

**SchoolZum Информатика — среда разработки для обучения
программированию на Python**

Автор:

Мещеряков Марк Вячеславович

Россия, г. Санкт-Петербург, ГБОУ Школа № 174, ученик 10-И класса.

Наставник:

Поляков Константин Юрьевич

Россия, г. Санкт-Петербург, ГБОУ Школа №174, учитель информатики; СПбГМТУ,
профессор кафедры судовой автоматики и измерений, доктор технических наук.

Введение

Среды разработки. В современном мире навыки программирования становятся неотъемлемой частью базового образования. В большинстве школ России для преподавания программирования в школьном курсе информатики применяется язык Python [1], который выделяется среди языков программирования своей простотой и читаемостью кода, что делает его удобным для обучения начинающих. Python не только широко используется в ведущих компаниях и организациях, таких как Google, Яндекс, CERN и NASA, но также активно применяется в образовательных платформах и инструментах, предназначенных для школьников [2].

Существует множество интегрированных сред разработки (IDE) и образовательных платформ, подходящих для обучения программированию на Python. Среди них можно отметить:

- **IDLE.** Стандартная среда разработки, поставляемая вместе с Python. IDLE предоставляет базовые инструменты для написания и выполнения кода, включая подсветку синтаксиса и интерактивную оболочку [4].
- **Thonny.** Кроссплатформенная среда разработки с открытым исходным кодом, специально созданная для начинающих. Thonny предлагает простой интерфейс, встроенный отладчик и подсветку синтаксиса [3].
- **PyCharm Edu.** Специальная версия популярной IDE PyCharm, ориентированная на образовательные цели. PyCharm Edu предлагает встроенные учебные материалы и задания, что способствует эффективному обучению [5].
- **Visual Studio Code.** Легкий и настраиваемый редактор кода с поддержкой множества языков, включая Python. Благодаря расширениям, VS Code предоставляет мощные инструменты для разработки и отладки кода на разных языках [6].

Большинство из этих сред не адаптированы под нужды школьного образования. Так PyCharm Edu и Visual Studio Code слишком сложны для начинающих, а в IDLE и Thonny, отсутствуют некоторые необходимые возможности. На уроках информатики в школах распространенной практикой является решение задач на знание языка и простые алгоритмы. Ни одна из этих сред не позволяет организовать автоматическую проверку. Эти обстоятельства определяют актуальность создания специализированных сред программирования, которые учитывают образовательные стандарты.

Системы тестирования. Среды разработки программ, используемые при обучении программированию, должны быть эффективными, т. е. экономить время ученика и учителя, развивать навыки решения задач и критическое мышление. С этой точки зрения одним из

ключевых инструментов являются тестирующие системы, которые проверяют правильность выполнения заданий [7]. Рассмотрим существующие подходы и преимущества использования тестирующих систем для проверки кода на языке Python.

Тестирующие системы — это программные решения, которые автоматически проверяют правильность работы кода с помощью набора тестов, сравнивая вывод программы с ожидаемыми результатами. Они широко используются в образовательных платформах, на олимпиадах и конкурсах по программированию и системах автоматизированного тестирования кода. Основная цель таких систем — избавиться от необходимости ручной проверки программ и обеспечить объективность оценивания решений [8].

В России наиболее известны следующие тестирующие системы:

1. **Ejudge** — система для проведения олимпиад по программированию. Она позволяет организовывать соревнования с автоматической проверкой задач и поддержкой большого числа языков программирования. Ejudge предоставляет мощные инструменты для настройки тестов и гибкость в определении критериев оценки [9].
2. **NSUts** — включает в себя средства безопасного запуска пользовательских программ в изолированной среде, обеспечивает контроль над ресурсными ограничениями, такими как процессорное время, память и файловый ввод-вывод, и предоставляет механизмы предотвращения мошенничества [10].
3. **Contester** — система для проведения турниров и индивидуального решения задач по олимпиадному программированию (спортивному программированию), которая обеспечивает возможность проверки решений на большинстве современных языков программирования [11].

Существуют также платформы с задачами по программированию — полноценные веб-решения, которые предоставляют интерфейс для просмотра заданий и отправки решений на проверку. Решения проверяются с помощью внутренней тестирующей системы сайта на заранее подготовленных наборах входных данных. Примеры таких платформ: Codeforces [12], LeetCode [13], Codewars [14], Timus Online Judge [15], Информатикс [16]. Ниже приведена таблица (Таблица 1) сравнения удобства написания программ в предоставляемых платформами средах по некоторым критериям.

Таблица 1. Сравнение встроенных сред разработки на платформах с задачами по программированию

	Наличие поля для кода	Подсветка синтаксиса	Авто-дополнение	Авто-закрытие скобок/кавычек	Автоматические отступы	Отладка кода
Codeforces	✗	-	-	-	-	-
LeetCode	✓	✓	✗	✓	✓	✓
Codewars	✓	✓	✗	✓	✓	✗

Timus Online Judge	✓	×	×	×	×	×
Информатикс	×	-	-	-	-	-
Stepik ¹	✓	✓	×	×	✓	×
SchoolZum Информатика	✓	✓	✓	✓	✓	✓

Из Таблицы 1 можно сделать вывод — на некоторых платформах нет возможности писать код, а там, где все же есть текстовый редактор, его функций недостаточно для комфортной работы с текстом программы. Поэтому пользователи вынуждены использовать сторонние среды разработки, а потом отправлять подготовленный в них файл с исходным кодом на тестирование.

Таким образом, на современном этапе при преподавании программирования в школьном курсе информатики приходится использовать две не связанные между собой системы: среду разработки на локальном компьютере и тестирующую систему, чаще всего развёрнутую в локальной или глобальной сети. Интеграция этих двух систем позволит сделать шаг к более эффективному и современному обучению программированию. С этой целью мной была разработана система *SchoolZum Информатика*, которая включает редактор текста программ и встроенный отладчик, а также обеспечивает возможность запуска программ на локальном компьютере и автоматическую проверку решений с помощью собственной тестирующей системы и сайта informatics.msk.ru.

Цель и задачи. Целью проекта является повышение качества образования за счет ускорения работы учеников и оптимизации работы учителя путем создания и внедрения в образовательный процесс системы *SchoolZum Информатика*. Отсюда можно определить показатели назначения проекта: среднее количество решенных за один урок задач (предполагаемый прирост – 10%) и количество проверок задач учителем (предполагаемое изменение – падение до нуля).

План работы (задачи проекта):

1. Проанализировать существующие решения и сформировать требования к новой образовательной системе.
2. Спроектировать архитектуру системы и выбрать язык программирования и фреймворк для создания программного обеспечения.
3. Разработать дизайн интерфейса будущего ПО.
4. Адаптировать мобильное приложение для реализации входа в аккаунт SchoolZum в ПО.

¹ Stepik — российская образовательная платформа и конструктор бесплатных и платных открытых онлайн-курсов и уроков. Эта платформа участвует в сравнении, так как предоставляет текстовое поле ввода кода в заданиях по программированию.

5. Создать собственный API сервер для связи с Информатиксом, предварительно изучив API и HTML-разметку платформы.
6. Создать разделы входа в аккаунт, карты уроков, настроек и разработки в ПО.
7. Создать раздел «Информатика» в меню администрации системы для создания возможности удобного добавления и редактирования уроков и задач, а также раздел «Журнал тестов кода» для просмотра отчета о проверке задач для учителей.
8. Создать собственную тестирующую систему на сервере.
9. Протестировать ПО SchoolZum Информатика на уроках информатики 9А класса в течении всей третьей четверти.
10. Исправить ошибки и недочеты, найденные во время тестирования. Также внести изменения в ПО, ориентируясь на отзывы тестирующей группы.

В Таблице 2 приставлены ресурсы, необходимые для реализации проекта.

Таблица 2. Ресурсы

Ресурс	Привлечение ресурса
компьютерная техника	личная, школьная
техническая документация и обучающие материалы по языкам программирования и фреймворкам	Интернет
русские и иностранные методические материалы	научные интернет-архивы, научные публикации
система контроля версий git и облачное хранилище GitHub	Интернет
техническая документация и открытый исходный код популярных сред разработки	Интернет

Система SchoolZum Информатика

SchoolZum — это школьная информационная экосистема, которая объединяет учеников и учителей на единой веб-платформе (разрабатывается с сентября 2022 года) [17]. Среда разработки *SchoolZum Информатика* является одним из элементов экосистемы SchoolZum и представляет собой программу для настольного компьютера.

Архитектура и технологии. Программа разрабатывалась с помощью фреймворка ElectronJS, который позволяет создавать кроссплатформенные приложения, используя платформу NodeJS (язык JavaScript) для системной части и веб-инструменты, такие как HTML и CSS, для создания интерфейса [18].

Веб-сервер, отвечающий за доступ и тестирование заданий, находящихся в системе SchoolZum, написан на NodeJS. Для хранения задач и другой информации используется база данных MongoDB [19].

Для реализации возможности легкого и беспрепятственного взаимодействия настольного приложения с платформой Информатикс² был создан отдельный веб-сервер на Python с использованием фреймворка Flask [20]. Роль этого сервера — быть связующим между *SchoolZum Информатикой* и сайтом Информатикс. По запросу от клиента (рабочего компьютера ученика) сервер запрашивает страницу с заданием или списком уроков у платформы Информатикс, анализирует полученный HTML-код и возвращает данные клиенту.

Оба веб-сервера находятся на одном виртуальном приватном сервере (VPS), к которому привязан домен schoolzum.ru, поэтому для маршрутизации запросов используется Nginx [21].

На рис. 1 показаны взаимосвязи ключевых элементов проекта.

² «Информатикс» — это образовательная онлайн-платформа, разработанная Московским центром непрерывного математического образования (МЦНМО) и Московским институтом открытого образования (МИОО). Она предназначена для дистанционного обучения информатике и программированию, а также для подготовки к олимпиадам по этим дисциплинам [16].



Рис. 1. Архитектура системы SchoolZum Информатика

Авторизация. Для работы в системе необходима авторизация (рис. 2), которая может быть выполнена одним из следующих способов:

- Аккаунт SchoolZum. При выборе этого варианта ученик получит доступ к базе заданий, составленной учителем в экосистеме SchoolZum. Для этого требуется
 - просканировать QR-код с помощью SchoolZum Mobile³, или
 - просканировать QR-код через другой сканнер и подтвердить авторизацию на сайте, или
 - ввести логин и пароль от своего аккаунта SchoolZum.
- Учетная запись Информатикс (используя логин и пароль). В этом случае ученику откроются задачи из авторского курса «Задачи из учебника К.Ю. Полякова и Е.А. Еремина» [22].

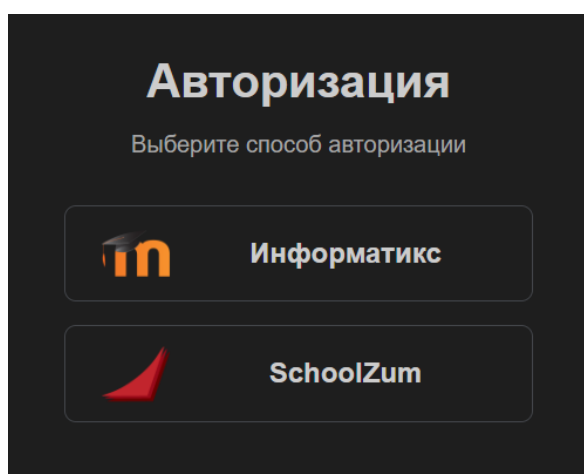


Рис. 2. Экран выбора способа авторизации

³ SchoolZum Mobile — мобильное приложение для Android, с помощью которого удобно взаимодействовать с экосистемой SchoolZum.

Уроки и задачи. После авторизации ученик попадает на карту уроков (см. рис. 3). На ней можно увидеть список всех уроков, каждый из которых содержит несколько задач.

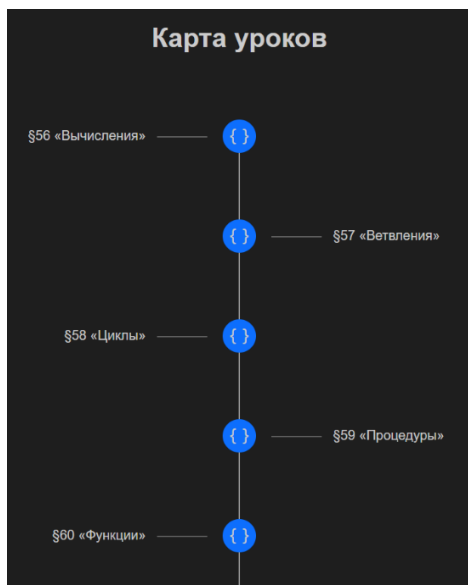


Рис. 3. Карта с уроками с платформы Информатикс

Перечень уроков и связанные с ними задачи можно редактировать с помощью панели администратора на сайте SchoolZum (рис. 4).

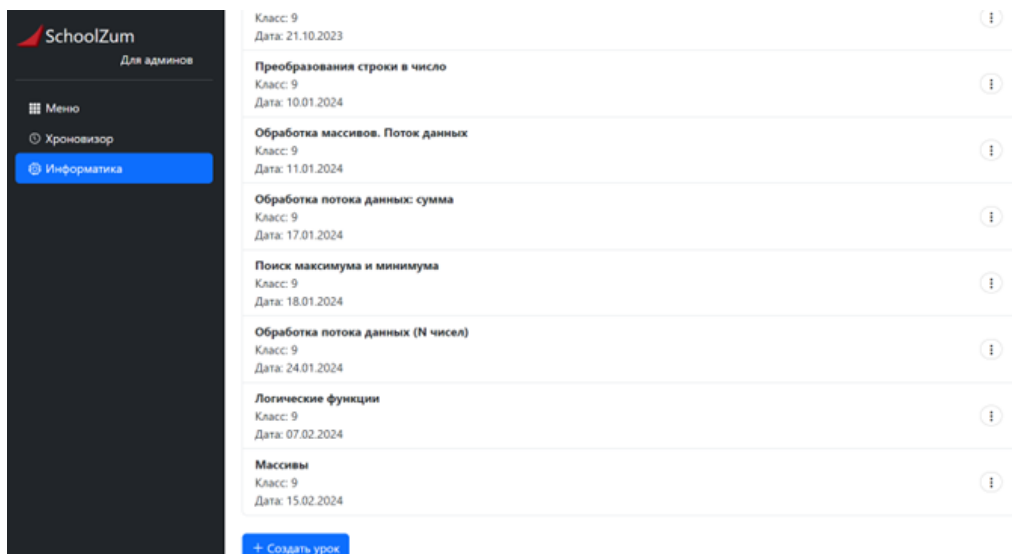


Рис. 4. Панель администратора на сайте SchoolZum

Экран разработки показан на рис. 5. В центре экрана расположено поле для написания кода, которое обладает подсветкой синтаксиса, автоматическими отступами и функцией автодополнения. Справа находится панель с разделом «Задача», где можно увидеть название и текст задания, а также примеры входных и соответствующих выходных данных.

На панели в нижней части экрана отображается процесс выполнения кода на компьютере. В разделе «Проблемы» отображаются все нарушения синтаксиса и стиля Python [23], которые были выявлены в коде.

На верхней панели размещены кнопки для запуска, остановки и отладки кода, а также для его отправки на проверку в тестирующую систему.

Результат проверки кода можно посмотреть в разделе «Решения» (рис. 6). Там отображается время отправки, результат проверки, набранные баллы (в случае частичного решения задачи), а также кнопка просмотра и вставки сохраненного кода решения в рабочее пространство. Таким образом ученик, например, может начать решать задачу в школе, а продолжить — дома.

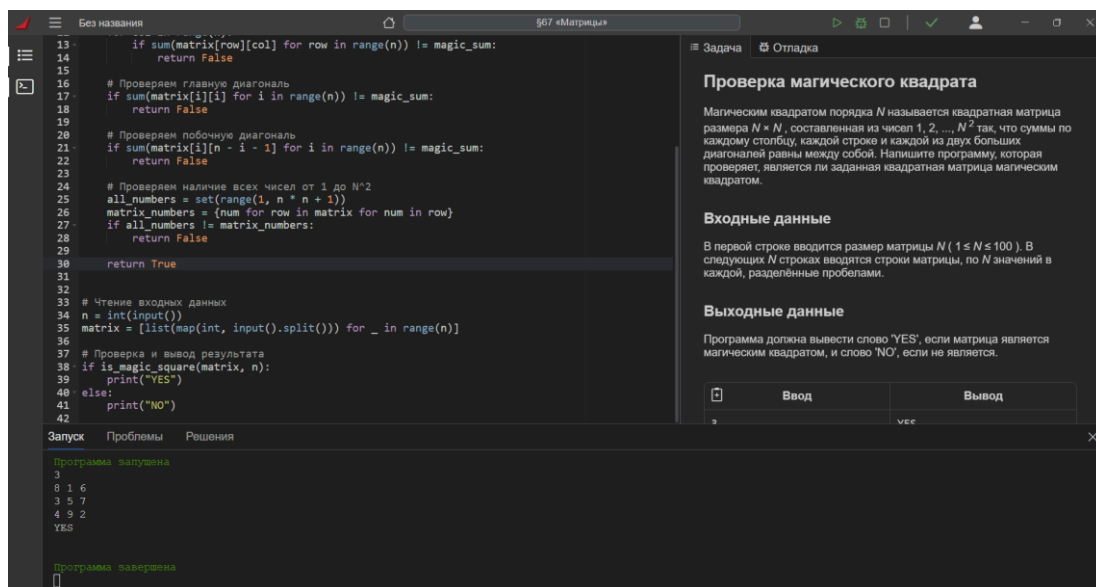


Рис. 5. Экран разработки

Время	Статус	Пройдено тестов	Баллы	Код
08.01.2025, 21:33:29	ОК	11	100	
04.01.2025, 20:15:37	Частичное решение	0	0	

Рис. 6. Раздел "Решения" нижней панели

Отладка. Это важный функционал любой среды разработки, с помощью которого обнаруживают, локализуют и устраняют ошибки. На рис. 7 показано окно программы в процессе отладки. Ученик может построчно выполнять написанную программу с помощью кнопок «Шаг через», «Шаг в» и «Шаг из». Желтым цветом в редакторе выделяется строка, которая выполнится на следующем шаге. Также ученик может наблюдать за изменением всех переменных и введенных им выражений в окнах «Переменные» и «Наблюдение» соответственно.

Система отладки основывается на открытой библиотеке debugpy, разработанной Microsoft, и являющейся основным механизмом отладки в редакторе Visual Studio Code для Python [24]. debugpy выступает сервером отладки, работающим по Debug Adapter Protocol

(DAP) [25], что позволяет SchoolZum Информатике удобно использовать её для локальной отладки.

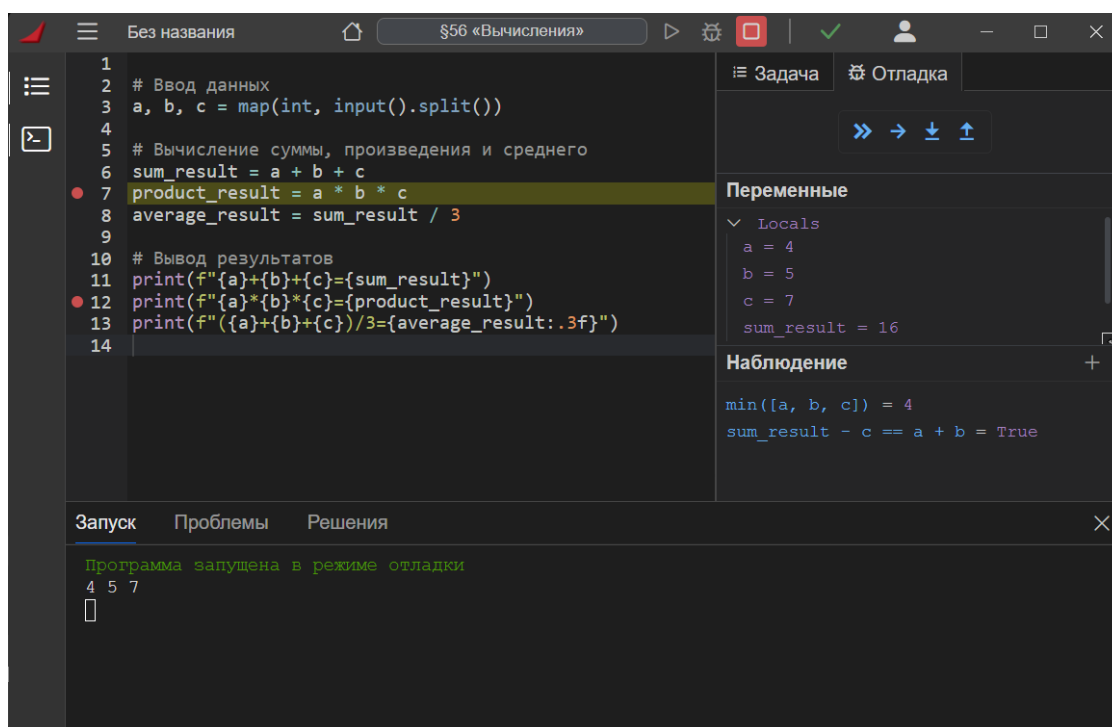


Рис. 7. Запуск кода в режиме отладки

Проверка кода. Задания, добавленные в SchoolZum, проверяются следующим образом: когда код отправляется учеником на проверку на сервер, он выполняется в специальной подготовленной среде («песочнице»), где выполняющаяся программа не может навредить системе. На вход программе подаются входные данные тестов, заранее добавленных в систему, а после завершения работы программы полученные выходные данные сравниваются с ожидаемыми. Все отчеты о проверках учитель может увидеть в специальном разделе на сайте SchoolZum (см. рис. 8).

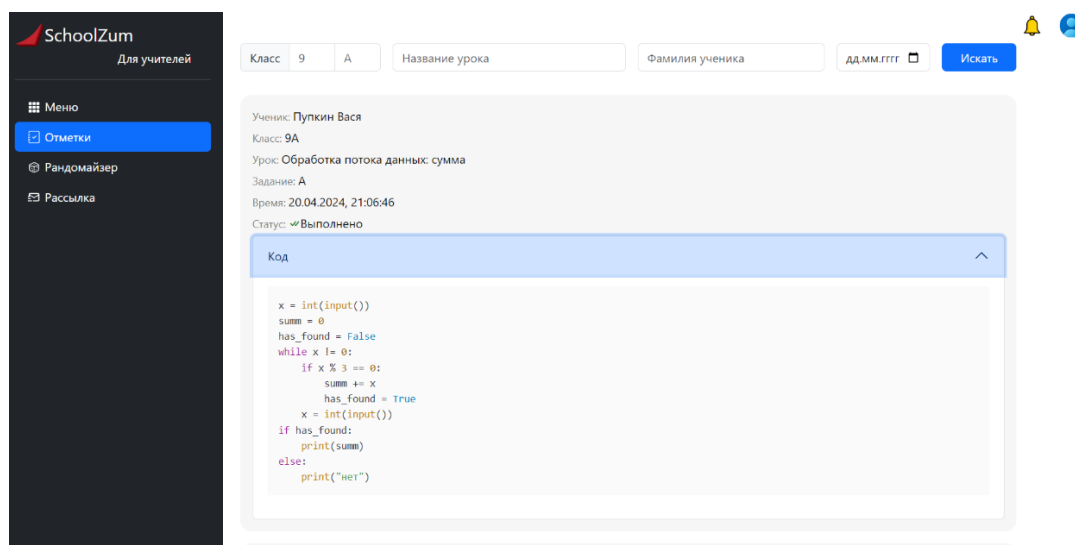


Рис. 8. Отчет о проверке кода

Решения задач из базы Информатикса программа отправляет через сервер SchoolZum на сервера Информатикса, где происходит проверка и сохраняется результат.

Тестирование системы

В школе №174 Санкт-Петербурга с 10.01.2024 до 15.02.2024 в течение 7 уроков (7 академических часов) происходило тестирование. Тестирующая группа состояла из 6 учеников 9А класса, которые на протяжении всего срока использовали на уроках информатики только среду разработки *SchoolZum Информатика*.

В результате тестирования было получено много обратной связи в виде нахождения ошибок и советов по улучшению, большая часть которой уже реализована. Например, было предложено создать контекстное меню с функциями «скопировать», «вставить», «вырезать» при нажатии ПКМ в редакторе кода, что и было реализовано. Также были найдены и исправлены ошибки копирования текста из вкладки «Запуск», отображения примеров ввода/вывода, фокуса курсора на поле ввода при неудачной попытке авторизации.

Показатели назначения проекта можно увидеть в Таблице 3.

Таблица 3. Показатели назначения проекта

	Без SchoolZum Информатики	Со SchoolZum Информатикой	Предполагаемый	Полученный
Среднее количество решенных задач за урок	4.5	5.3	Прирост 10%	Прирост 17.78%
Количество проверок задач учителем	31	0	Падение на 100%	Падение на 100%

Заключение

В этом проекте представлено описание разработанной автором за период июль 2023 – февраль 2025 специализированной среды программирования *SchoolZum Информатика* для поддержки школьного курса информатики. Проанализированы существующие среды для обучения программированию на Python, выявлены их недостатки и определены требования к новой образовательной платформе.

Разработанная среда *SchoolZum Информатика* предоставляет ученикам интуитивно понятный интерфейс и интеграцию с образовательными ресурсами. Внедрение автоматизированной системы проверки кода в *SchoolZum Информатике* позволяет ученикам сразу получать объективную оценку своих программ, а преподавателям — сосредоточиться на педагогической деятельности.

Для обеспечения удобного распространения программного обеспечения была создана отдельная веб-страница на портале *SchoolZum* [26].

В дальнейшем предполагается расширение функциональных возможностей платформы, включая создание «Домашней» и «Школьной» версий приложения, расширение адаптивности интерфейса для повышения удобства использования, реализация аналитических инструментов для мониторинга прогресса учеников, создание системы проверки знаний и тестирования, добавление теоретических материалов как по программированию, так и по другим смежным темам.

Список литературы

1. Лутц М. Изучаем Python, 4-е изд. / пер. с англ. СПб.: Символ-Плюс, 2011. 1280 с.
2. Васильев Д. А. Методические особенности изучения языка Python школьниками // Символ науки. 2017. №1. URL: <https://cyberleninka.ru/article/n/metodicheskie-osobennosti-izucheniya-yazyka-python-shkolnikami>.
3. Thonny — идеальная IDE для обучения программированию на Python в школах детей [Электронный ресурс] — Режим доступа: <https://wowgold-seller.com/ru/stories/8749-thonny-is-an-ideal-ide-for-teaching-python-programming-in-sc>, свободный.
4. IDLE [Электронный ресурс] — Режим доступа: <https://docs.python.org/3/library/idle.html>, свободный.
5. PyCharm for Education [Электронный ресурс] — Режим доступа: <https://www.jetbrains.com/pycharm-edu/>, свободный.

6. VS Code for Educators and Students [Электронный ресурс] — Режим доступа: <https://code.visualstudio.com/docs/getstarted/educators-and-students>, свободный.
7. Зубков О. В., Семичева Н. Л. Роль автоматизированных тестирующих систем в процессе формирования навыков алгоритмического мышления // Педагогический ИМИДЖ. 2019. №4 (45). URL: <https://cyberleninka.ru/article/n/rol-avtomatizirovannyh-testiruyuschih-sistem-v-protssesse-formirovaniya-navykov-algoritmicheskogo-myshleniya>.
8. Русакова М. С., Меерсон Р. И. Проектирование системы автоматизированной проверки решений для проведения соревнований по программированию // Вестник СамГУ. 2013. №3 (104). URL: <https://cyberleninka.ru/article/n/proektirovanie-sistemy-avtomatizirovannoy-proverki-resheniy-dlya-provedeniya-sorevnovaniy-po-programmirovaniyu>.
9. Горчаков Л. В., Стась А. Н., Карташов Д. В. Обучение программированию с использованием системы Ejudge // Вестник ТГПУ. 2017. №9 (186). URL: <https://cyberleninka.ru/article/n/obuchenie-programmirovaniyu-s-ispolzovaniem-sistemy-ejudge>.
10. Боженкова Е. Н., Иртегов Д. В., Киров А. В., Нестеренко Т. В., Чурина Т. Г. Автоматизированная система тестирования NSUts: требования и разработка прототипа // Вестник НГУ. Серия: Информационные технологии. 2010. №4. URL: <https://cyberleninka.ru/article/n/avtomatizirovannaya-sistema-testirovaniya-nsuts-trebovaniya-i-razrabotka-prototipa>.
11. Долгушин Н.А., Оленькова М.Н. Использование системы Contester для проведения олимпиад по программированию // Международный студенческий научный вестник. — 2016. — № 3-2. URL: <https://eduherald.ru/ru/article/view?id=14960>.
12. Codeforces [Электронный ресурс] — Режим доступа: <https://codeforces.net/>, свободный.
13. LeetCode - The World's Leading Online Programming Learning Platform [Электронный ресурс] — Режим доступа: <https://leetcode.com/>, свободный.
14. Codewars - Achieve mastery through coding practice and developer mentorship [Электронный ресурс] — Режим доступа: <https://www.codewars.com/>, свободный.
15. Timus Online Judge [Электронный ресурс] — Режим доступа: <https://acm.timus.ru/>, свободный.
16. Информатикс [Электронный ресурс] — Режим доступа: <https://informatics.msk.ru/>, свободный.
17. SchoolZum — Школьная Информационная Экосистема [Электронный ресурс] — Режим доступа: <https://schoolzum.ru/>, свободный.

18. ElectronJS [Электронный ресурс] — Режим доступа: <https://www.electronjs.org/>, свободный.
19. MongoDB — Википедия [Электронный ресурс] — Режим доступа: <https://ru.wikipedia.org/wiki/MongoDB>, свободный.
20. Flask (веб-фреймворк) — Википедия [Электронный ресурс] — Режим доступа: [https://ru.wikipedia.org/wiki/Flask \(веб-фреймворк\)](https://ru.wikipedia.org/wiki/Flask_(веб-фреймворк)), свободный.
21. Nginx [Электронный ресурс] — Режим доступа: <https://nginx.org/ru/>, свободный.
22. Курс «Задачи из учебника К.Ю. Полякова и Е.А. Еремина» [Электронный ресурс] — Режим доступа: <https://informatics.msk.ru/course/view.php?id=666>, свободный.
23. PEP 8 — Style Guide for Python Code [Электронный ресурс] — Режим доступа: <https://peps.python.org/pep-0008/>, свободный.
24. microsoft/debugpy: An implementation of the Debug Adapter Protocol for Python [Электронный ресурс] — Режим доступа: <https://github.com/microsoft/debugpy>, свободный.
25. Official page for Debug Adapter Protocol [Электронный ресурс] — Режим доступа: <https://microsoft.github.io/debug-adapter-protocol/>, свободный.
26. SchoolZum Информатика [Электронный ресурс] — Режим доступа: <https://schoolzum.ru/informatics>, свободный.