуществом использования CSS в веб-разработке является архитектура, предполагающая вынесение стилевых правил в отдельные файлы, что обеспечивает эффективное кэширование и существенно упрощает процесс поддержки и обновления дизайна [8, с. 121–122].

Следует отметить, что CSS обладает рядом ограничений. Наиболее значимым является проблема кросс-браузерной совместимости, заключающаяся в том, что различные браузеры могут по-разному интерпретировать одинаковые стилевые правила. Еще одним ограничением является тесная взаимосвязь между CSS-селекторами и HTML-структурой, которая требует синхронного внесения изменений в оба компонента, что усложняет процесс разработки в некоторых сценариях. Несмотря на эти ограничения, CSS остается неотъемлемой частью современной веб-разработки.

JavaScript представляет собой высокоуровневый язык сценариев, выполняемый на стороне клиента и предназначенный для обеспечения интерактивности веб-страниц. Данная технология позволяет динамически изменять содержимое и поведение веб-страниц, реагируя на действия пользователя.

К ключевым преимуществам JavaScript относится универсальность, которая заключается в том, что язык позволяет реализо-

вывать как простые элементы интерфейса, так и сложную клиентскую логику, поддерживая при этом интеграцию с многочисленными API и сторонними сервисами. Также важным достоинством является кроссплатформенность, обеспечивающая выполнение программного кода в большинстве современных браузеров без необходимости адаптации под конкретные устройства или операционные системы. К дополнительным преимуществам можно отнести динамическое обновление контента без перезагрузки страницы и открытую лицензионную модель [9, с. 28].

Основным недостатком JavaScript является сложность написания и поддержки кода. К основным сложностям можно отнести асинхронную природу языка, динамическую типизацию и гибкость синтаксиса, которые могут приводить к трудностям в отладке кода и необходимости написания дополнительных тест-кейсов. Также стоит отметить, что JavaScript выполняется на стороне клиента и имеет уязвимости в виде межсайтовой подделки запросов, js-инъекций, межсайтового скриптинга. Для предотвращения этих угроз необходимо использовать CSP (Content Security Policy), настраивать механизмы аутентификации и авторизации пользователей, использовать современные API и экранировать пользовательский ввод [9, с. 28–29].

Цель исследования – проведение комплексного сравнительного анализа двух подходов к веб-разработке: применение no-code платформы Tilda и традиционного метода программирования на базе HTML, CSS и JavaScript.

Материалы и методы исследования

Для проведения исследования был использован сравнительно-аналитический метод, обеспечивающий всестороннее сопоставление ключевых параметров no-code платформы Tilda и традиционного подхода к веб-разработке на основе HTML, CSS и JavaScript.

Результаты исследования и их обсуждение

Современные веб-сайты могут создаваться с помощью двух абсолютно разных подходов, которые имеют свои преимущества, недостатки, особенности и ограничения. В результате проведенного сравнительного анализа ключевых аспектов каждого из рассматриваемых методов была составлена сравнительная таблица, наглядно показывающая разницу между no-code платформой Tilda и традиционной разработкой на базе HTML, CSS и JavaScript.

Сравнительная характеристика no-code платформы Tilda
и традиционной веб-разработки

Критерий сравнения	No-code платформа Tilda	Традиционная разработка на базе HTML, CSS и JavaScript
Производительность	Средняя (избыточный сгенерированный программный код)	Высокая (оптимизированный программный код)
Безопасность	Базовый уровень	Высокий уровень
Кастомизация	Ограниченная (в рамках возможностей платформы)	Полная (неограниченные возможности)
Масштабируемость	Ограниченная	Высокая
Долгосрочная поддержка	Ограниченная	Неограниченная
Стоимость разработки	Платная подписка для доступа к полному функционалу платформы	Бесплатные технологии
Скорость разработки	Высокая (используются готовые шаблоны и блоки)	Низкая (требуется написание программного кода)
Порог вхождения	Низкий (не требуются навыки программирования)	Высокий (необходимы технические знания и навыки программирования)
Адаптивность	Хорошая (используются встроенные адаптивные шаблоны)	Полная (точный контроль адаптации)
Интеграция с API	Ограниченная	Полноценная
Сферы применения	Лендинги, сайты-визитки, небольшие проекты	Проекты любого уровня сложности и масштаба

Данный анализ позволил выявить технологические возможности, преимущества и ограничения каждого из подходов, а также показал, что no-code платформы и традиционный подход к созданию веб-сайтов сильно отличаются между собой и могут успешно использоваться в зависимости от уровня сложности разрабатываемого проекта, наличия ресурсов и предъявляемых требований.

Заключение

В результате проведенного исследования были определены ключевые различия между no-code решениями и традиционными подходами к веб-разработке. No-code платформы, например Tilda, могут быть успешно применимы для создания простых проектов, для которых важна оперативная разработка рабочей и визуально оформленной версии продукта в кратчайшие сроки и минимальные требования к техническим навыкам разработчиков. Основные преимущества данных платформ заключаются в наличии интуитивно понятного интерфейса и готовой библиотеки шаблонов, а также автоматизации сложных backend-процессов. Технологическими ограничениями no-code решений являются неполные возможности кастомизации при использовании бесплатной подписки, низкая производительность и недостаточный уровень безопасности, а также прямая зависимость от функциональности выбранной платформы.

Традиционные методы программирования на базе HTML, CSS и JavaScript чаще всего используются для реализации сложных коммерческих проектов, но также могут быть применимы и для создания более простых веб-сайтов. К преимуществам данного подхода можно отнести наличие полного контроля над программным кодом, поддержку возможности глубокой оптимизации и кастомизации, обеспечение высокого уровня безопасности и масштабирования. Разработка с помощью традиционных методов программирования требует специальных технических знаний и навыков программирования от разработчиков и длительные сроки для реализации проектов.

Таким образом, каждый из рассматриваемых подходов может быть успешно применен для реализации веб-сайтов. No-code платформы оптимально используются для создания простых маркетинговых лендингов, целевых страниц, интернет-журналов, корпоративных сайтов-визиток и небольших интернет-магазинов. Традиционные методы программирования остаются незаменимыми для реализации сложных коммерческих систем, высоконагруженных проектов и решений, требующих уникальной функциональности, высокого уровня безопасности и масштабируемости.

Выбор подхода к реализации веб-ресурса должен основываться на оценке проектных требований, которые включают в себя сроки реализации, бюджетные огра-

ничения, необходимый уровень функциональности, безопасности и оптимизации, уровень знаний разработчиков. В некоторых случаях рациональным решением может являться применение гибридного подхода, сочетающего в себе преимущества каждого из рассмотренных в исследовании методов.

Список литературы

1. Антонов В.Ф. Анализ характеристик современных фреймворков // Научный альманах. 2020. № 9–2 (71). С. 15–17. URL: <https://www.elibrary.ru/item.asp?id=44120664> (дата обращения: 15.05.2025).
2. Павлушина В.А., Мостяева И.В. Создание и разработка персонального сайта учителя с использованием современных конструкторов сайтов // Информатика и прикладная математика. 2021. № 27. С. 54–57. URL: <https://elibrary.ru/item.asp?id=47398282> (дата обращения: 25.05.2025).
3. Яблокова А.А., Ковалев И.В. Обзор исследований в области применения инструментов проектирования и создания информационных систем на базе стека frontend, backend и fullstack технологий // Перспективы молодежной науки. 2023. Т. 10. С. 183–186. URL: <https://www.elibrary.ru/item.asp?id=54511324> (дата обращения: 15.05.2025).
4. Сулягина Е.А. Технология Drag and Drop как инструмент формирования учебно-познавательной деятельности

на уроках математики // Современные научные исследования и разработки. 2016. № 3 (3). С. 395–397. URL: <https://elibrary.ru/item.asp?id=26464173> (дата обращения: 25.05.2025).

5. Турдубаева Ж.А., Исманов О.М., Саланова А.Ж. Создание сайта автосалона с помощью конструктора сайтов Tilda Publishing // Бюллетень науки и практики. 2025. Т. 11. № 4. С. 156–159. URL: <https://elibrary.ru/item.asp?id=80668802> (дата обращения: 25.05.2025). DOI: 10.33619/2414-2948/113/23.

6. Эркенова М.У., Бытдаева Л.И., Хатуаев Т.А. Особенности использования web-конструктора Tilda при разработке web-проектов для малого бизнеса // Тенденции развития науки и образования. 2022. № 92–6. С. 131–133. URL: <https://elibrary.ru/item.asp?id=50398204> (дата обращения: 25.05.2025). DOI: 10.18411/trnio-12-2022-299.

7. Домбровский Я.А., Ткачев Е.В. Создание веб-приложения средствами HTML, CSS, JavaScript // Дневник науки. 2025. № 3. URL: https://dnevniknauki.ru/images/publications/2025/3/technics/Dombrovsky_Tkachev.pdf (дата обращения: 25.05.2025).

8. Новиков С.А. CSS-каскадные таблицы стилей. Внедрение CSS в html // Информационные технологии и прикладная математика. 2017. С. 121–124. URL: <https://elibrary.ru/item.asp?id=30515921> (дата обращения: 25.05.2025).

9. Галимуллин Н.Р. Javascript: разработка интерактивных веб-страниц с помощью javascript // Экономика и управление: проблемы, решения. 2024. Т. 9. № 9 (150). С. 27–32. URL: <https://elibrary.ru/item.asp?id=71727936> (дата обращения: 25.05.2025). DOI: 10.36871/ek.up.p.r.2024.09.09.004.

УДК 004.738.5

РАЗРАБОТКА МНОГОФУНКЦИОНАЛЬНОЙ ПЛАТФОРМЫ ДЛЯ АГРЕГИРОВАНИЯ НОВОСТНЫХ СТАТЕЙ

Кольева Н.С., Хасимьянова И.В., Кольев И.Н., Гонцова А.О.

ФГБОУ ВО «Уральский государственный экономический университет»,
Екатеринбург, e-mail: kolyeva_ns@usue.ru.ru

В статье рассматривается разработка многофункциональной платформы для агрегирования, фильтрации и анализа новостных статей из различных источников. Цель исследования – создание системы, обеспечивающей быстрый доступ к актуальной и проверенной информации, автоматическую классификацию контента и персонализацию для пользователей. В качестве материалов и методов исследования использованы современные веб- и мобильные технологии, включая React Native для клиентской части и NewsAPI для серверной части. Авторы провели SWOT-анализ существующих новостных агрегаторов, таких как Google News и Яндекс.Новости, выявив их преимущества и недостатки. В результате разработки система обеспечила автоматизированный сбор, обработку и визуализацию новостных данных, а также внедрила механизмы проверки достоверности информации и защиты персональных данных. Проведенное исследование демонстрирует, что предложенная платформа способна эффективно решать проблемы информационного шума, дублирования контента и отсутствия персонализации, что делает ее ценным инструментом для оперативного получения и анализа новостей. Важным результатом является создание масштабируемой и безопасной системы, адаптированной под современные требования медиапотребления. Разработанное решение открывает перспективы для дальнейшего совершенствования алгоритмов обработки данных и расширения функциональных возможностей платформы.

Ключевые слова: агрегирование новостей, новостная платформа, обработка данных, фильтрация новостей, персонализация

DEVELOPMENT OF A MULTIFUNCTIONAL PLATFORM FOR AGGREGATION OF NEWS ARTICLES

Koleva N.S., Khasimyanova I.V., Kolev I.N., Gontsova A.O.

Ural State University of Economics, Yekaterinburg, e-mail: kolyeva_ns@usue.ru.ru

The article discusses the development of a multifunctional platform for aggregating, filtering and analyzing news articles from various sources. The purpose of the study is to create a system that provides quick access to relevant and verified information, automatic content classification and personalization for users. Modern web and mobile technologies were used as materials and research methods, including React Native for the client part and NewsAPI for the server part. The authors conducted a SWOT analysis of existing news aggregators, such as Google News and Yandex.News, identifying their advantages and disadvantages. As a result of the development, the system provided automated collection, processing and visualization of news data, and also implemented mechanisms for verifying the reliability of information and protecting personal data. The study demonstrates that the proposed platform can effectively solve the problems of information noise, duplicate content and lack of personalization, which makes it a valuable tool for promptly receiving and analyzing news. An important result is the creation of a scalable and secure system adapted to modern media consumption requirements. The developed solution opens up prospects for further improvement of data processing algorithms and expansion of the platform's functionality.

Keywords: news aggregation, news platform, data processing, news filtering, personalization

Введение

В современном цифровом мире ежедневно публикуются миллионы новостных материалов. Эта информация поступает из множества источников: официальные СМИ, независимые новостные платформы, блоги, социальные сети. Пользователь, желающий получать актуальную информацию по интересующим темам, сталкивается с проблемой информационного шума, дублирующихся материалов и нерелевантного контента.

В связи с этим появляется острая потребность в создании инструментов, которые способны автоматизированно агрегировать, фильтровать, структурировать и

персонализировать новостной поток. Такие системы позволяют не только экономить время пользователя, но и повышать качество принимаемых решений – особенно в профессиональной среде (аналитика, PR, управление рисками, финансы) [1].

Цель исследования – оптимизация процессов сбора, обработки и представления новостных данных для повышения их актуальности и удобства использования.

Материалы и методы исследования

Для разработки многофункциональной платформы для агрегирования новостных статей использованы следующие материалы: источники новостей: официальные

сайты новостных агентств (РИА Новости, TACC, BBC, CNN), новостные агрегаторы (GoogleNews, Yandex.News, News360), RSS-ленты. Для обработки и анализа текстовой информации взаимодействия в API выбраны библиотеки: NewsAPI, BeautifulSoup, NLTK, TensorFlow. Использование всех этих источников обеспечивает широкий охват актуальных новостных данных для последующего анализа и агрегирования на платформе.

В качестве методов исследования выбраны следующие: интервью и анкетирование применялись для определения спецификаций и требований проекта и потребностей пользователей; моделирование систем необходимо для выбора технологий и инструментов для разработки; методы парсинга – для обработки данных, методы машинного обучения и NLP – для классификации и фильтрации новостей; метрики precision, recall применялись для оценки точности агрегирования и скорости обработки новостей.

Результаты исследования и их обсуждение

В настоящее время существуют разные типы агрегаторов: некоторые управляют на основе предпочтений пользователей, например Digg; другие, такие как The Huffington Post или The Daily Beast, сочетают автоматическую агрегацию новостей с созданием собственного оригинального контента, включая обсуждения и «гражданскую» журналистику [2].

Несмотря на популярность таких новостных сервисов, как Google News, Яндекс.Новости, Flipboard и News360, у них имеются ограничения: недостаток расширенной персонализации и возможности

сохранять пользовательские настройки, невозможность вручную формировать подборки, закрытая архитектура без опций расширения и лицензионные ограничения на использование данных.

Обратим внимание на ресурсы, которые систематизируют новостные потоки полностью без участия человека. Такие платформы претендуют на создание цельной и объективной картины событий, позволяя читателю ознакомиться с различными точками зрения по интересующей теме. В этом случае автоматизация процесса агрегации новостей полностью реализуется программным обеспечением. Даже выделение ключевых событий и формирование подборок новостей может осуществляться машинным способом. В таблице представим сильные и слабые стороны популярных новостных сервисов.

В 2025 г., согласно данным Mediascope, телевидение сохраняет значительную аудиторию, с охватом 65 % населения в среднем за день, а охват за месяц достигает 99 % россиян старше 4 лет. Среднее время просмотра ТВ на человека – 3 ч 36 мин в день, при этом у старшего поколения (65+) этот показатель достигает 7 ч 13 мин в день. Интернет также остается важным источником информации, с среднесуточным охватом 84 % россиян старше 12 лет и средним временем использования 4 ч 33 мин на человека в день. При этом молодежь в возрасте 12–24 лет тратит на интернет в среднем 6 ч 49 мин в день, а старшее поколение 65+ – 2 ч 2 мин. Аудитория прессы (газеты и журналы) в 2025 г., согласно данным Mediascope за полгода (сентябрь 2024 – февраль 2025), составляет 56 % населения старше 16 лет, при этом аудитория среднего номера – 27 % россиян [3].

SWOT-анализ существующих агрегаторов

Сервис	Преимущества	Недостатки
Google News	– использует AI для персонализации; – огромный охват источников; – удобный интерфейс	– алгоритмы закрыты; – недостаточная поддержка нишевых и региональных СМИ
Яндекс.Новости	– хороший охват русскоязычных и региональных источников; – быстрая доставка новостей	– слабая персонализация; – меньший охват по сравнению с Google News
Feedly	– гибкая настройка через RSS; – пользователь сам выбирает источники; – поддержка интеграций	– нет автоматической персонализации; – требуется ручная настройка, не подходит новичкам
Flipboard	– привлекательный визуальный стиль; – журнальный формат подачи; – тематическая сортировка	– упор на дизайн, а не на аналитику; – меньше акцента на релевантность и фильтрацию
News360	– глубокая персонализация на основе интересов; – машинное обучение для отбора новостей	– ограниченное количество источников; – менее популярен, меньше обновлений и поддержка

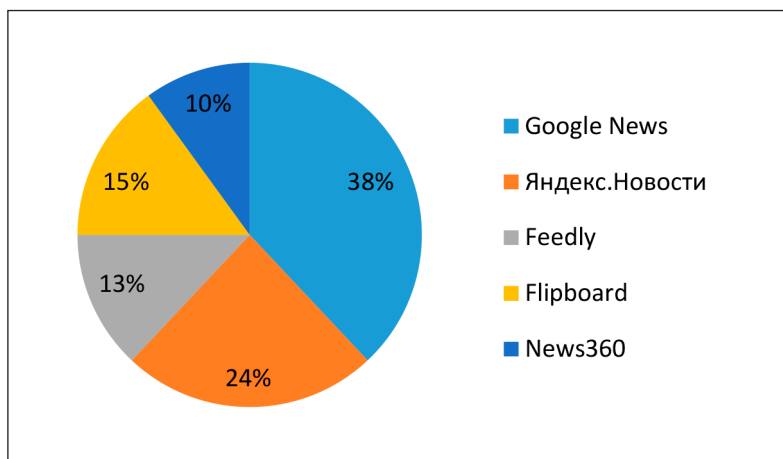


Диаграмма «Популярные агрегаторы новостей»

На основе вышеперечисленных данных разработана диаграмма популярности новостных источников в соответствии с рисунком.

Информационная система агрегации новостных статей реализует следующие функциональные модули:

- Мониторинг новостных потоков – автоматизированный сбор данных из множества источников с целью формирования репрезентативной выборки актуальных событий и информационных сообщений.

- Контент-анализ – процесс тематической категоризации, идентификации ключевых терминов и концептов, а также фильтрации данных на основе заданных критериев (тематических, географических, временных и источниковых).

- Персонализация контента – алгоритмическая адаптация новостной ленты в соответствии с пользовательскими предпочтениями, обеспечивающая повышение релевантности выводимой информации и создание персонализированного информационного пространства.

- Управление пользовательскими данными – механизмы регистрации, аутентификации и настройки профиля, гарантирующие безопасное хранение персональных данных и их использование для оптимизации взаимодействия пользователя с системой.

- Визуализация данных – представление новостного контента в графическом формате (карточки, ленты, дашборды), что улучшает восприятие и аналитическую обработку информации.

Эти процессы автоматизируются с использованием современных веб- и мобильных технологий, а также API-интерфейсов, предоставляющих доступ к глобальной базе новостей.

Автоматизация агрегирования новостных статей из множества источников направлена на создание системы, которая позволит собирать, обрабатывать и предоставлять пользователям актуальную, структурированную и персонализированную информацию. Система должна сократить время на поиск новостей, повысить точность анализа данных и минимизировать участие человека в рутинных процессах [4].

Проблемы, требующие автоматизации, охватывают несколько ключевых аспектов. Одним из них является фрагментация информации: пользователям приходится переключаться между разными платформами, что затрудняет формирование целостной картины событий. Также актуальна проблема больших объемов данных, так как ручная обработка новостных материалов невозможна из-за их масштабов и скорости появления.

Дополнительно возникает проблема дублирования данных, когда одни и те же новости публикуются на различных платформах, что приводит к информационному шуму. Еще один важный аспект – достоверность информации, требующая автоматизации проверки фактов для исключения фейковых новостей. Наконец, отсутствие персонализации приводит к тому, что пользователи сталкиваются с нерелевантным контентом, не соответствующим их интересам [5].

Цели автоматизации:

- автоматизированный сбор новостей из множества источников;
- фильтрация данных, удаление дубликатов и категоризация;
- выбор предпочтений и интересов пользователей, сохранение истории взаимодействия;

– реализация методов анализа текста для определения ключевых тем и актуальности информации;

– проверка достоверности информации посредством сравнения с проверенными источниками;

– сокращение времени, необходимое для поиска информации [6].

Автоматизированная система обладает высокой масштабируемостью, что позволяет ей адаптироваться к изменениям медиарынка и эффективно реагировать на новые вызовы.

Выделим основные этапы автоматизации:

– анализ предметной области и существующих решений. Изучение источников данных и методов их интеграции. Выявление проблем и недостатков текущих подходов;

– разработка архитектуры системы. Определение модулей системы, включая сбор, обработку и предоставление данных;

– реализация ключевых алгоритмов. Автоматический сбор новостей. Очистка и категоризация данных. Внедрение функций персонализации;

– тестирование системы. Проверка на корректность, надежность и производительность;

– внедрение и сопровождение. Настройка системы для конечных пользователей и обучение работе с ней.

Система агрегации новостей обрабатывает три основных типа входных данных:

– основной контент включает полнотекстовые новостные материалы с сохранением оригинального форматирования, заголовки статей с возможностью автоматической нормализации, а также краткие аннотации;

– метаинформация содержит точные временные метки публикации с указанием временных зон, канонические URL-адреса оригинальных публикаций, идентификаторы и наименования источников с поддержкой многоязычных вариантов;

– классификационные данные состоят из тематической рубрики по внутреннему классификатору системы, географических привязок с указанием стран, регионов и городов, а также специальных флагов, обозначающих важность и срочность материала. Все эти данные поступают в систему в структурированном виде и проходят многоэтапную обработку перед публикацией для конечных пользователей.

Приведем пример входного документа JSON от сервера:

```
{
  "article_id": "bbc_20250615_12345",
  "header": {
    "title": "Новые санкции ЕС",
    "subtitle": "Расширение списка ограничений"
  },
  "content": {
    "text": "Полный текст новости...",
    "formatted_html": "<p>Полный текст...</p>"
  },
  "metadata": {
    "published": "2025-06-15T08:45:00+01:00",
    "source": {
      "id": "bbc",
      "name": "BBC News"
    },
    "authors": [
      { "name": "John Doe", "role": "correspondent" }
    ],
    "classification": {
      "category": "politics",
      "geo_tags": ["EU", "UK"],
      "priority": "high"
    }
  }
}
```

Процесс обработки и нормализации входящих документов представляет собой многоступенчатую процедуру, включающую парсинг и валидацию структуры данных, извлечение и преобразование ключевой информации, унификацию форматов

дат и времени, дополнение метаданными и создание поисковых индексов для обеспечения быстрого доступа к информации. Система автоматически выявляет и корректирует различные аномалии в данных, такие как неполные или поврежденные записи,

несоответствия форматов, дублирующий контент и устаревшие материалы, что гарантирует высокое качество и актуальность хранимой информации.

Все поступающие документы подвергаются комплексному хранению с сохранением исходных данных в оригинальном формате (сырые данные), обработанной и нормализованной версии контента, а также специализированных поисковых индексов, оптимизированных для быстрого поиска и анализа. Такой трехуровневый подход к хранению обеспечивает как сохранность первоначальной информации, так и эффективность работы с актуальными нормализованными данными, поддерживая при этом высокую производительность поисковых операций. Архивация осуществляется с учетом временных меток и частоты обращения к данным, что позволяет оптимально управлять ресурсами хранения.

Разработанная система агрегации новостных статей реализована с использованием современных технологий и инструментов, обеспечивающих высокую производительность, кроссплатформенность и удобство разработки. Выбор программного обеспечения осуществлялся на основе анализа требований к функциональности, производительности и совместимости [7].

Основой клиентской части приложения стал React Native – популярный фреймворк для создания кроссплатформенных мобильных приложений. Его использование позволило разработать единую кодовую базу для платформ iOS и Android, что значительно сократило временные затраты на разработку. React Native обеспечивает нативную производительность благодаря оригинальным компонентам платформ, сохраняя при этом все преимущества React, включая виртуальный DOM и компонентный подход [8].

Для ускорения процесса разработки и тестирования был задействован Expo – комплекс инструментов и сервисов, работающий поверх React Native. Expo предоставляет готовые решения для типовых задач мобильной разработки, включая доступ к API устройства, систему сборки и удобные механизмы отладки. Особую ценность представляет Expo Go – мобильное приложение, позволяющее тестировать разрабатываемое решение на физических устройствах без необходимости сложной настройки среды разработки [9, 10].

Серверная часть системы построена на взаимодействии с NewsAPI – специализированным веб-сервисом для агрегации новостей. Выбор данного API обусловлен его надежностью, простотой интеграции

и широким охватом источников. Сборка финального приложения осуществлялась с использованием EAS Build – облачного сервиса от Expo для компиляции готовых билдов. Данное решение необходимо для настройки локальной среды различных платформ.

В связи с этим разработанная многофункциональная платформа агрегации новостей имеет технические требования, что обеспечивает бесперебойную работу приложения на всех стадиях его жизненного цикла. Комплекс технических решений был подобран с учетом особенностей кроссплатформенной разработки и специфики работы с новостными API.

Особое внимание уделено адаптивности системы к различным техническим условиям:

- оптимизация работы при нестабильном интернет-соединении;
- поддержка различных разрешений экранов и плотностей пикселей;
- корректная работа на устройствах с различными техническими характеристиками.

Выбранная архитектура технического обеспечения обеспечивает стабильную работу приложения на 98% современных мобильных устройств, быструю адаптацию к изменениям в API новостных источников, простоту масштабирования системы при росте числа пользователей, возможность легкой интеграции новых функций и модулей.

Технические решения были подобраны с учетом баланса между производительностью, стоимостью обслуживания и простотой развертывания, что делает систему экономически эффективной и удобной в эксплуатации.

Разработанная система реализует комплекс мер по обеспечению информационной безопасности, соответствующий современным требованиям к защите данных. Безопасность приложения обеспечивается на нескольких уровнях. На уровне передачи данных применяется защищенное HTTPS-соединение для всех запросов к NewsAPI. Это гарантирует шифрование передаваемой информации и предотвращает возможность перехвата данных третьими лицами. Для аутентификации при работе с API используется механизм API-ключей, которые хранятся в защищенной конфигурации среды Expo, исключая их попадание в публичный доступ [11, 12].

В клиентской части приложения реализованы механизмы валидации входящих данных от API, что предотвращает потенциальные атаки через внедрение вредоносного кода. Все получаемые новостные материалы проходят санитарную обработку перед отображением, исключая возможность XSS-атак. Для обработки поль-

звательского ввода строгие регулярные выражения. Особое внимание уделено защите персональных данных в соответствии с требованиями GDPR и ФЗ-152 [13, 14]. Комплексный подход к информационной безопасности позволяет гарантировать надежную защиту системы и соответствие современным стандартам безопасности мобильных приложений [15].

Заключение

Разработка многофункциональной платформы для агрегирования новостных статей играет ключевую роль в повышении продуктивности информационных агентств и улучшении удобства для пользователей. Применение передовых технологий обеспечивает оперативный доступ к актуальным материалам, стабильное обновление контента и удобные инструменты для анализа данных. Внедрение данной системы способствует оптимизации процессов получения, обработки и распространения информации, что способствует значительному улучшению качества медиамониторинга и скорости предоставления данных.

Список литературы

1. Абилюдинова Г.М., Айтимова А.М. Математическая модель оптимизации работ при разработке автоматизированных информационных систем // Цифровые модели и решения. 2024. Т. 3. № 4. С. 91–100. DOI: 10.29141/2949-477X-2024-3-4-6. EDN: JXILNM.
2. Лешук Н.С. Сайты-агрегаторы как средство информационной адаптации в глобальной сети интернет // Вестник науки и образования. 2021. № 11–1 (114). С. 29–32. EDN: QYRISV.
3. Сундатов В.О., Тупиков В.О. Проект сайта-агрегатора с автоматизированным наполнением контентом при помощи веб-парсинга // Интернаука. 2025. № 6–1 (370). С. 44–45. EDN: XNTDQM.
4. Абдрафикова С.Р. Характеристика сайта-агрегатора для бронирования жилья «Яндекс.Путешествия» // Ответственные и экономические науки. Студенческий научный форум: сборник статей по материалам LXXI студенческой международной научно-практической конференции (Москва, 05 февраля 2024 г.). М.: ООО «Международный центр науки и образования», 2024. С. 21–23. EDN: PYWCBU.
5. Сайдалиев Э.С., Тимошук О.А. Возможности сайта-агрегатора // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2020. № 4 (30). С. 102–108. EDN: JWMVDB.
6. Кольева Н.С., Шемакин В.В. Примеры использования блокчейн-технологии // Математическое и компьютерное моделирование: сборник материалов XII международной научной конференции (Омск, 14 марта 2025 г.). Омск: Омский государственный университет им. Ф.М. Достоевского, 2025. С. 211–212. EDN: SKIFOR.
7. Сундатов В.О., Тупиков И.В. Разработка сайта агрегатора новостей. Использование ботов для информационной рассылки // Студенческий вестник. 2024. № 21–11 (307). С. 22–23. EDN: PDMINU.
8. Гусейнова А.Р., Магеррамов З.Т. Алгоритм работы онлайн-платформы для автоматизации поиска вакансий // Sciences of Europe. 2023. № 114 (114). С. 110–113. DOI 10.5281/zenodo.7811577. EDN: PMRJXE.
9. Лаврова Г.Н., Чубрина К.А. Развитие модели агрегации информационных сервисов электронной коммерции // Экономика и управление: проблемы, решения. 2018. Т. 6. № 12. С. 95–99. EDN: PNJOGR.
10. Рванова А.С., Кольева Н.С., Панова М.В. Разработка модели оценки структурной сложности программ // Цифровые модели и решения. 2024. Т. 3. № 2. С. 5–16. DOI: 10.29141/2949-477X-2024-3-2-1. EDN: UEZYBI.
11. Беликов А.Г. Разработка системы агрегации новостей сообществ социальных сетей // Постулат. 2021. № 1 (63). EDN: KQDFKV.
12. Черняк А.Г., Корелина Т.В., Зеленина А.Н. Создание агрегатора интернет-сервис-провайдеров с подсистемой автоматизированной обработки адресных баз // Вестник Воронежского института высоких технологий. 2022. № 2 (41). С. 123–129. EDN: AIXZRI.
13. Кортенко Л.В., Рудник В.А., Хасимьянова И.В. Безопасность персональных данных сотрудников при анкетировании по методологии «Оценка 360°» в ИС: Зарплата и управление персоналом ПРОФ // Современные проблемы обеспечения безопасности: сборник материалов XXVI Международной научно-практической конференции (Екатеринбург, 24–25 апреля 2024 г.). Екатеринбург: Уральский институт Государственной противопожарной службы МЧС России, 2024. С. 182–188. EDN: DDVKPZ.
14. Елизарова М.Ю. Правовой режим персональных данных лиц, участвующих в гражданском судопроизводстве // Известия Юго-Западного государственного университета. Серия: История и право. 2024. Т. 14. № 3. С. 108–120. DOI: 10.21869/2223-1501-2024-14-3-108-120. EDN: GYGFD.
15. Филиппов Д.С. Информационный сервис для сбора и анализа данных из Telegram в целях содействия принятию управленческих решений в экономической сфере на отраслевом и региональном уровне // ФЭС: Финансы. Экономика. Стратегия. 2023. Т. 20, № 4. С. 37–42. EDN: VXZPY.

НАУЧНЫЙ ОБЗОР

УДК 664:637.07

**ПРИМЕНЕНИЕ КОЖУРЫ ГРАНАТА
В ПИЩЕВОЙ ПРОМЫШЛЕННОСТИ****Хамзина Е.И.***ФГБОУ ВО «Уральский государственный экономический университет»,
Екатеринбург, e-mail: xei260296@mail.ru*

Гранат (*Punica granatum L.*) с древних времен известен своими полезными для здоровья свойствами, причем не только сок, но и кожура плода обладает ценными биологическими активностями. Целью является обзор последних исследований, посвященных применению кожуры граната в пищевой промышленности, с акцентом на ее антиоксидантные, антимикробные и противовоспалительные свойства. В качестве материалов были рассмотрены научные статьи российских и зарубежных ученых за последние 10-12 лет. В статьях описаны экстракты и порошки, полученные из кожуры граната, а также их применение в различных пищевых продуктах, таких как молочные продукты, мясо, выпечка и упаковочные материалы. Согласно литературным данным добавление кожуры граната улучшает антиоксидантную активность, увеличивает срок годности продуктов, снижает окислительные процессы и подавляет рост патогенных микроорганизмов. Например, экстракт кожуры граната продемонстрировал эффективность в замедлении микробной порчи мяса и рыбы, а также в сохранении свежести фруктов при использовании в съедобных покрытиях. Кроме того, порошок кожуры граната повысил пищевую ценность выпечки за счет увеличения содержания клетчатки и минералов. Важным аспектом является также разработка биоразлагаемых упаковочных материалов с добавлением экстракта кожуры граната, которые обладают антимикробными свойствами и способствуют устойчивому развитию. В заключение отмечается, что кожура граната представляет собой ценный природный ресурс для пищевой промышленности, позволяющий не только улучшать качество и безопасность продуктов, но и сокращать отходы, что соответствует современным тенденциям устойчивого производства.

Ключевые слова: экстракт, кожура, антиоксидант, полифенолы, молочные продукты, мясо**APPLICATION OF POMEGRANATE PEEL
IN THE FOOD INDUSTRY****Khamzina E.I.***Ural State University of Economics, Ekaterinburg, e-mail: xei260296@mail.ru*

Pomegranate (*Punica granatum L.*) has been known for its health benefits since ancient times, and not only the juice but also the peel of the fruit possesses valuable biological activities. The aim is to review the latest research on the application of pomegranate peel in the food industry, focusing on its antioxidant, antimicrobial and anti-inflammatory properties. Research articles by Russian and international scientists published over the past 10-12 years were reviewed. The articles describe extracts and powders obtained from pomegranate peel, as well as their use in various food products, such as dairy products, meat, baked goods, and packaging materials. According to the literature, the addition of pomegranate peel improves antioxidant activity, extends the shelf life of products, reduces oxidative processes, and inhibits the growth of pathogenic microorganisms. For example, pomegranate peel extract demonstrated efficacy in retarding microbial spoilage of meat and fish, and in preserving fruit freshness when used in edible coatings. In addition, pomegranate peel powder improved the nutritional value of baked goods by increasing the fiber and mineral content. An important aspect is also the development of biodegradable packaging materials with the addition of pomegranate peel extract, which have antimicrobial properties and contribute to sustainable development. In conclusion, it is noted that pomegranate peel is a valuable natural resource for the food industry, allowing not only to improve the quality and safety of products, but also to reduce waste, which is in line with modern trends in sustainable production.

Keywords: extract, peel, antioxidant, polyphenols, dairy, meat**Введение**

Гранат (*Punica granatum L.*) с древних времен является сельскохозяйственной культурой. Его название происходит от латинского *Malum granatum*, что означает «зернистое яблоко» [1]. Гранатовое дерево имеет искривленный и колючий ствол, удлиненные зеленые с гладкой поверхностью листья и цветки оранжевой или малиновой окраски, плод граната имеет шаровидную форму диаметром 6-12 см. Окраска кожуры граната варьируется от желтой, зеленой и розовой, которая, в свою очередь,

может переходить в интенсивно-красную и темно-фиолетовую. Гранат очень хорошо адаптируется к климатическим условиям и распространен в регионах с тропическим и субтропическим климатом. В настоящее время гранат растет на территории Ирана, по всему Средиземноморью, в Турции, Индии, Мексике, США, Китае. В народной медицине гранат применяется при заболеваниях желудочно-кишечного тракта, печени, почек, сердечно-сосудистой системы, для поддержания иммунной системы, при хронических воспалительных про-

цессах (ревматоидный артрит). Лечебные свойства плодов граната объясняются содержанием различных полифенолов, среди которых наиболее распространены гидролизуемые танины (галлотаннины и эллагитаннины), гидроксикоричные кислоты, флаванолы (эпигаллокатехин), флаваноны (нарингенин и гесперидин), флавонолы (мирицетин, кверцетин и кемпферол), флавоны (апигенин и лютеолин), антоцианы (дельфинидин, цианидин и пеларгонидин) и дигидрохалконы (флоридзин) [2].

Целью настоящего исследования является анализ последних научных данных о применении экстрактов кожуры граната в качестве функциональных ингредиентов в пищевой промышленности. В обзоре рассматриваются, главным образом, их функциональные свойства: антиоксидантные, антимикробные и противовоспалительные.

Материалы и методы исследования

В исследовании проведен поиск научных статей в электронных базах данных Google Scholar, PubMed, Science Direct, Scopus, eLibrary.ru, в которых рассматриваются тенденции применения кожуры граната в пищевой промышленности. Поиск проводили по ключевым словам: «кожура граната», «функциональные продукты питания», «пищевая упаковка». По результатам поиска было получено 102 статьи. В итоговую выборку были отобраны статьи российских и зарубежных авторов, написанные за последние 10-12 лет (2013-2025 гг.).

Результаты исследования и их обсуждение

Съедобная часть граната составляет только 50% от общего веса, а оставшиеся 50% приходятся на кожуру, которую обычно выбрасывают как отходы. Растущий интерес к устойчивым методам производства продуктов питания в последнее время привел к тому, что все больше внимания уделяется использованию сельскохозяйственных отходов в пищевой промышленности. Одним из видов сельскохозяйственных отходов, получивших значительное внимание, является кожура граната. Она содержит разнообразный набор фенольных соединений (пуникалагин, галловая кислота, кофейная кислота и эллаговая кислота), флавоноидов (катехин, эпикатехин и галлокатехин) и антоцианов [3, 4]. Пуникалагин является одним из основных соединений, содержащихся в кожуре граната, характеризуется высокой молекулярной массой и считается отличительной особенностью этого фрукта. Пуникалагин обладает различными потенциальными терапевтическими преимуще-

ствами, включая антипролиферативные, противовоспалительные, гепатопротекторные и антигенотоксические свойства. Эллаговая кислота обладает различными свойствами, включая антиоксидантное, антигепатотоксическое, антихолестатическое, антифиброгенное, антигепатоканцерогенное и противовирусное действие. Свойства вышеупомянутых веществ способствуют улучшению структуры и функции печени, повышая устойчивость к различным вредным воздействиям [5]. Таким образом, эти соединения, отвечающие за мощные антиоксидантные, противовоспалительные и антимикробные свойства кожуры, делают ее ценным ресурсом в пищевой промышленности. Кроме того, кожура содержит значительные количества сырого белка включая лизин, лейцин, ароматические аминокислоты (фенилаланин и тирозин), треонин, валин и аминокислоты, содержащие серу (метионин и цистеин) и изолейцин. Основными минералами в кожуре граната являются К, Са, Р и Na, также сообщается о значительном содержании Fe, Zn, Cu и Se. Кожура граната также содержит высокие уровни витамина В1, витамина В2, витамина С, витамина Е, а также витамина А [6]. Кожура граната является хорошим источником пищевых волокон, которые необходимы для поддержания здоровья пищеварительной системы. Клетчатка способствует регулярному стулу и может способствовать общему здоровью желудочно-кишечного тракта с контролируемым уровнем сахара в крови. Содержание клетчатки составляет от 33% до 62%, поэтому ее можно использовать в качестве натуральной альтернативы нейтральным сахарам, и она может снизить всасывание холестерина. Кожура граната и её экстракт, помимо своей питательной ценности, выполняют важные технические функции, такие как антиоксидантное, антибактериальное действие, натуральный краситель и усилитель вкуса. Именно поэтому, кожуру граната можно добавлять в функциональные продукты питания, повышая их пищевую ценность и полезные для здоровья свойства и сохраняя или улучшая качества продуктов питания.

Т. Ismail с соавторами обнаружили, что кожура граната может использоваться для увеличения содержания клетчатки в печени и связанных с ним продуктах питания [7]. Жмых кожуры граната содержит относительно меньше белка, но богат углеводами, поэтому его можно использовать в качестве добавки в более высоких концентрациях в безбелковых калорийных продуктах питания. Печень, обогащенная 7,5% кожуры граната, показало снижение кало-

рийности и заметное увеличение ($p < 0,05$) пищевых волокон и неорганических компонентов в продукте. Кроме того, была отмечена положительная связь между содержанием фенолов в печени и результатами тестов на антиоксиданты, а именно анализом на 2,2-дифенил-1-пикрилгидразил (DPPH) и анализом антиоксидантной активности железа. Печень с добавлением кожуры граната, обладающее значительными антиоксидантными свойствами, снижало количество тиобарбитуровой кислоты на 67% и снижало рост количества аэробных микроорганизмов ($2,04-1,30 \log_{10} \text{КОЕ/г}$) и дрожжей/плесени ($1,70-1,05 \log_{10} \text{КОЕ/г}$). В другом исследовании при добавлении порошка кожуры граната в кексы наблюдалось значительное улучшение общего фенольного состава кексов, увеличения содержания минералов (Ca, Mg и K), общих антиоксидантных свойств и увеличение содержания пищевых волокон [8]. Кроме того, добавление экстракта кожуры граната в пшеничную муку привело к увеличению пищевой ценности маффинов за счет значительного увеличения содержания клетчатки (с 4,39 до 10,66%), антиоксидантной активности (с 75,94% до 99,36%), кальция (с 200,33 до 294,33 мг 100 г⁻¹), калия (с 227,33 до 425,33 мг 100 г⁻¹) и магния (с 96,33 до 288,33 мг 100 г⁻¹). Образцы маффинов были органолептически приемлемы при содержании экстракта кожуры граната до 8%. Содержание свободных жирных кислот, перекисное число и количество микроорганизмов в маффине с 8% экстракта кожуры граната были значительно ниже, чем в контрольном образце, и обладали большей окислительной и микробной стабильностью в течение срока хранения 21 и 28 дней при температуре окружающей среды и в холодильнике соответственно [9]. Экстракт кожуры граната добавляют в молочные продукты для повышения их антиоксидантной активности и увеличения сроков хранения [10]. Было обнаружено, что добавление кожуры граната в мороженое в концентрации 0,5 и 1,0% по массе усиливает антиоксидантные свойства (нейтрализация свободных радикалов) и ингибирует α -глюкозидазу [11]. Примечательно, что в мороженом не было обнаружено вяжущего вкуса, что свидетельствует о потенциальной приемлемости для потребителей.

Добавляя экстракт или порошок из кожуры граната в мягкий сыр в качестве натурального консерванта, производители повысили микробиологическую безопасность продукта, не прибегая к синтетическим добавкам. Кроме того, антиоксидантные свойства кожуры граната предотвращают порчу

сыра, увеличивая срок годности, и сохраняя при этом вкус и текстуру сыра [12]. Также использование кожуры граната при производстве сыра приводит к улучшению окислительной стабильности липидов [13]. Однако добавление этого экстракта может привести к ухудшению сенсорных характеристик сыров в основном из-за терпкости и горечи [14].

Кожа граната была исследована в качестве замены синтетических консервантов в мясных продуктах для продления срока годности при сохранении качественных характеристик мяса. Включение экстрактов кожуры граната может предложить различные преимущества, такие как улучшенная текстура, более длительный срок годности, улучшенная стабильность цвета и повышенная антиоксидантная активность. Например, добавление экстракта кожуры граната в сырой свиной фарш позволило снизить перекисное число, количество бактерий в хранившемся при температуре $4 \pm 1^\circ\text{C}$ в течение 12 дней фарше по сравнению с контрольной группой [15]. Перекисное окисление липидов ингибировалось фенольными соединениями из кожуры граната, которые блокировали реакции свободнорадикального окисления. Включая экстракты кожуры граната в мясные продукты, производители могут предложить потребителям более здоровую и альтернативу, удовлетворяя их растущий спрос на натуральные и функциональные продукты.

Экстракт кожуры граната обладает антимикробными свойствами, демонстрируя ингибирующий эффект против различных бактерий и грибов из-за наличия большого количества танинов. В исследовании J. Chen и соавторов сообщалось, что кожа граната оказывает антибактериальное действие против пищевых патогенов, таких как *Bacillus subtilis*, *Escherichia coli*, *Penicillium italicum* и *Fusarium sambucinum* [16]. С использованием экстракта кожуры граната были изготовлены антибактериальные пленки эффективные против роста бактерий *Salmonella* и *Staphylococcus aureus* [17]. Экстракты кожуры граната потенциально могут использоваться в качестве антибактериальных агентов в рыбных продуктах. Например, когда креветки замачивали в метанольных экстрактах кожуры граната в течение 15 минут рост микробов был замедлен на срок до шести дней хранения в холодильнике [18]. В другом исследовании рыбные котлеты, приготовленные с натуральным экстрактом кожуры граната, имели более длительный срок хранения до 11 дней в условиях розничной торговли [19]. А также гранат оказывал сильное ингибирующее действие на *Listeria monocytogenes*.

Упаковка для пищевых продуктов имеет огромное значение для пищевой промышленности. Ее основная роль заключается в защите пищевых продуктов от внешних загрязнителей и факторов окружающей среды, тем самым обеспечивая их безопасность, качество и долговечность. Однако использование традиционных упаковочных пленок (полиэтилен) создает серьезные экологические проблемы. Они в основном производятся из нефти, не поддаются биологическому разложению и сохраняются в окружающей среде сотни лет, способствуя загрязнению и деградации экосистем. Более того, серьезную озабоченность вызывает образующийся в процессе эксплуатации микропластик. Несмотря на удобство и эффективность, воздействие на окружающую среду традиционных упаковочных пленок, таких как полиэтилен, подчеркивает острую необходимость в устойчивых упаковочных решениях, направленных на сокращение отходов и минимизацию воздействия на окружающую среду на протяжении всего жизненного цикла упаковки [20]. Остро стоит вопрос создания упаковок с минимальным воздействием на окружающую среду и не только защитные, барьерные функции, но и улучшить сенсорные свойства пищевых продуктов, обеспечить безопасность пищевых продуктов и сохранить качество пищевых продуктов. Антимикробные и антиоксидантные соединения, могут быть добавлены в упаковочный материал или на его поверхность для продления срока годности упакованных пищевых продуктов [21].

В настоящее время наблюдается рост интереса к разработке новых упаковочных материалов для пищевых продуктов и, особенно, съедобных упаковочных материалов. В целом, съедобная упаковка является биоразлагаемой, нетоксичной по своей природе и может использоваться как напыление, так и в виде пленки в зависимости от природы пищевых продуктов. Съедобная упаковка сохраняет или, в некоторых случаях, даже улучшает пищевой состав, биологические и сенсорные характеристики продукта за счет снижения потерь влаги, улучшения внешнего вида или минимизации биохимических изменений во время хранения. В след за научными исследованиями использование съедобной упаковки, содержащей экстракты отходов переработки фруктов с высоким содержанием биологически активных соединений, становится все более популярной и в пищевой промышленности. Их включают в съедобные упаковочные материалы для улучшения антиоксидантных, антимикробных и сенсорных характеристик продукта, а также для изменения структуры

покрытия, что улучшает их функциональность для пищевых применений. Гранатовая кожура содержит пектин (20–25%), целлюлозу (5–17%), поэтому ее активно используются для разработки активных биоразлагаемых пленок для пищевых продуктов, которые являются нетоксичными, экологически чистыми и съедобными из-за повышенного содержания фенолов и антиоксидантных и противомикробных свойств. Например, в работе А.Н. Nabeel и соавторов описано получение съедобных пленок на основе карбоксиметилцеллюлозы/желатина с добавлением экстракта кожуры граната для продления сроков хранения говядины [22]. Полученные пленки демонстрируют улучшенные функциональные свойства, включая улучшенную механическую прочность, светонепроницаемость, более низкую проницаемость для водяного пара и контролируемое высвобождение биоактивных соединений. Полученные «гранатовые» композитные пленки с добавлением демонстрируют эффективное ингибирование развитие пищевых микробов и замедление процессов липидного окисления в говядине, увеличивая срок хранения говядины до 3 дней. Это подчеркивает их применимость в качестве активных упаковочных материалов для пищевых продуктов вместо обычных пластиков. В другой работе при использовании двухслойных пленок с порошком кожуры граната в качестве активной упаковки для мяса страуса при хранении в холодильнике было обнаружено, что добавление порошка граната, особенно в концентрации 3% в пленку, эффективно подавляет рост микроорганизмов, снижает перекисное окисление липидов и улучшает сенсорные свойства мяса, в конечном итоге продлевая срок годности мяса страуса [23]. Порошок кожуры граната также использовался для приготовления съедобных пленок на основе крахмала [17]. Порошок использовался в качестве антимикробного и укрепляющего агента. Новая пленка продемонстрировала ингибирующее действие как на золотистый стафилококк, так и на сальмонеллу.

Добавление экстрактов кожуры граната (75 мг/мл) в пленку на основе белка – зеина – приводит к увеличению прочности пленки на разрыв, общего содержания фенолов и антиоксидантной активности зеиновых пленок, тогда как растворимость пленки и скорость пропускания водяных паров снижаются. Было показано, что эта зеиново-гранатовая пленка обладает ингибирующей активностью против нескольких патогенных бактерий и использовалась для упаковки сыра, пода-

влияя рост микроорганизмов, вызывающих порчу, но не влияя на молочнокислые бактерии. Кроме того, сыр в зеиново-гранатовой упаковке имеет низкое содержание продуктов окисления белков и липидов при хранении по сравнению с сыром, упакованным в зеиновую пленку без экстракта кожуры граната [24].

Для хранения белого хлеба была изготовлена съедобная упаковочная пленка из кожуры граната и глицерина в качестве пластификатора. Съедобная гранатовая пленка позволила уменьшить потерю веса хлеба при хранении. Это может быть связано со способностью пленки на основе гранатовой кожуры ограничивать потерю влаги из белого хлеба. Микробная обсемененность хлеба, завернутого в съедобную пленку, значительно снизилась по сравнению с хлебом, завернутым в полиэтилен [25].

Свежие фрукты и овощи являются скоропортящимися, и требуют защиты в первую очередь от гниения. Гниение фруктов и овощей после сбора может быть вызвано дремлющими и латентными инфекциями, которые возникают в период между цветением и созреванием плодов, а также повреждениями плодов, возникающими во время сбора, обработки и хранения урожая. Экстракт кожуры граната показал свою эффективность в продлении срока годности и сохранении микробиологических свойств пищевых продуктов при использовании в качестве антимикробного агента. Например, покрытие из эмульсии нанохитозана, настоянного на экстракте кожуры граната, эффективно сохраняет качество и продлевает срок годности плодов абрикоса [26]. В другом исследовании съедобные покрытия из хитозана и камеди рожкового дерева, обогащенные экстрактом кожуры граната, позволили минимизировать послеуборочную гниль апельсинов из-за *Penicillium digitatum* на 28% и 49% соответственно по сравнению с контрольными образцами [27].

Заключение

Последние годы потребность пищевой промышленности в антиоксидантах природного происхождения постоянно растёт, особенно в связи с увеличением числа сообщений о неблагоприятных токсикологических свойствах многих синтетических соединений. Таким образом, кожа граната, обладающая высокими антиоксидантными и антимикробными свойствами, имеет большой потенциал для применения в пищевых продуктах. Многочисленные исследования показали, что эти добавки могут положительно влиять на общие сенсорные качества и, следовательно, на срок

годности пищевых продуктов. Такой подход не только улучшает здоровье потребителей, снижая риск заболеваний пищевого происхождения, но и соответствует растущему спросу на продукты без химических добавок, с чистой этикеткой и безопасные. Более того, повторное использование кожуры фруктов способствует сокращению отходов и устойчивому развитию, превращая побочные продукты в ценные ресурсы для пищевой промышленности.

Список литературы

1. El Barnossi A., Moussaid F., Housseini A.I. Tangerine, banana, and pomegranate peels valorization for a sustainable environment: A review // *Biotechnology Reports*. 2021. Vol. 29. P. e00574. DOI: 10.1016/j.btre.2020.e00574.
2. Buccato D.G., Delgado-Osorio A., De Lellis L. Fr., Morone M.V., Ullah H., Izzo L., Lombardi S., Di Minno A., Riccioni C.V., Moriki D., Rufián-Henares J.Á., Daglia M. Exploring the influence of a pomegranate extract on the functionality of healthy and diseased human gut microbiota: An in vitro study // *Molecules*. 2025. Vol. 30. Is. 7. P. 1634. DOI: 10.3390/molecules30071634.
3. Kaderides K., Kyriakoudi A., Mourtzinos I., Goula A.M. Potential of pomegranate peel extract as a natural additive in foods // *Trends in Food Science and Technology*. 2021. Vol. 115. P. 380-390. DOI: 10.1016/j.tifs.2021.06.050.
4. Aziz M., Tamam A.A., Nagm El-diin M.A., Hamdy A.M. Utilization of pomegranate peels powder as a novel preservative in white soft cheese making // *Assiut Journal of Agricultural Sciences*. 2023. Vol. 54. Is. 1. P. 50-65. DOI: 10.21608/AJAS.2023.169946.1191.
5. Rahul P.B., Ravindra K.T., Kshirod K.D., Maanas Sh. Recent advances in encapsulation of pomegranate peel extract and combination of wall materials: a review of encapsulation technologies, characterization and applications in the food industry // *Sustainable Food Technology*. 2025. Vol. 3. Is. 1. P. 123-144. DOI: 10.1039/D4FB00196F.
6. Rowayshed G., Salama A., Abul-Fadl M., Akila-Hamza S., Emad A. Nutritional and chemical evaluation for pomegranate (*Punica granatum* L.) fruit peel and seeds powders by products // *Middle East Journal of Applied Sciences*. 2013. Vol. 3. Is. 4. P. 169-179. URL: <https://www.curreweb.com/mejas/mejas/2013/169-179.pdf> (дата обращения 27.08.2025).
7. Ismail T., Akhtar S., Riaz M., Hameed A., Afzal Kh., Sheikh A.S. Oxidative and microbial stability of pomegranate peel extracts and bagasse supplemented cookies // *Journal of Food Quality*. 2016. Vol. 39. Is. 6. P. 658-668. DOI: 10.1111/jfq.12231.
8. Topkaya C., Isik F. Effects of pomegranate peel supplementation on chemical, physical, and nutritional properties of muffin cakes // *Journal of Food Processing and Preservation*. 2019. Vol. 43. P. 1-11. DOI: 10.1111/jfpp.13868.
9. Giri N.A., Gaikwad P., Gaikwad N.N., Manjunatha N., Krishnakumar Th., Kad V., Raigond P., Suryavanshi S., Marathe R.A. Development of fiber-enriched muffins using pomegranate peel powder and its effect on physico-chemical properties and shelf life of the muffins // *Journal of the Science of Food and Agriculture*. 2024. Vol. 104. P. 2346-2358. DOI: 10.1002/jsfa.13138.
10. Chan C.L., Gan R.Y., Shah N.P., Corke H. Enhancing antioxidant capacity of *Lactobacillus acidophilus*-fermented milk fortified with pomegranate peel extracts // *Food Bioscience*. 2018. Vol. 26. P. 185-192. DOI: 10.1016/j.fbio.2018.10.016.
11. Cam M., Icyer N.C., Erdogan F. Pomegranate peel phenolics: Microencapsulation, storage stability and potential ingredient for functional food development // *LWT – Food Science and Technology*. 2014. Vol. 55. P. 117-123. DOI: 10.1016/j.lwt.2013.09.011.

12. Aguilar-Méndez M.A., Campos-Arias M.P., Quiroz-Reyes C.N., Ronquillo-de Jesús E., Cruz-Hernández M.A. Fruit peels as sources of bioactive compounds with antioxidant and antimicrobial properties // *Revista de la Facultad de Ciencias Agrarias UNCuyo*. 2020. Vol. 52. № 1. P. 360-371. URL: <https://revistas.uncu.edu.ar/ojs3/index.php/RFCA/article/view/2945> (дата обращения 27.08.2025).
13. Mahajan D., Bhat Z.F., Kumar S. Pomegranate (*Punica granatum*) rind extract as a novel preservative in cheese // *Food Bioscience*. 2015. Vol. 12. P. 47-53. DOI: 10.1016/j.fbio.2015.07.005.
14. Sandhya S., Khamrui K., Prasad W., Kumar M.C.T. Preparation of pomegranate peel extract powder and evaluation of its effect on functional properties and shelf life of curd // *LWT*. 2018. Vol. 92. P. 416-421. DOI: 10.1016/j.lwt.2018.02.057.
15. Qin, Y.Y. Zhang Z.H., Li L., Xiong W., Shi J.Y., Zhao T.-R., Fan J. Antioxidant effect of pomegranate rind powder extract, pomegranate juice, and pomegranate seed powder extract as antioxidants in raw ground pork meat // *Food Science and Biotechnology*. 2013. Vol. 22. P. 1063-1069. DOI: 10.1007/s10068-013-0184-8.
16. Chen J., Liao C., Ouyang X., Kahramanoğlu I., Gan Y., Li M. Antimicrobial activity of pomegranate peel and its applications on food preservation // *Journal of Food Quality*. 2020. Vol. 16. P. 1-8. DOI: 10.1155/2020/8850339.
17. Ali A., Chen Y., Liu H., Yu L., Baloch Z., Khalid S., Zhu J., Chen L. Starch-based antimicrobial films functionalized by pomegranate peel // *International Journal of Biological Macromolecules*. 2019. Vol. 129. P. 1120-1126. DOI: 10.1016/j.ijbiomac.2018.09.068.
18. Basiri S., Shekarforoush S.S., Aminlari M., Akbari S. The effect of pomegranate peel extract (PPE) on the polyphenol oxidase (PPO) and quality of Pacific white shrimp (*Litopenaeus vannamei*) during refrigerated storage // *LWT*. 2015. Vol. 60. P. 1025-1033. DOI: 10.1016/j.lwt.2014.10.043.
19. Martínez L., Castillo J., Ros G., Nieto G. Antioxidant and antimicrobial activity of rosemary, pomegranate and olive extracts in fish patties // *Antioxidants*. 2019. Vol. 8. P. 86. DOI: 10.3390/antiox8040086.
20. Baniasadi H., Fathi Z., Lizundia E., Cruz Cr.D., Abidnejad R., Fazeli M., Tammela P., Kontturi E., Lipponen J., Niskanen J. Development and characterization of pomegranate peel extract-infused carboxymethyl cellulose composite films for functional, sustainable food packaging // *Food Hydrocolloids*. 2025. Vol. 158. P. 110525. DOI: 10.1016/j.foodhyd.2024.110525.
21. Малинин А.В., Потороко И.Ю., Руськина А.А. Разработка ультразвуковой технологии получения биоматериалов-сенсоров для упаковки пищевых продуктов // *Индустрия питания*. 2024. Т. 9. № 4. С. 5-12. DOI: 10.29141/2500-1922-2024-9-4-1.
22. Nabeel A. H., Yong Y., Wang S., Munawar N., Zhu J. Development of novel carboxymethyl cellulose/gelatin-based edible films with pomegranate peel extract as antibacterial/antioxidant agents for beef preservation // *Food Chemistry*. 2024. Vol. 443. P. 138511. DOI: 10.1016/j.foodchem.2024.138511.
23. Ghorbanlou S., Mahmoudi R., Ghajarbeygi P., Shahsavari S., Mehrabi A., Kazemina M. Investigate microbial, chemical properties and sensory evaluation of ostrich meat in bilayer films containing pomegranate peel powder during refrigerated storage, as an active packaging // *Journal of Microbiology, Biotechnology and Food Sciences*. 2023. Vol. 12. P. 1-5. DOI: 10.55251/jmbfs.6191.
24. Mushtaq M., Gani A., Punoo H.A., Masoodi F.A. Use of pomegranate peel extract incorporated zein film with improved properties for prolonged shelf life of fresh Himalayan cheese (Kalari/kradi) // *Innovative Food Science and Emerging Technologies*. 2018. Vol. 48. P. 25-32. DOI: 10.1016/j.ifset.2018.04.020.
25. Venkatesh A., Sutariya H. Studies on formulation and properties of fruit peel waste incorporated edible film and its effect on quality of bread // *Journal of Packaging Technology and Research*. 2019. Vol. 3. P. 99-108. DOI: 10.1007/s41783-019-00062-z.
26. Gull A., Bhat N., Wani S.M., Masoodi F.A., Amin T., Ganai Sh. A. Shelf life extension of apricot fruit by application of nanochitosan emulsion coatings containing pomegranate peel extract // *Food Chemistry*. 2021. Vol. 349. P. 129149. DOI: 10.1016/j.foodchem.2021.129149.
27. Kharchoufi S., Parafati L., Licciardello F., Muratore G., Hamdi M., Cirvilleri G., Restuccia C. Edible coatings incorporating pomegranate peel extract and biocontrol yeast to reduce *Penicillium digitatum* postharvest decay of oranges // *Food Microbiology*. 2018. Vol. 74. P. 107-112. DOI: 10.1016/j.fm.2018.03.011.