Мир науки. Педагогика и психология / World of Science. Pedagogy and psychology https://mir-nauki.com

2023, Том 11, № 6 / 2023, Vol. 11, Iss. 6 https://mir-nauki.com/issue-6-2023.html

URL статьи: https://mir-nauki.com/PDF/40PDMN623.pdf

5.8.2. Теория и методика обучения и воспитания (по областям и уровням образования) (педагогические науки)

Ссылка для цитирования этой статьи:
Лучанинов, Д. В. Система автоматизированной проверки решений заданий по программированию с функцией локального антиплагиата / Д. В. Лучанинов, Р. И. Баженов, Д. В. Фатеенков, А. С. Дорофеев // Мир науки. Педагогика и психология. — 2023. — Т. 11. — № 6. — URL: https://mir-nauki.com/PDF/40PDMN623.pdf

For citation:
Luchaninov D.V., Bazhenov R.I., Fateenkov D.V., Dorofeev A.S. The system of automated verification of programming problems' solutions with the local anti-plagiarism function. *World of Science. Pedagogy and psychology*. 2023; 11(6): 40PDMN623. Available at: https://mir-nauki.com/PDF/40PDMN623.pdf. (In Russ., abstract in Eng.)

УДК 378

ГРНТИ 14.35.09

Лучанинов Дмитрий Васильевич
ФГБОУ ВО «Приамурский государственный университет имени Шолом-Алейхема», Биробиджан, Россия
Старший преподаватель кафедры «Информационных систем, математики и правовой информатики»
E-mail: dvluchano@mail.ru
РИНЦ: https://elibrary.ru/author_profile.asp?id=622813

Баженов Руслан Иванович
ФГБОУ ВО «Приамурский государственный университет имени Шолом-Алейхема», Биробиджан, Россия
Заведующий кафедрой «Информационных систем, математики и правовой информатики»
Кандидат педагогических наук, доцент
E-mail: r-i-bazhenov@yandex.ru
РИНЦ: https://elibrary.ru/author_profile.asp?id=642728

Фатеенков Данила Витальевич
ФГБОУ ВО «Приамурский государственный университет имени Шолом-Алейхема», Биробиджан, Россия
E-mail: wiosna97@yandex.ru

Дорофеев Андрей Сергеевич
ФГБОУ ВО «Иркутский национальный исследовательский технический университет»
Доцент лаборатории аппаратных и программных средств вычислительной техники
Кандидат технических наук, доцент
E-mail: dorbaik2007@mail.ru
РИНЦ: https://elibrary.ru/author_profile.asp?id=189500

Система автоматизированной проверки решений заданий по программированию с функцией локального антиплагиата

Аннотация. Целью данной работы является изучение процессов, направленных на конструирование в веб-среде обучения программированию студентов образовательных организаций высшего образования с помощью информационной системы, автоматизирующей проверку отправленного кода к практическим заданиям в рамках выбранной дисциплины. В исследовании рассмотрены частые тонкости, возникающие в рамках практического применения систем проверки решений и показана реализация, способная устранить возможные затруднения. В рамках работы также рассмотрены основные критические моменты, которые

должны быть учтены разработчиком при разработке системы. Изучены основные функциональные элементы информационной системы, такие как: добавление методических материалов, добавление заданий практической направленности, учёт успеваемости обучающихся, просмотр отправленных учащимися решений, проверка решений на факт заимствований (локальный антиплагиат). Также описаны принципы проверки решений, отправляемых студентами: рассмотрены возможные результаты тестирования решений и проблемы, которые могут возникнуть на этапе компиляции и проверки работоспособности отправленного кода. Проведена экспериментальная оценка эффективности использования разработанной системы в течение весеннего семестра 2022–2023 учебного года и осеннего семестра 2023–2024 учебного года в Приамурском государственном университете имени Шолом-Алейхема и Иркутском национальном исследовательском техническом университете, в исследовании участвовало 42 и 57 студентов в контрольной и экспериментальной группах соответственно. Результаты исследований показывают, что в экспериментальной группе улучшение составило 21 процент, в то время как в контрольной группе улучшение дало лишь 12 процентов. Разработанная информационная система на данный момент находится в процессе подтверждения свидетельства о государственной регистрации программы для ЭВМ.

Ключевые слова: автоматизированная проверка решений; конструирование электронных курсов; самостоятельная работа студента; обучение программированию; автоматизация обучения программированию; локальный антиплагиат; информационная система

Введение

Современное развитие информационных технологий стремительно преобразует сферу образования, представляя новые возможности по организации процесса обучения студентов различных направлений. Традиционные методы обучения программированию постепенно утрачивают свою актуальность, уступая место системам, автоматизирующим процессы создания и изучения различных курсов, включая программирование [1–3]. Образовательные учреждения активно внедряют онлайн-подходы в преподавание. В данные процессы также входит использование систем массовых открытых онлайн-курсов (от англ. аббр. MOOK [4]), использование систем управления обучением (от англ. аббр. LMS [5]), включая обучение программированию с автоматизацией проверки отправляемого пользователями кода.

Автоматизация процесса организации обучения программированию в онлайн среде является актуальной темой в настоящее время, так как она позволяет упростить работу для преподавателей и сделать процесс обучения намного интереснее для обучающихся. Подобные системы выросли из подхода к реализации соревнований по спортивному программированию с массовым участием [6–9]. В настоящее время существует несколько информационных систем (ИС), предоставляющие инструменты для создания собственных обучающих курсов и позволяющих организовать учебный процесс [10]. Проблема таких систем заключается в их подходе — из-за их открытой модели распространения отсутствуют частные модули и от этого функционал может не удовлетворять требованиям преподавателя.

Разработка системы, направленной на автоматизацию процесса обучения программированию, является достаточно комплексной задачей, при решении которой необходимо учесть большое количество критических моментов. В первую очередь нужно внимательно изучить вопрос безопасности в подобной системе — пользователи могут отправлять вредоносный код, который может навредить системе или серверу, на котором расположена информационная система. Для решения данной проблемы существуют различные методы изоляции запускаемого кода, предоставляющие различный набор инструментариев для реализации виртуальных сред [11].

Немаловажно также учесть подход к организации проверки отправляемых пользователями решений. Для проверки часто используется модель проверки на заранее заготовленных тестах, в которых описаны входные и выходные данные [12]. В изолированной среде код после запуска будет обрабатывать входные значения, которые будут подаваться на ввод с помощью отдельных программных модулей.

Процесс проверки отправленных пользователями решений часто нуждается в доработке и дополнительных оптимизациях [13]. Это связано, в первую очередь, с разнообразием языков программирования и направлений подготовки. В различных IT сферах программы создаются для выполнения уникальных между собой задач, что требует постоянного расширения системы, чтобы у преподавателей была возможность организовать обучение по всем доступным в учебном заведении направлениям подготовки.

Разработки в данной сфере ведутся уже много лет. На данный момент представлены системы для организации обучения программированию в самых разных вариациях, например, в виде API сервиса, на основе которого можно сконструировать собственную обучающую систему [14].

Ещё одной важной задачей, которую решает разработчик во время создания системы, автоматизирующей процесс проверки решений задач практического характера, это проектирование архитектуры системы [15]. Важно спроектировать оптимальную по денежным затратам систему, которая будет эффективно обрабатывать отправляемую в неё информацию и минимизирующей сбои во время выполнения основных задач.

Целью данной работы является изучение процессов, направленных на организацию в веб-среде обучения программированию студентов высших учебных заведений с помощью информационной системы, предоставляющей функционал для проведения учебных занятий по профильным дисциплинам и автоматизирующей проверку отправленных решений в рамках созданных дисциплин.

Материалы и методы

Рассматриваемая в рамках данного исследования информационная система позволяет автоматизировать процесс проверки решений к практическим задачам, написанных студентами на различных языках программирования. Система предоставляет весь необходимый функционал для создания обучающих и интерактивных курсов и дисциплин: студент может просматривать доступные материалы для самостоятельного изучения, а также решать задания для самостоятельной подготовки; а преподаватель может создавать дисциплины, наполнять их материалами для обучения, отслеживать ход работы каждого студента и просматривать отправленные пользователями решения.

В первую очередь необходимо решить проблему запуска кода в системе. Стоит учитывать, что пользователь может послать вредоносный код, который может замедлить или прервать работу ИС. Для предотвращения данной ситуации был выбран Docker. Он позволяет создать изолированную среду для запуска кода, что является отличным решением при проектировании ИС, которая предполагает загрузку написанного пользователями кода. С помощью Docker Compose был сформирован контейнер, который включает в себя образы трёх языков программирования — C++ (GCC 9.5.0), Python (3.11.2), PHP (7.4.1).

Также был включён MySQL для хранения и обработки данных в системе. В том числе хранится информация о доступных в системе языках программирования — им присвоен уникальный целочисленный идентификатор, по которому система определяет образ, к которому необходимо обратиться после отправки решений. В системе решения хранятся в закодированном виде, чтобы минимизировать занимаемое дисковое пространство на сервере.

Удобство использования Docker также заключается и в автоматизированном процессе обновления версий языков программирования. Все образы хранятся в официальном Docker репозитории (Docker Hub) и после выхода новой версии можно за короткий промежуток времени обновить версии, при этом не тратя много времени на ручную установку. Но стоит учитывать, что, не смотря на такое упрощение доставки версий, необходимость отладки и администрирования системы не уходит на второй план, а значит тестирование и проверка работоспособности системы остаётся первостепенной задачей во время обновления компонентов системы.

Лекционные занятия: проводятся преподавателем в аудитории и на лекциях студенты изучают необходимый для дальнейшей работы материал. Не смотря на наличие таких занятий, необходимость предоставления студентам методических материалов остаётся важной частью процесса обучения программированию. Для этого в рассматриваемой системе есть возможность создания учебных разделов, в которых преподаватель может опубликовать лекционный материал или методическое пособие. Загруженный материал будет отображаться на странице раздела (рис. 1).

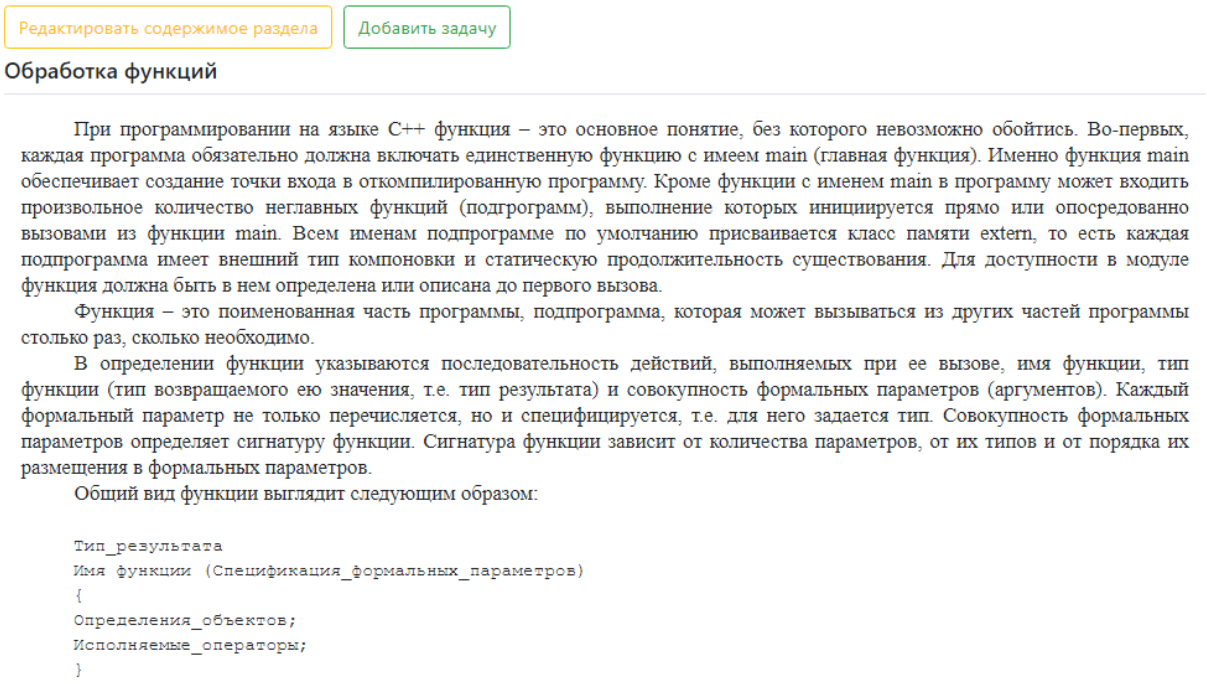


Рисунок 1. *Страница обучающего раздела по теме «Обработка функций» (разработано авторами)*

Так как на странице разделов информация (лекционный материал) может быть представлена в различных формах (текст, изображения, видео, таблицы и аудио), то необходимо использовать специальную форму для ввода (рис. 2). Формы, предоставляющие возможность составить мультимедийный материал, называются WYSIWYG (What You See Is What You Get).

В информационной системе используется такая форма при создании новых разделов и задач. Добавленная информация в эту форму сохраняется в HTML-файл на сервере и отображается на странице раздела, к которому привязана.

WYSIWYG форма позволяет также задавать форматирование текста: выбирать различные шрифты, задавать расположение объектов, задавать цвет текста, размер изображений и др.

Редактировать загруженный материал можно на странице учебного раздела — в системе предусмотрена возможность динамического обновления содержимого разделов.

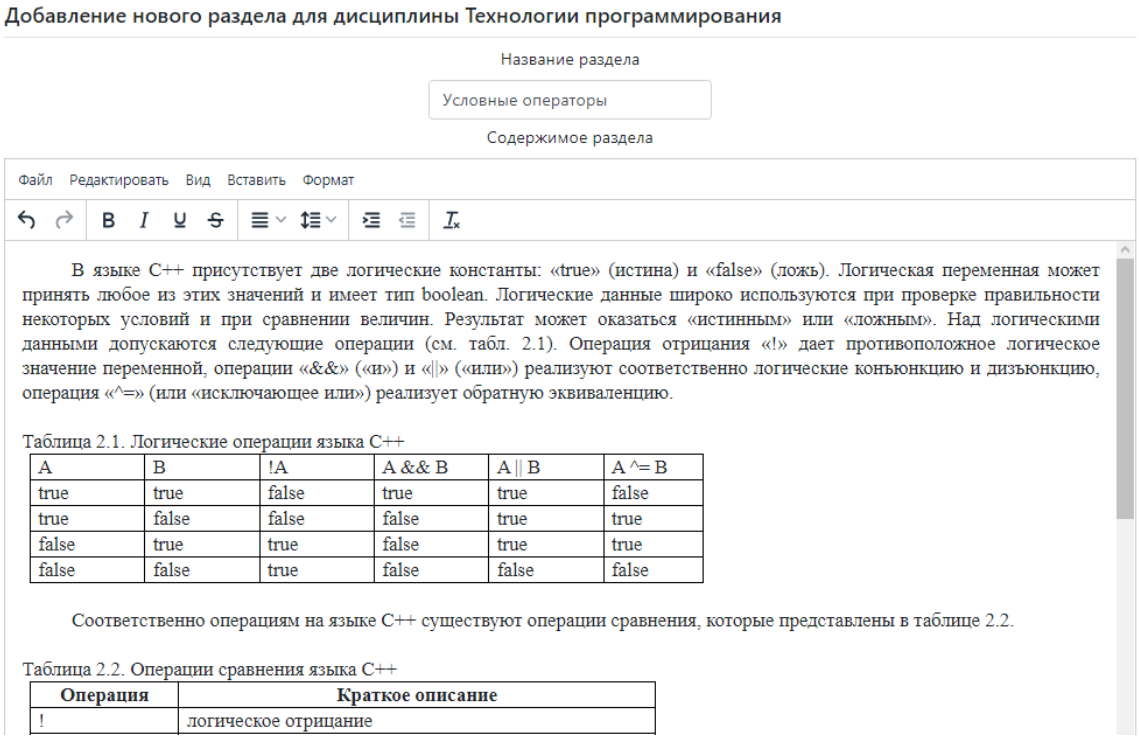


Рисунок 2. *Добавление обучающего материала по теме «Условные операторы» (разработано авторами)*

После загрузки разделов и обучающих материалов, список доступных разделов будет отображен на странице «О дисциплине» (рис. 3).

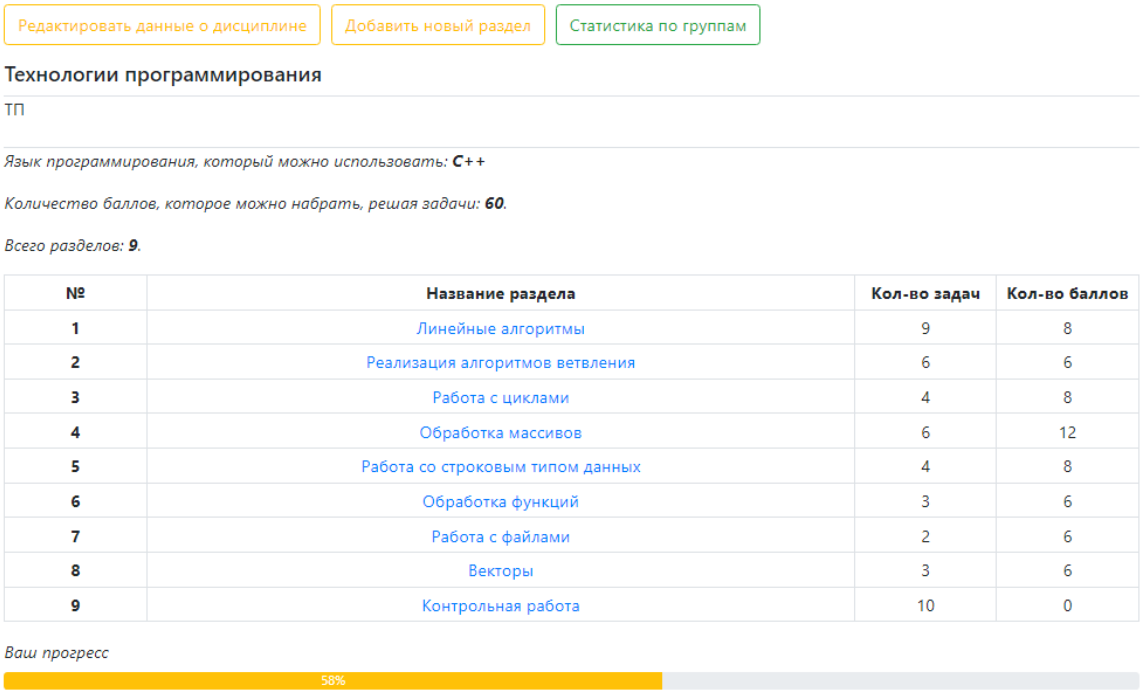


Рисунок 3. *Страница просмотра информации о дисциплине «Технологии программирования» (разработано авторами)*

40PDMN623
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

На данной странице студент может получить следующую информацию о дисциплине:

1. Язык программирования, который можно использовать в рамках выбранной дисциплины.
2. Максимальное количество баллов, которое можно набрать, решая практические задачи.
3. Список разделов, представленный в виде таблице. В таблице, помимо названия разделов, также представлена информация о количестве доступных задач и количестве баллов, которые можно набрать, решая задачи определённого раздела.

У преподавателей также в верхней части страницы (немного выше названия дисциплины) есть также элементы управления дисциплиной: редактирование дисциплины, добавление разделов и просмотр статистики определённой группы.

Помимо всего этого в нижней части страницы представлена шкала прогресса, с помощью которой студент может определить — как много он выполнил заданий и сколько ему ещё осталось до завершения прохождения дисциплины. Шкала учитывает количество набранных баллов и выводит значение в процентном соотношении, поэтому если у задачи стоит 0 баллов за правильное решение (это может быть тривиальная задача, например, задача сложения двух чисел. Количество баллов за правильное решение определяется преподавателем).

На странице разделов, после прочтения методического материала, студенту предлагается закрепить знания, решая задачи практического уровня. Задачи представлены в виде списка в нижней части страницы (рис. 4).

Задачи для самостоятельной подготовки:

Название задачи	Последнее решение	Количество баллов
4.1	Код	2
4.2	Код	2
4.3	Код	2
4.4	Код	2
4.5	-	2
4.6	Код	2

Рисунок 4. *Перечень задач для раздела «Обработка массивов» (разработано авторами)*

Если задача была решена правильно хотя бы один раз, то данная задача будет помечена зелёным цветом. Это нужно для того, чтобы студенту было легче ориентироваться в материалах дисциплины.

Также представлена следующая информация о задачах:

1. Название задачи.
2. Ссылка на последнее решение. При нажатии на ссылку будет открыта новая страница, на которой отобразится код решения и результат проверки (последнее решение также может быть неправильным). Если пользователь ещё не отправлял решений к задаче, то будет стоять на месте ссылки знак тире «-».
3. Количество баллов, которое можно набрать при решении задачи.

Задачи практического характера обладают следующими свойствами:

1. Название задачи. Нужно для идентификации их в системе (на стороне клиента).
2. Количество баллов за правильное решение. Нужно для отслеживания прогресса прохождения обучения.

3. Ограничения по времени. Для того, чтобы решение задачи не заняло слишком много времени и тем самым не замедлило работу ИС, была введена данная характеристика. Также полезно задавать данное ограничение, когда в задаче предполагается использование конкретного алгоритма или подхода к решению (например, динамическое программирование или метод двух указателей).

4. Ограничения по памяти. Чтобы решение не заняло слишком много пространства в оперативной памяти при проверке, необходимо ограничить до допустимых значений (например, до 64 Мб). Особенно полезно, когда в задаче, в силу своего контекста (например, алгоритм, в котором сохраняются значения типа данных «long long» в массив длиной 10^9), возможна утечка памяти на одном из этапов проверки.

5. Набор тестовых данных. Набор состоит из двух компонентов: входные и выходные значения. Под входными значениями предполагаются данные, которые должны подаваться на вход программы. Выходные значения — значения, которые программа должна вернуть на конкретном тесте. То есть результат работы программы на каждом из тестов должен соответствовать выходным значениям, которые предопределены для каждого входного набора данных.

Студент не может видеть все тесты к задаче. Сделано это для того, чтобы пользователи не могли написать решение, полностью опирающееся на заготовленные преподавателем тесты (такая программа может не учитывать определённые значения и работать неправильно). При этом на странице с информацией о задачах можно увидеть всю необходимую для написания решения информацию (рис. 5).

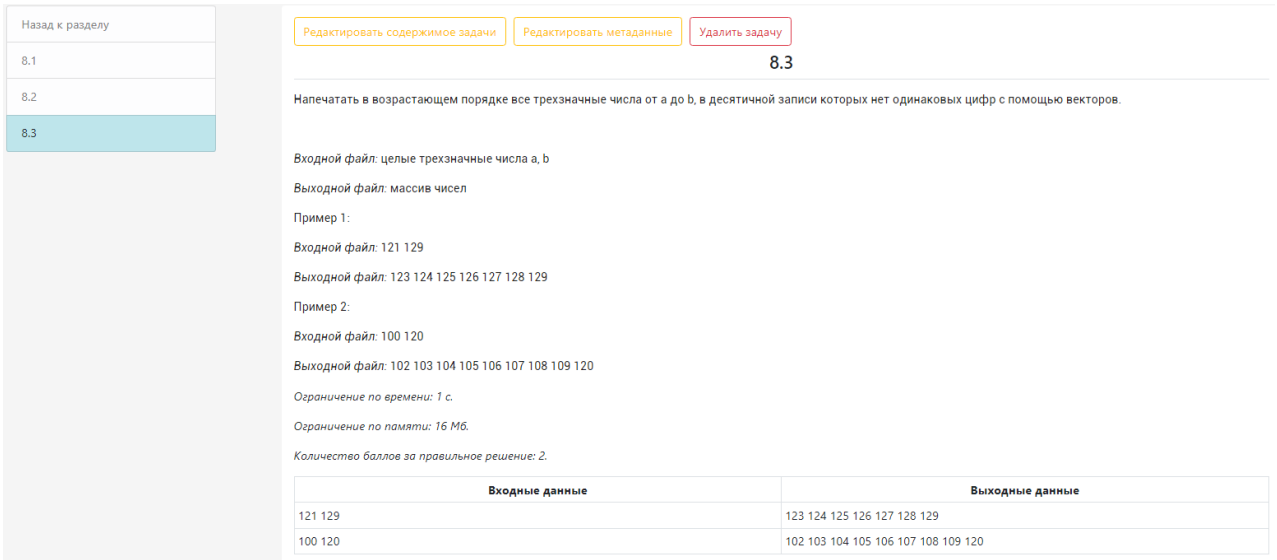


Рисунок 5. *Задача в разделе «Векторы в C++» (разработано авторами)*

На странице представлена следующая информация:

1. Название и содержание задачи (преподаватель, как и в случае с разделами, может публиковать не только текст, но и изображения и другую медиаинформацию на страницах с содержанием задачи).

2. Допустимые ограничения по времени и памяти.

3. Количество баллов за правильное решение.

4. Таблица с примером входных и выходных значений. Берутся первые 2 теста из заданного преподавателем набора данных.

Также на странице с задачей слева от блока с основной информацией выводится список всех задач раздела. Сделан этот список был для обеспечения удобства перемещения между задачами в рамках конкретного раздела.

В верхней части основного блока страницы расположены элементы управления задачами:

1. Редактирование условия задачи. Работает также, как и редактирование содержимого методических материалов.
2. Редактирование метаданных задачи. К метаданным относятся ограничения, количество баллов, тесты, а также дополнительные настройки (рис. 6).
3. Кнопка удаления задачи.

При нажатии на кнопку «Редактирование метаданных» на странице открывается окно с редактированием основных данных, связанных с выбранной задачей. В задачу можно добавлять новые тестовые данные, а также удалять уже добавленные. Также можно менять поток ввода/вывода данных в задачах (элемент выбора «Файловый поток»).

В задачах на данный момент можно использовать 2 потока для ввода и вывода данных:

1. Стандартный (консольный ввод).
2. Файловый ввод данных.

В системе представлена возможность ограничения использования потоков при решении задач. Если в условии задачи предполагается использования файлов, то стандартный поток не подойдёт для решения задачи и такое решение будет отклонено системой автоматически. Допускается использование входных и выходных файлов с названиями “input.txt” и “output.txt” соответственно.

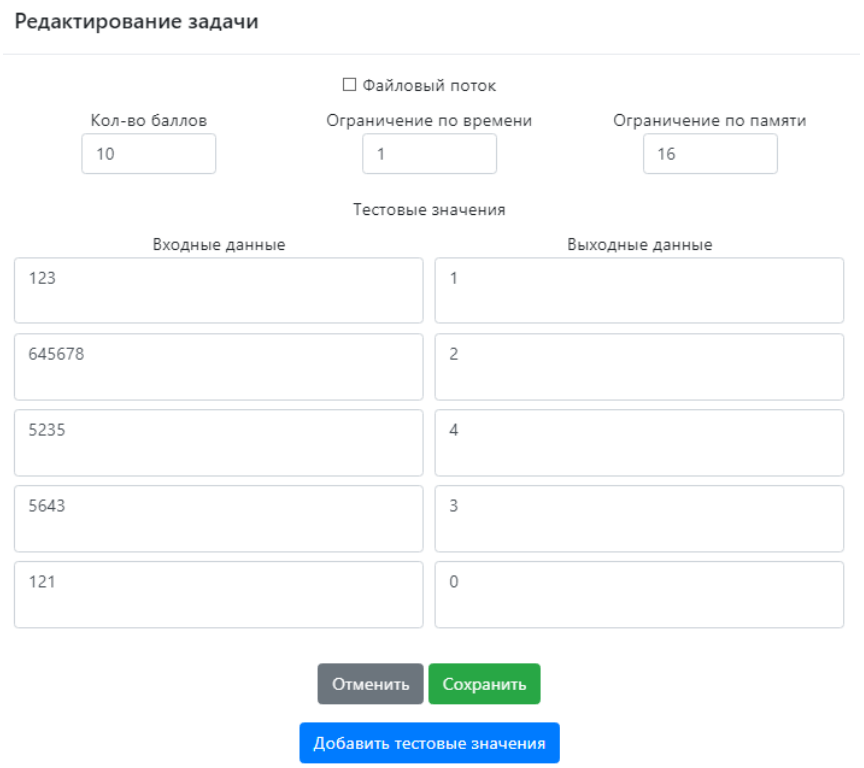


Рисунок 6. *Окно редактирования тестовой задачи в системе (разработано авторами)*

40PDMN623
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

Для того, чтобы пользователь мог различить — какой поток допускается к использованию в задаче, ему нужно посмотреть на таблицу примеров входных данных. В этой таблице в заголовочной строке будут прописаны названия файлов, если предполагается использование файловых потоков в выбранной задаче.

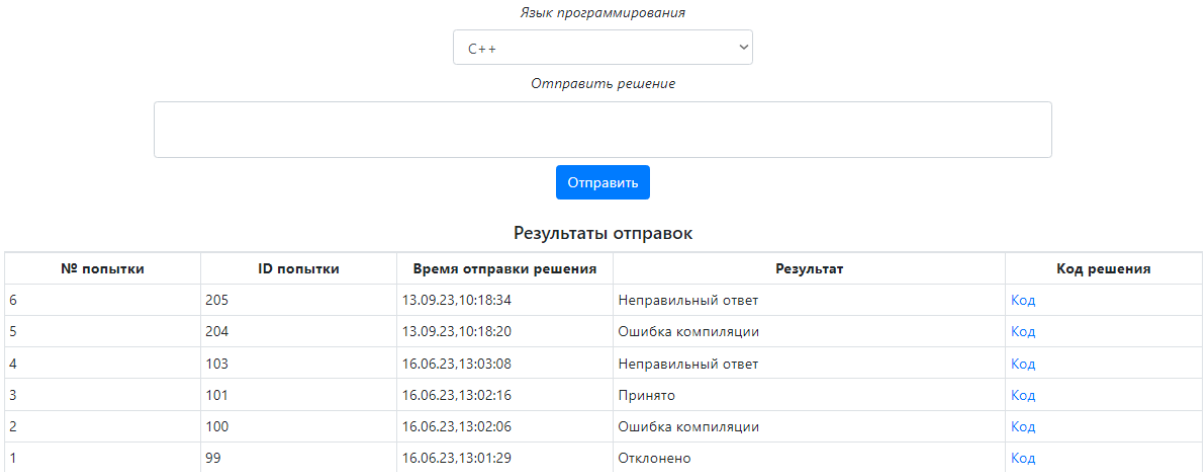
При этом стоит учитывать, что в задачах, в которых предполагается использование стандартных потоков, работа с файлами не допускается. Вывод из “output.txt” будет игнорироваться в таких задачах.

Окно редактирования метаданных не загружается на странице задачи, если пользователь обладает правом доступа «Студент», поэтому получить информацию о тестах задачи такой пользователь не сможет.

Ниже после условия задачи и примеров ввода и вывода данных расположена форма отправки решения и таблица со всеми отправленными решениями пользователем для выбранной задачи. В таблице представлена следующая информация:

1. Порядковый номер отправки (номер строки в таблице).
2. Идентификатор решения в системе. Может пригодиться, чтобы преподаватель мог найти решение в системе по ID.
3. Дата и время отправки решения.
4. Результат проверки решения.
5. Ссылка на решение.

Также перед полем отправки решения может располагаться выпадающий список, в котором студент может выбрать предпочтительный язык программирования. Такой список появляется только в тех дисциплинах, в которых преподаватель вместо выбора конкретного языка программирования выбрал вариант «На выбор студента». В таком случае студенту доступны все языки для решения практических задач (рис. 7).



№ попытки	ID попытки	Время отправки решения	Результат	Код решения
6	205	13.09.23,10:18:34	Неправильный ответ	Код
5	204	13.09.23,10:18:20	Ошибка компиляции	Код
4	103	16.06.23,13:03:08	Неправильный ответ	Код
3	101	16.06.23,13:02:16	Принято	Код
2	100	16.06.23,13:02:06	Ошибка компиляции	Код
1	99	16.06.23,13:01:29	Отклонено	Код

Рисунок 7. *Форма отправки решений*
и таблица результатов отправок (разработано авторами)

Все решения сохраняются в папку пользователя в виде текстового файла и могут быть получены студентом в любой момент времени через таблицу отправок. При нажатии в таблице на ссылку будет открыта страница, в которой будет представлена информации об отправленном решении: исходный код, ID решения, вердикт проверки. Также преподаватель на данной странице может вручную сменить вердикт, а именно отклонить (или принять) решение по определённым причинам (рис. 8).

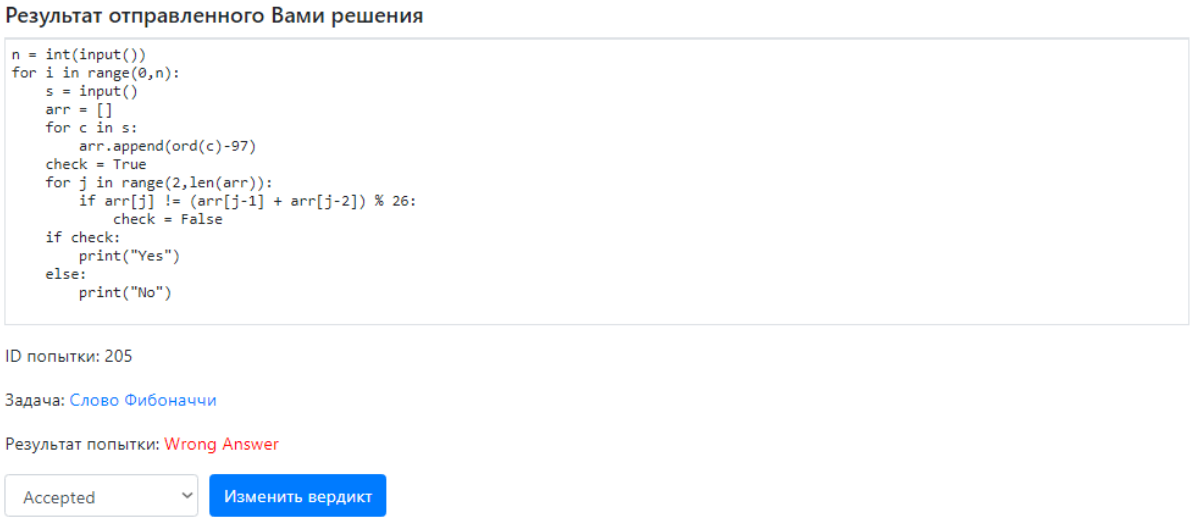


Рисунок 8. *Результат проверки решения задачи «Слово Фибоначчи» (разработано авторами)*

В системе представлены следующие результаты проверок отправленных студентами решений (описана краткая, полная записи вердиктов, а также их значение в системе):

1. OK или Accepted (принято). Решение прошло все тесты и засчитано системой как правильное.
2. WA или Wrong Answer (неправильный ответ). В ходе тестирования код вернул ответ, не совпадающим с тестом. При наличии хоть одной такой ошибки будет вынесен данный вердикт.
3. REJ или Rejected (отклонено проверяющей системой). Чаще всего данная ошибка возникает, если на сервере возникают неполадки. Причиной возникновения данной ошибки может служить в том числе и наличие ошибок во время проверки решения, которые нельзя отнести к описанным ниже. Также решения отклоняются, если в задаче, где предполагается использование файлов, используются стандартные потоки для ввода и вывода данных.
4. TL или Time Limit Exceeded (превышено ограничение по времени). Ошибка возникает в том случае, если решение работает слишком долго на определённых преподавателем тестах.
5. ML или Memory Limit Exceeded (превышено ограничение по памяти). Код, который неэффективно использует ресурсы системы, отклоняется с данной ошибкой.
6. RE или Runtime Error (ошибка исполнения программы). Данная ошибка возникает, если в коде решения допущена ошибка (то есть он не может быть скомпилирован), либо в случае, если в ходе тестирования программа завершила работу с ошибкой.
7. MREJ или Manually Rejected (отклонено преподавателем). Данная ошибка возникает только в случае ручной перепроверки отправленного решения со стороны преподавателя.

Стоит больше внимания уделить вердикту Runtime Error. Студенты могут по различным причинам не знать, в чём заключается ошибка исполнения кода, написанного ими. К данному вердикту можно отнести большое количество ошибок. В ходе тестирования системы была выявлена следующая ошибка, которая встречалась на первых занятиях чаще всего: студенты,

40PDMN623
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

работавшие ранее с C++ на базе операционных систем Windows, использовали директиву «windows.h». Это приводило к ошибке компиляции программ, так как запуск кода (как и компоненты всей системы) осуществляется на базе Linux, а именно Debian последней на момент написании исследовательской работы версии.

Если ошибка возникает на этапе компиляции программы, то содержимое потока stderr (поток для вывода ошибок в программах) сохраняется, и студент может ознакомиться с ошибкой на странице отправленного решения (рис. 9). Но если ошибка возникает при тестировании программы (то есть компиляция прошла успешно), то она не будет выведена на странице результата решения. Сделано это было для того, чтобы студенты вручную проверяли свои программы на различных тестах, чтобы они учились не только писать программы, но и тестировать их.

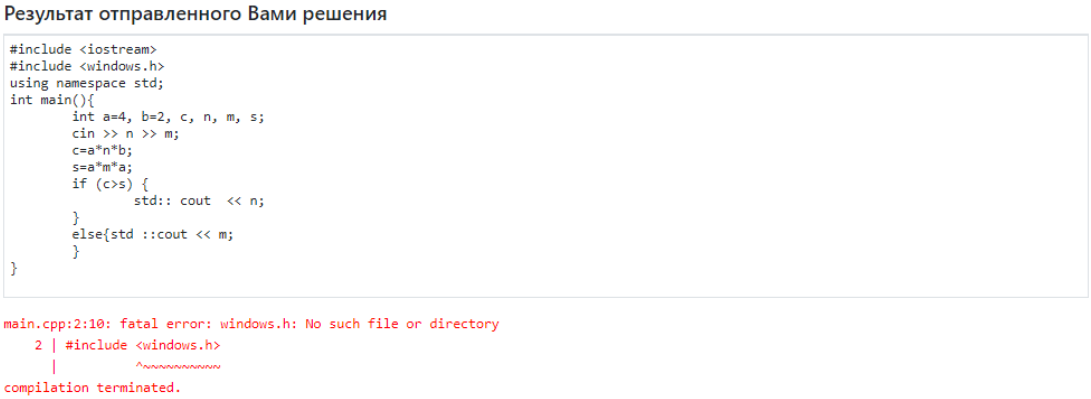
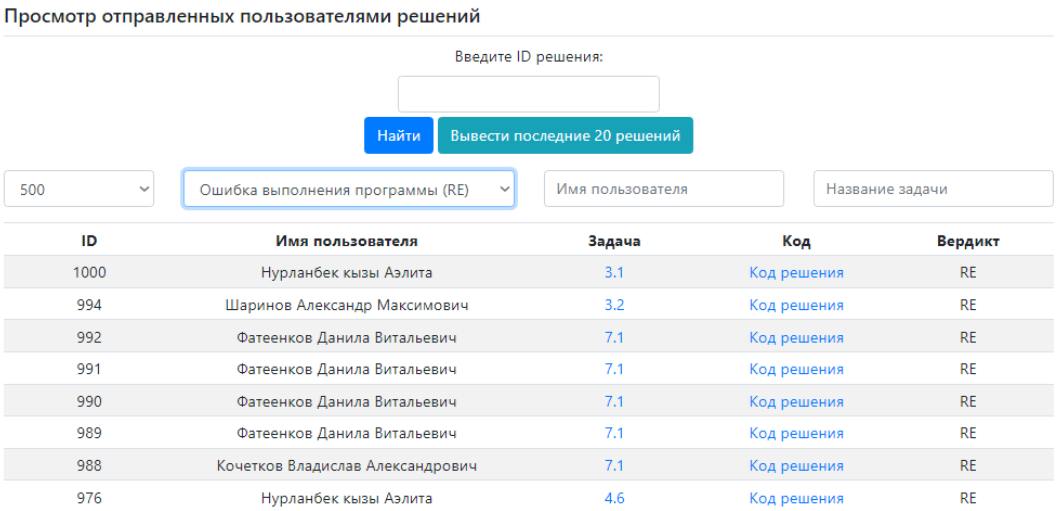


Рисунок 9. *Результат компиляции программы с использованием библиотеки «windows.h» (разработано авторами)*

Если преподаватель вручную отклоняет решение, то ему будет предложено написать причину отклонения, чтобы студент мог оперативно узнать — в чём проблема отправленного им решения.

Преподаватель может просматривать отправленные студентами решения. Для этого была создана отдельная страница, на которой выводится таблица с отправленными решениями. Таблица динамически обновляется и новые отправки выводятся в таблицу без обновления самой страницы (рис. 10).



ID	Имя пользователя	Задача	Код	Вердикт
1000	Нурланбек кызы Аэлита	3.1	Код решения	RE
994	Шаринов Александр Максимович	3.2	Код решения	RE
992	Фатеенков Данила Витальевич	7.1	Код решения	RE
991	Фатеенков Данила Витальевич	7.1	Код решения	RE
990	Фатеенков Данила Витальевич	7.1	Код решения	RE
989	Фатеенков Данила Витальевич	7.1	Код решения	RE
988	Кочетков Владислав Александрович	7.1	Код решения	RE
976	Нурланбек кызы Аэлита	4.6	Код решения	RE

Рисунок 10. *Страница просмотра отправленных решений (разработано авторами)*

В таблице представлена следующая информация:

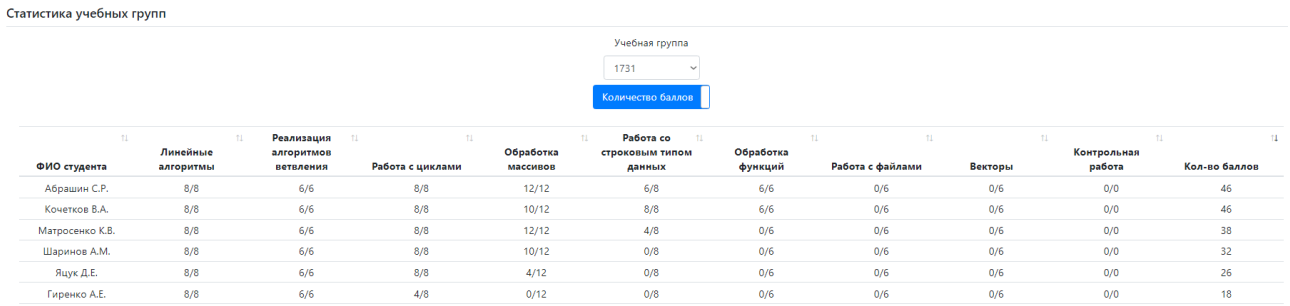
1. ID решения.
2. Автор отправки решения.
3. Ссылка на задачу.
4. Ссылка на страницу с кодом решения.
5. Результат проверки решения.

Также на странице есть поля для фильтрации решений:

1. Количество решений в таблице (20, 50, 100, 500).
2. Результат проверки (будут выведены только те решения, результат проверки которых совпадает с выбранным в выпадающем списке).
3. Имя пользователя (поиск по регулярному выражению, можно находить решения студента как по имени, так и по фамилии).
4. Название задачи (также поиск по регулярным выражениям).
5. ID решения.

Для сброса фильтров на странице присутствует кнопка «Вывести последние 20 решений».

Помимо просмотра решений, преподаватель может также получить статистику по всей группе студентов. Статистика представлена в виде таблицы, в которой выводятся построчно имена студентов, их успехи в каждом из разделов и общее количество набранных баллов (рис. 11). Данные в таблице можно сортировать по каждому из столбцов. Для реализации такой возможности использовался DataTable — плагин для работы с таблицами в jQuery.


Статистика учебных групп

Учебная группа

1731

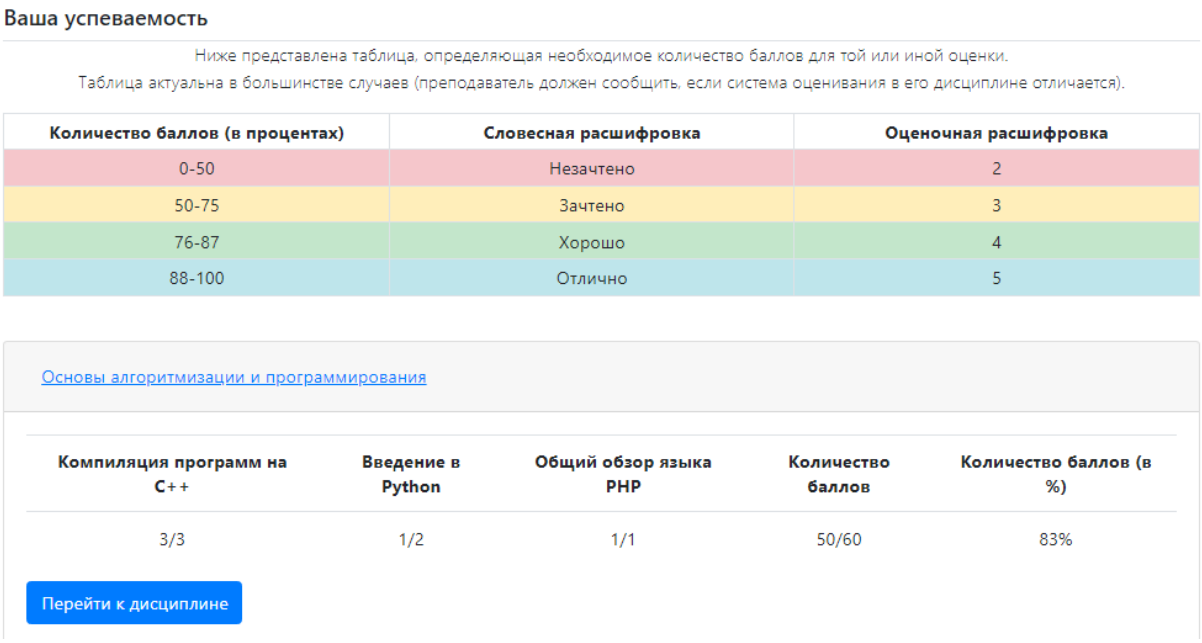
Количество баллов

ФИО студента	Линейные алгоритмы	Реализация алгоритмов ветвления	Работа с циклами	Обработка массивов	Работа со строковым типом данных	Обработка функций	Работа с файлами	Векторы	Контрольная работа	Кол-во баллов
Абрашин С.Р.	8/8	6/6	8/8	12/12	6/8	6/6	0/6	0/6	0/0	46
Кочетков В.А.	8/8	6/6	8/8	10/12	8/8	6/6	0/6	0/6	0/0	46
Матросенко К.В.	8/8	6/6	8/8	12/12	4/8	0/6	0/6	0/6	0/0	38
Шаринов А.М.	8/8	6/6	8/8	10/12	0/8	0/6	0/6	0/6	0/0	32
Яцук Д.Е.	8/8	6/6	8/8	4/12	0/8	0/6	0/6	0/6	0/0	26
Гиренко А.Е.	8/8	6/6	4/8	0/12	0/8	0/6	0/6	0/6	0/0	18

Рисунок 11. *Статистика обучения группы 1731 (разработано авторами)*

Также на странице предусмотрена возможность переключения между отображением баллов на каждый раздел и количеством задач для каждого раздела. В первом случае будут выводиться данные о набранных студентом баллах в каждом из разделов (первое значение в ячейке показывает количество набранных баллов, далее следует разделённое обратной косой чертой количество баллов, которое можно набрать, решая задачи из раздела), во втором будут выводиться количество решённых задач и общее количество задач в разделе.

Студент не может просматривать данную таблицу, но может ознакомиться с успеваемостью на отдельной странице. Студент может просматривать информацию об изучении разделов, на которые он был добавлен преподавателями. Статистика обучения также представлена таблицей, в которой присутствует только одна строка — количество решённых задач в каждом из разделов и общее количество набранных баллов в рамках выбранной дисциплины (рис. 12). Также на странице студент может ознакомиться с балльно-рейтинговой системой университета.



Ваша успеваемость

Ниже представлена таблица, определяющая необходимое количество баллов для той или иной оценки.
Таблица актуальна в большинстве случаев (преподаватель должен сообщить, если система оценивания в его дисциплине отличается).

Количество баллов (в процентах)	Словесная расшифровка	Оценочная расшифровка
0-50	Незачтено	2
50-75	Зачтено	3
76-87	Хорошо	4
88-100	Отлично	5

Основы алгоритмизации и программирования

Компиляция программ на C++	Введение в Python	Общий обзор языка PHP	Количество баллов	Количество баллов (в %)
3/3	1/2	1/1	50/60	83%

Перейти к дисциплине

Рисунок 12. *Страница просмотра успеваемости студента (разработано авторами)*

Для дополнительного контроля отправляемых студентами решений в систему был внедрён локальный антиплагиат. Алгоритм антиплагиата построен на определении расстояния Левенштейна. Данный параметр характеризует различие строк на основе нахождения минимального количества односимвольных преобразований (удаления, вставки или замены), необходимых, чтобы превратить одну строку в другую. То есть чем меньше расстояние Левенштейна среди двух строк, тем больше они похожи.

Перед проверкой решений, код сначала форматируется в строку путём удаления специальных символов и пробелов.

Сложность алгоритма составляет O(n*m), где n и m — длины строк.

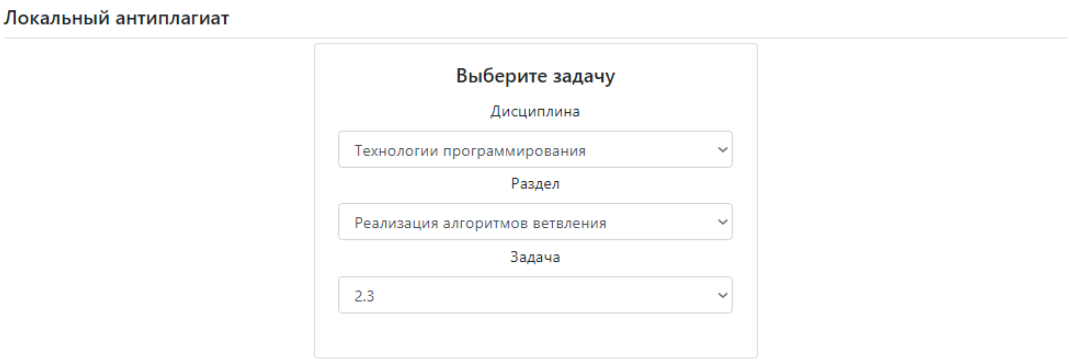
Антиплагиат проверяет только правильные решения (то есть прошедшие проверку и получившие вердикт OK). Все решения для выбранной задачи сохраняются в массив (ссылка на файл на сервере) и попарно проверяются. Также проверяется существует ли файл и не совпадают ли ID пользователей (в таком случае шаг проверки пропускается).

На странице, созданной для проверки решений на плагиат, можно выделить следующие элементы: форма для выбора дисциплины, раздела и задачи, а также таблица, в которой представлены результаты работы антиплагиата (рис. 13).

В таблице представлена следующая информация:

1. ID решения, которое было отправлено раньше.
2. ID решения, которое было отправлено позже.
3. Имя пользователя, который отправил решение раньше.
4. Имя пользователя, который отправил решение позже.
5. Процент совпадения двух решений.

Преподаватель может перейти на страницу решений (ID содержат ссылки на решения задач) и просмотреть коды лично (и отклонить решения, если такая необходимость возникла).



ID 1	ID 2	Имя студента (ранняя отправка)	Имя студента (поздняя отправка)	Процент копирования
494	530	Матросенко Кристина Витальевна	Кузьмин Игорь Сергеевич	98.41
494	604	Матросенко Кристина Витальевна	Кочетков Владислав Александрович	98.5
494	635	Матросенко Кристина Витальевна	Абрашин Степан Романович	99.36
494	636	Матросенко Кристина Витальевна	Дегтярёв Александр Игоревич	96.29
494	663	Матросенко Кристина Витальевна	Саяпин Никита Анатольевич	100
494	705	Матросенко Кристина Витальевна	Нурланбек кызы Аэлита	100

Рисунок 13. *Работа локального антиплагиата для одной из задач в системе (разработано авторами)*

Эксперимент

Апробация проводилась в течение весеннего семестра 2022–2023 учебного года и осеннего семестра 2023–2024 учебного года в Приамурском государственном университете имени Шолом-Алейхема и Иркутском национальном исследовательском техническом университете, в исследовании участвовало 42 и 57 студентов в контрольной и экспериментальной группах соответственно.

При отборе студентов для эксперимента учитывался уровень их подготовки, для участия были отобраны те, кто не участвовал в олимпиадах и не состоял в кружках по программированию, поскольку наличие этих студентов в выборке могло исказить данные. В период эксперимента студенты проходили в разработанной системе курсы программирования на языках C++ и Python.

Студенты экспериментальной группы проходили обучение с использованием автоматизированной системы проверки решений и с помощью системы поддержки через социальные сети, студенты контрольной группы использовали онлайн обучение с проверкой решений преподавателем и поддержкой через электронную почту. Обучение оценивалось по сто балльной системе, оценка проводилась в начале каждого учебного года до запуска обучения и в конце учебного года после окончания обучения. Все студенты согласно оцениванию, распределялись на три группы согласно уровню их подготовки, низкий (меньше 33 баллов), средний (34–66 баллов), высокий (больше 67 баллов) по результатам тестирования.

Результаты исследования представлены на рисунках 14 и 15 для контрольной и экспериментальной групп соответственно.

Результаты эксперимента показывают, что в экспериментальной группе улучшение составило 21 процент, в то время как в контрольной группе улучшение дало лишь 12 процентов. Кроме того, следует отметить значительное уменьшение количества студентов в низкой группе, что говорит о позитивном мотивационном аспекте при данном подходе к онлайн-обучению.

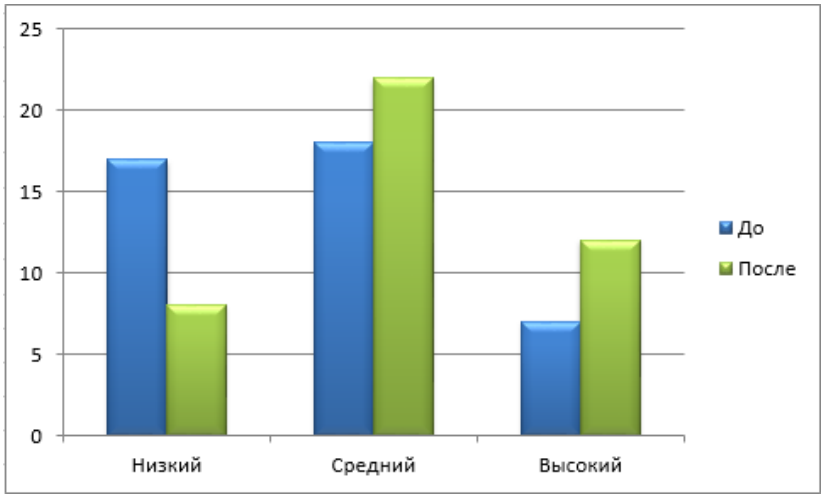


Рисунок 14. *Результаты исследования в контрольной группе (разработано авторами)*

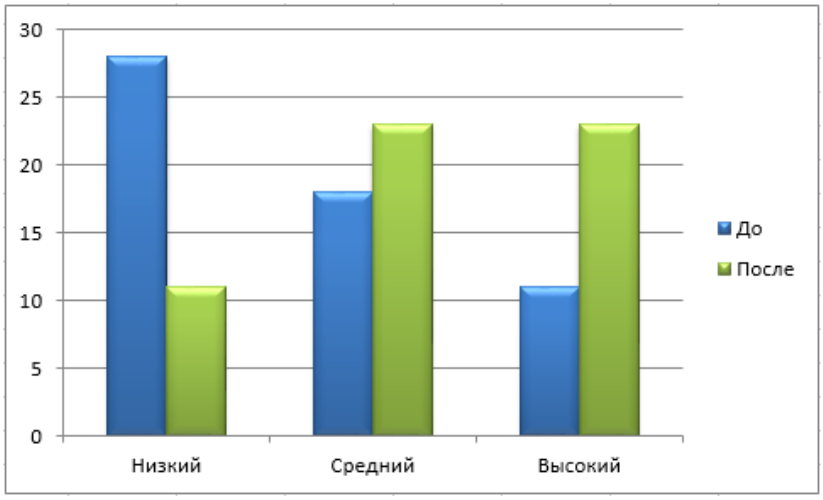


Рисунок 15. *Результаты исследования в экспериментальной группе (разработано авторами)*

Заключение

В рамках данной работы была рассмотрена информационная система, позволяющая во многих аспектах упростить и автоматизировать процесс обучения программированию. Система содержит в себе набор функциональных элементов, позволяющих в полной мере организовать обучение направлениям, связанных с программированием. Рассмотренная информационная система обладает таким свойством как масштабируемость, то есть предполагается, что данный проект может улучшаться и дополняться новым функциями. Дальнейшие улучшения могут включать в себя добавление функционала по работе с чекерами (код, проверяющий правильность выполнения задач без выходных данных, основываясь на работе некого «эталонного» решения, загруженного преподавателем) и генераторами (код, создающий тесты к задачам). Также важным аспектом работы данной системы является также дистанционное консультирование студентов по вопросам, которые могут возникнуть при изучении учебных материалов и решении практических задач. Эту часть программы также необходимо проработать в будущих обновлениях.

Разработанная информационная система на данный момент находится в процессе подтверждения свидетельства о государственной регистрации программы для ЭВМ.

ЛИТЕРАТУРА

1. Combéfis S., Beresnevičius G., Dagienė V. Learning programming through games and contests: overview, characterisation and discussion // Olympiads in Informatics. — 2016. — Т. 10. — № 1. — С. 39–60.
2. Горчаков Л.В., Карташов Д.В. Обучение программированию с использованием системы Ejudge // Вестник Томского государственного педагогического университета. — 2017. — № 9(186). — С. 109–112.
3. Михеев И.В. Применение автоматизированной системы проверки задач по программированию для студентов информационных направлений подготовки // Современные образовательные Web-технологии в системе школьной и профессиональной подготовки. — 2017. — С. 497–503.
4. Рогова Н.Н. Применение массовых открытых онлайн курсов для организации самостоятельной работы студентов // Балтийский гуманитарный журнал. — 2017. — Т. 6. — № 4(21). — С. 390–392.
5. Rabiman R., Nurtanto M., Kholifah N. Design and Development E-Learning System by Learning Management System (LMS) in Vocational Education // Online Submission. — 2020. — Т. 9. — № 1. — С. 1059–1063.
6. Ведерникова Т.И., Чепченко К.Б. Система проведения соревнований и проверки решений задач по программированию // Baikal Research Journal. — 2015. — Т. 6. — № 5. — С. 19.
7. Макиева З.Д. Проектирование автоматизированной системы проверки олимпиадных заданий по программированию // Известия Кыргызского государственного технического университета имени И. Раззакова. — 2016. — № 2. — С. 54–61.
8. Русакова М.С., Меерсон Р.И. Проектирование системы автоматизированной проверки решений для проведения соревнований по программированию // Вестник Самарского государственного университета. — 2013. — № 3(104). — С. 206–218.
9. Фролов Д.А., Балашов А.Д. Проектирование автоматизированной системы проверки олимпиадных заданий по программированию // Естественные и математические науки в современном мире. — 2015. — № 6(30). — С. 50–58.
10. Буянова И.В. Требования к системе онлайн-обучения студентов программированию и обзор существующих решений // Печатается по решению Редакционно-издательского совета ФГБОУ ВО «Хакасский государственный университет им. Н.Ф. Катанова». — 2021. — С. 109.
11. Буянова И.В., Замулин И.С. Выбор технологии защиты ресурса для онлайн-обучения программированию от недоверенного кода // В сборнике: Инженерные технологии: традиции, инновации, векторы развития. Материалы VIII Всероссийской научно-практической конференции с международным участие. Науч. и отв. редактор Д.Ю. Карандеев. Абакан. — 2022. — С. 122–124.
12. Данилова И.И., Полицын С.А. Применение автоматизированных тестов и инструментов статического анализа в информационной системе проверки программ в рамках обучения программированию / И.И. Данилова, С.А. Полицын // В сборнике: Информатика: проблемы, методология, технологии. Сборник материалов XIX международной научно-методической конференции. Под ред. Д.Н. Борисова. — 2019. — С. 921–926.

13. Кузовенков А.О. Совершенствование процесса автоматической проверки задач по программированию // В сборнике: Прикладная математика и информатика: современные исследования в области естественных и технических наук. Материалы VI Международной научно-практической конференции (школы-семинара) молодых ученых. — 2020. — С. 864–869.

14. Došilović H.Z., Mekterović I. Robust and scalable online code execution system // 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO). — IEEE, 2020. — С. 1627–1632.

15. Hafiz A., Jin K. Architecture of a Flexible and Cost-Effective Remote Code Execution Engine // arXiv preprint arXiv: 2105.01266. — 2021, URL: https://www.researchgate.net/publication/351342337_Architecture_of_a_Flexible_and_Cost-Effective_Remote_Code_Execution_Engine.

Luchaninov Dmitrii Vasilyevich
Sholom-Aleichem Priamursky State University, Birobidzhan, Russia
E-mail: dvluchano@mail.ru
RSCI: https://elibrary.ru/author_profile.asp?id=622813

Bazhenov Ruslan Ivanovich
Sholom-Aleichem Priamursky State University, Birobidzhan, Russia
E-mail: r-i-bazhenov@yandex.ru
RSCI: https://elibrary.ru/author_profile.asp?id=642728

Fateenkov Danila Vitalievich
Sholom-Aleichem Priamursky State University, Birobidzhan, Russia
E-mail: wiosna97@yandex.ru

Dorofeev Andrey Sergeevich
Irkutsk National Research Technical University, Irkutsk, Russia
E-mail: dorbaik2007@mail.ru
RSCI: https://elibrary.ru/author_profile.asp?id=189500

The system of automated verification of programming problems' solutions with the local anti-plagiarism function

Abstract. The purpose of this work is the research of the processes aimed at designing programming in a web environment for students of educational institutions of higher education using an information system that automates the verification of submitted code for practical tasks within the chosen discipline. The study examines frequent subtleties that arise in the practical application of solution verification systems and shows an implementation that can eliminate possible difficulties. The work also examines the main critical points that must be taken into account by the developer when developing the system. The main functional elements of the information system have been studied, such as: adding teaching materials, adding practical assignments, taking into account student progress, viewing solutions sent by students, checking solutions for borrowing (local anti-plagiarism). The principles of testing solutions sent by students are also described: possible results of testing solutions and problems that may arise at the stage of compiling and checking the functionality of the submitted code are considered. An experimental assessment of the effectiveness of using the developed system was carried out during the spring semester of the 2022–2023 academic year and the autumn semester of the 2023–2024 academic year at the Amur State University named after Sholom Aleichem and the Irkutsk National Research Technical University, 42 and 57 students participated in the study in control and experimental groups respectively. The research results show that the experimental group showed a 21 percent improvement, while the control group only showed a 12 percent improvement. The developed information system is currently in the process of confirming the certificate of state registration of a computer program.

Keywords: automated solution checking; student's independent work; design of electronic courses; programming training; automation of programming training; local anti-plagiarism