**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 7](#_Toc187267412)

[1 Постановка задачи 8](#_Toc187267413)

[2 Существующие аналоги 9](#_Toc187267414)

[2.1 Обзор возможностей существующих аналогов 9](#_Toc187267415)

[2.2 Обзор способов автоматизированной проверки решений задач 15](#_Toc187267416)

[3 Обзор литературы 18](#_Toc187267417)

[3.1 Безопасность выполнения недоверенного кода 18](#_Toc187267418)

[ЗАКЛЮЧЕНИЕ 20](#_Toc187267419)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 21](#_Toc187267420)

ВВЕДЕНИЕ

В современном мире информационные технологии играют ведущую роль в развитии различных отраслей экономики и науки. Следовательно, спрос на IT-специалистов, включая разработчиков программного обеспечения, постоянно растет. В связи с этим, возникает необходимость в подготовке новых специалистов в области программирования. Образовательные информационные системы с алгоритмическими задачами по программированию становятся неотъемлемой частью этого процесса подготовки. Эти системы предоставляют студентам и начинающим программистам возможность развивать свои навыки, решая разнообразные задачи с различным уровнем сложности. Благодаря таким системам студенты могут улучшить свои навыки алгоритмизации, научиться эффективно решать задачи и практиковаться в программировании на различных языках. Более того, образовательные информационные системы обеспечивают доступ к обучающим материалам, а также предоставляют возможность соревноваться с другими пользователями, что мотивирует студентов к развитию и позволяет им оценить свой прогресс в сравнении с другими. Такие системы также часто предлагают системы автоматической проверки решений, что позволяет студентам получать обратную связь и исправлять ошибки, улучшая свои навыки программирования.

# **1 Постановка задачи**

В рамках дальнейшего научного исследования планируется разработать информационную систему обучения и проверки навыков программирования «SupremeCode» с автоматизированной проверкой решений задач.

Объектом исследования являются информационные системы обучения и проверки навыков программирования.

Предметом исследования является процесс разработки информационной системы обучения и проверки навыков программирования.

В связи с определенными выше объектом и предметом исследования была поставлена цель работы, которая заключается в разработке информационной системы обучения программированию.

Для реализации поставленной цели сформулированы следующие задачи:

* проанализировать аналоги будущего приложения;
* выбрать модель проверки решений задач;
* изучить возможность безопасного выполнения недоверенного кода в контейнерах;
* разработать серверное приложение, обрабатывающее запросы пользователей и взаимодействующее с базой данных;
* реализовать интерфейс, который позволит пользователю взаимодействовать с системой и решать задачи;
* провести анализ полученных результатов.

# **2 Существующие аналоги**

Существует множество информационных систем для обучения программированию с автоматизированной проверкой решений задач, однако во время анализа существующих решений всегда можно выявить их недостатки. Именно они создают предпосылки для разработки нового решения, которое будет более полно соответствовать потребностям пользователей.

## **2.1 Обзор возможностей существующих аналогов**

На сегодняшний день существует множество онлайн-платформ для решения алгоритмических задач и проверки знаний в области программирования. Одной из самых известных и популярных платформ на международном уровне является «LeetCode» [1], а в России — «Информатикс» [2].

Эти две платформы значительно отличаются по своим подходам и целевым аудиториям. «Информатикс» сосредоточена на дистанционном обучении в области информатики и предлагает разнообразные курсы, задачи и тесты для школьников и студентов, а также для учителей. Эта платформа ориентирована на обучение и развитие студентов, позволяя преподавателям контролировать и отслеживать результаты учащихся.

В отличие от «Информатикс», «LeetCode» поддерживает большее количество языков программирования, что делает платформу более доступной для различных категорий программистов. В то же время сложность задач является более высокой, что увеличивает порог входа и уменьшает возможности решения слабо подготовленным студентам. Также отмечается формат задач: в частности, на платформе «Информатикс» менее обширное описание, нет статистики, задачи не делятся по тегам, напротив, в пользу данной платформы говорит наличие сопутствующих теоретических материалов в помощь студентам.

Важным преимуществом «LeetCode» является наличие встроенного редактора с подсветкой синтаксиса и возможностью тестирования программы на предоставленных данных, что облегчает процесс отладки и проверки кода. В отличие от этого, «Информатикс» предоставляет лишь кнопку для загрузки файла, что вынуждает пользователя писать и отлаживать код локально.

Также стоит упомянуть платформу «Codewars» [3], которая предлагает задачи для программистов всех уровней, позволяя участникам повышать свои навыки посредством обучения через решение задач, созданных сообществом. Подобно «LeetCode», «Codewars» поддерживает различные языки программирования и позволяет пользователям сравнивать свои решения с решениями других участников, что способствует улучшению своих навыков. Более того, «Codewars» предоставляет возможность создавать свои собственные задачи, что привлекает более опытных программистов и дает возможность сообществу активно участвовать в создании новых задач для платформы.

Платформа «HackerRank»[4] фокусируется на задачах для практики навыков программирования и технических собеседований. Платформа ориентирована как на индивидуальных пользователей, так и на компании, предоставляя инструменты для тестирования навыков кандидатов. Поддерживаются задачи на множество языков программирования, а также есть возможность участия в соревнованиях. Встроенные редакторы и тестовые среды позволяют эффективно решать задачи и оттачивать навыки.

С точки зрения пользовательского интерфейса, «Информатикс» не дотягивает до современных стандартов, так как его дизайн устарел, а главная страница переполнена информацией, затрудняющей навигацию. Поиск задач также затруднен, поскольку они обозначены номерами, которые часто известны только преподавателям. Интерфейс страницы задачи представлен на рисунке 1.1.

В отличие от него, «LeetCode» предлагает современный дизайн и тщательно продуманный список задач с различными возможностями сортировки, тегирования и поиска по темам. Это упрощает пользователям поиск необходимых задач и улучшает общее взаимодействие с платформой. Интерфейс страницы задачи представлен на рисунке 1.2.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 1.1 – Платформа «Информатикс»

«Codewars» также отличается удобным пользовательским интерфейсом, ориентированным на задачи для программистов всех уровней. Платформа предоставляет возможность поиска задач по разным уровням сложности, языкам программирования и темам. Кроме того, интерфейс предоставляет возможность взаимодействия с сообществом путем обсуждения задач и решений, что повышает качество обучения и способствует обмену знаниями между участниками. Интерфейс страницы задачи представлен на рисунке 1.3.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 1.2 – Платформа «LeetCode»

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 1.3 – Платформа «CodeWars»

Платформа «HackerRank» обладает удобным и интуитивно понятным пользовательским интерфейсом, ориентированным на задачи для программистов различных уровней подготовки. Платформа предлагает структурированный каталог задач, которые можно сортировать по темам, уровням сложности и языкам программирования.

Интерфейс каждой задачи включает подробное описание условия, вводные и выходные данные, а также примеры с пояснениями. Встроенный редактор кода с подсветкой синтаксиса и интегрированным запуском позволяет пользователям тестировать свои решения на реальных данных без необходимости использования сторонних инструментов.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 1.4 – Платформа «HackerRank»

Платформа «LeetCode» имеет значительное преимущество перед другими платформами благодаря наличию онлайн-компилятора (playground), который позволяет выполнять пользовательские программы вне решения задач. Этот инструмент предоставляет разработчикам гибкость и свободу для экспериментов с кодом в интерактивной среде, что повышает качество обучения и развития навыков.

Онлайн-компилятор позволяет пользователям проверять, тестировать и оптимизировать свои алгоритмы и решения без давления или ограничений конкретной задачи. Это дает возможность разработчикам работать над собственными проектами или идеями, применять новые концепции и подходы к программированию, а также исследовать различные языки программирования.

Интерактивный характер онлайн-компилятора позволяет пользователям сразу видеть результаты выполнения своего кода, что ускоряет процесс обучения и отладки. Это также позволяет пользователям проверять эффективность и производительность своего кода, сравнивая различные подходы к решению задач.

Кроме того, наличие онлайн-компилятора помогает пользователям развивать навыки программирования в более широком контексте, чем просто выполнение конкретных задач. Это стимулирует творчество и инновации, позволяя пользователям экспериментировать и исследовать новые способы решения проблем.

Еще одной интересной возможностью является совместная работа над решением задачи в реальном времени. Это позволяет пользователям обмениваться идеями, исправлять ошибки друг друга и коллективно находить наилучшие решения.

При использовании функции совместного редактирования пользователи могут одновременно просматривать, редактировать и анализировать код друг друга. Это способствует обмену опытом и знаниями, а также повышает эффективность работы в команде.

Преимущества совместного редактирования кода включают возможность мгновенного обратного связи, ускорение процесса разработки и улучшение качества кода за счет совместного обсуждения и корректировки. Кроме того, это помогает развивать навыки командной работы и обучаться новым подходам к решению задач.

На данный момент ни одна из рассматриваемых платформ не имеет возможности совместного редактирования в реальном времени.

## **2.2 Обзор способов автоматизированной проверки решений задач**

Автоматизированное тестирование играет ключевую роль в проверке решений задач на платформах программирования. В данном разделе рассматриваются подходы к формату тестовых данных и процессу добавления задач на платформах «LeetCode», «CodeWars», «HackerRank» и «Информатикс».

Платформа «CodeWars» предоставляет возможность автоматизированной проверки решений задач через указание unit-тестов[5]. На листинге 1.1 представлен пример функции с решением. На листинге 1.2 представлен Unit-тест для проверки решения, описанного в листинге 1.1.

Листинг 1.1 – Функция с решением задачи на «CodeWars»

function rgb(r, g, b) {

return "FFFFFF";

}

Листинг 1.2 – Unit-тест для проверки решения задачи на «CodeWars»

describe("Tests", () => {

const { strictEqual } = require('chai').assert;

function doTest(r, g, b, expected) {

const actual = rgb(r, g, b);

const message = `for r = ${r} g = ${g} b = ${b}`;

strictEqual(actual, expected, message);

}

it("Sample Tests", () => {

doTest( 0, 0, 0, '000000');

doTest( 0, 0, -20, '000000');

doTest(300, 255, 255, 'FFFFFF');

doTest(173, 255, 47, 'ADFF2F');

});

});

Проверка решений через unit-тесты позволяет очень гибко проверять функциональность программы.

На платформе «HackerRank» проверка решений основывается на стандартном вводе/выводе [6]. Система принимает тестовые данные через стандартный ввод (stdin) и проверяет совпадение результатов стандартного вывода (stdout) с ожидаемыми.

Это позволяет пользователям сосредоточиться на алгоритме, не задумываясь о сложных форматах тестирования. Тем не менее, таким образом нельзя проверить некоторые виды задач, как например, наличие методов в классе, проверка их сигнатуры. Такой метод проверки решений больше ориентирован на алгоритмические задачи.

Платформа «Информатикс» предоставляет такой же функционал проверки по stdin/stdout, как и «HackerRank».

Платформа «LeetCode» не предоставляет возможности добавления пользовательских задач, поэтому схему проверки решений можно проанализировать по пользовательскому образцу теста. Для шаблона решения задачи представленного на листинге 1.3 можно указать тестовые данные в тестовом формате, как на листинге 1.4.

Листинг 1.3 – Функция с решением задачи на «LeetCode»

vector<int> twoSum(vector<int>& nums, int target){}

Листинг 1.4 – Тестовые данные в текстовом формате на «LeetCode»

nums = [2,7,11,15]  
target = 9

Такой формат предоставляет единообразность тестовых данных для разных языков и некоторой степени лучше проверки по stdin/stdout из-за ориентации на сигнатуру функции.

В этой главе были рассмотрены способы автоматизированной проверки решений. Самым удобным для алгоритмических задач является формат как у «LeetCode». Для обучения программированию больше подходит проверка через unit-тесты, как у «CodeWars».

# **3 Обзор литературы**

Глава 3 содержит информацию о тех источниках, которые понадобятся для написания диссертационной работы.

## **3.1 Безопасность выполнения недоверенного кода**

Тема безопасности на платформах, где запускается пользовательский код, является особенно актуальной. Запуск пользовательского кода создает потенциальные риски, включая возможность выполнения вредоносного кода и несанкционированного доступа к данным или ресурсам системы. Решением данной проблемы является применение контейнеров приложений. Docker [1] — это платформа для контейнеризации, которая позволяет создавать, развертывать и управлять приложениями в изолированных средах, называемых контейнерами. Контейнер — это легковесный, автономный пакет, содержащий все необходимое для запуска приложения, включая код, библиотеки, зависимости и конфигурации.

Книга Найджела Поултона «Docker Deep Dive Zero to Docker in a single book» позволяет ознакомиться с тем, как работать с образами, томами, контейнерами и сетями, как обеспечить безопасность контейнеров и инфраструктуры. Хорошим дополнением является официальная документация Docker [8].

Контейнеры предоставляют возможность развертывать приложения в согласованной и предсказуемой среде, независимо от того, где они запускаются — на локальной машине, в облаке или на других серверах. Возможности контейнеров включают изоляцию процессов, файловой системы и сетевых интерфейсов, что позволяет им работать в среде, отделенной от хост-системы и других контейнеров.

Использование контейнеров для изоляции автоматизированной проверки на онлайн-платформах алгоритмических задач может существенно улучшить безопасность и стабильность платформы. Контейнеры позволяют запускать каждое решение пользователя в отдельной изолированной среде, тем самым ограничивая возможные взаимодействия между кодом пользователя и основной инфраструктурой платформы. Это предотвращает потенциальные атаки, такие как доступ к системным файлам хост-системы или запуск вредоносного кода.

Контейнеры также позволяют контролировать ресурсы, такие как процессорное время и память, которые доступны для каждого контейнера. Это обеспечивает справедливые условия для всех пользователей и предотвращает злоупотребление ресурсами. Изолированная среда контейнера также позволяет платформе применять ограничения безопасности, такие как запрет на определенные системные вызовы, что дополнительно снижает риск вредоносных действий.

В целом, использование контейнеров для изоляции автоматизированной проверки на онлайн-платформах алгоритмических задач повышает безопасность и надежность платформы, обеспечивая при этом справедливую и предсказуемую среду для всех пользователей. Это способствует защите данных и ресурсов платформы, а также улучшает опыт работы пользователей.

ЗАКЛЮЧЕНИЕ

В рамках данной работы сформулирована тема диссертационной работы, определены объект, предмет и цель исследования, поставлены задачи по ее достижению.

Проведен подбор и анализ существующих аналогов информационных систем с автоматизированной проверкой решений задач. Проведен анализ литературных источников по применению контейнеров для безопасного запуска недоверенного кода.

Недостатки существующих платформ достаточно значительны и подчеркивают необходимость создания собственной онлайн-платформы «SupremeCode».

Цель разрабатываемой платформы – предоставить пользователям возможность учиться программированию и информационным технологиям с возможностью практического решения задач, которые будут автоматически проверяться.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Leetcode платформа с задачами [Электронный ресурс],   
   URL: https://leetcode.com/ (дата обращения: 14.12.2024).
2. Информатикс Олимпиадное программирование [Электронный ресурс],   
   URL: https://informatics.msk.ru/ (дата обращения: 14.12.2024).
3. Codewars Обучение программированию [Электронный ресурс],  
   URL: https://www.codewars.com/dashboard (дата обращения: 14.12.2024).
4. HackerRank Онлайн тесты по программированию [Электронный ресурс], URL: https://www.hackerrank.com/ (дата обращения: 14.12.2024).
5. Модульное тестирование [Электронный ресурс], URL: https://ru.wikipedia.org/wiki/Модульное\_тестирование (дата обращения: 14.12.2024).
6. Стандартные потоки [Электронный ресурс], URL: https://ru.wikipedia.org/wiki/Стандартные\_потоки (дата обращения: 14.12.2024).
7. Найджел Поултон. Docker Deep Dive Zero to Docker in a single book 2020.
8. Документация Docker. Электронный ресурс. URL: https://docs.docker.com/engine (дата обращения: 14.12.2024).