

## Разработка системы для организации обучения программированию в веб-среде

*Фатеенков Данила Витальевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### **Аннотация**

В статье рассмотрен процесс разработки системы, которая позволит автоматизировать процесс организации обучения программированию. Система должна позволить преподавателям загружать учебные материалы по дисциплинам, добавлять практические задачи, просматривать решения пользователей и выводить отчёты об успеваемости студентов. Система также должна автоматизировать процесс проверки отправленного пользователями к практическим задачам кода, так как вручную данный процесс может занимать слишком много времени.

**Ключевые слова:** HTML, JavaScript, PHP, Bootstrap, обучение программированию

## Development of a system for organizing programming training in a web environment

*Fateenkov Danila Vitalievich*

*Sholom-Aleichem Priamursky State University*

*Student*

### **Abstract**

The article considers the process of developing a system that will automate the process of organizing programming education. The system should allow teachers to upload course materials, add practice problems, view user solutions and display reports on students' progress. The system should also automate the process of checking the code sent by the users to the practical tasks, because manually this process can take too much time.

**Keywords:** HTML, JavaScript, PHP, Bootstrap, programming training

## **1. Введение**

### **1.1 Актуальность**

Информационные технологии в настоящее время стремительно развиваются, что может создавать проблемы в области обучения. Традиционные методы обучения программированию также стремительно теряют свою актуальность и люди всё чаще отдают предпочтение системам, которые автоматизируют процесс создания и изучения курсов программирования. Учебные учреждения переходят на собственные онлайн

разработки в области преподавания, в том числе преподавания программирования.

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. В настоящее время существует несколько ИС, предоставляющие инструменты для создания собственных обучающих курсов и позволяющих организовать учебный процесс. 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. Поэтому такая система необходима для адаптации под собственные нужды.

### **1.2 Обзор исследований**

И.В. Буянова и И.С. Замулин рассмотрели доступные технологии изоляции недоверенного кода от системы, в которой выполняется его тестирование [1]. В статье рассмотрены преимущества виртуализации с использованием ПО Docker и другие методы обеспечения безопасности при проверке кода.

И.И. Данилова и С.А. Полицын описали принципы применения автоматизированных тестов в информационной системе проверки программ в рамках обучения программированию [2].

А.О. Кузовенков выделил критерии для систем, которые автоматизируют процесс проверки кода [3]. В статье рассмотрены требования к таким системам, а также рассмотрены уже существующие решения, автоматизирующие данный процесс.

### **1.3 Цель исследования**

Цель – разработать систему, в которой можно будет организовать обучения студентов дисциплинам, связанным с программированием.

## **2. Материалы и методы**

Для реализации поставленной задачи использовались HTML, JavaScript, PHP, а также фреймворк Bootstrap.

## **3. Результаты и обсуждения**

В первую очередь необходимо определить тип разрабатываемой системы – веб-ориентированная система, в которой данные хранятся в БД и на сервере в виде файлов. Также система не предполагает наличие свободной регистрации в силу того, что она предназначена для внутреннего пользования в рамках учебного заведения. При этом пользователи смогут получить доступ к системе, не находясь на территории учреждения, но для этого им необходимо знать данные для входа.

База данных была построена с использованием СУБД MySQL и в ней могут содержаться следующие данные: информация о пользователях,

дисциплинах, учебных модулях, задачах и технической составляющей, без которой не получится наладить работу системы должным образом. В итоге получилось 11 таблиц.

Также был проведён анализ потоков данных, в ходе которого была построена логическая модель на основе методологии IDEF0 (см. рис. 1).



Рисунок 1. IDEF0 диаграмма главного процесса ИС

Из диаграммы можно выделить следующие данные:

1. Входные: список студентов (в формате таблицы); информация о дисциплине, разделе и задаче; информация об отправленном решении к задаче.

2. Результаты: отчёт об успеваемости группы студентов в рамках выбранной преподавателем дисциплины, отчёты о работе системы, отчёты о результате обучения студента в рамках выбранной дисциплины.

3. Организация обучения осуществляется на основании следующих документов: методические указания, аннотации к рабочей программе дисциплины, описание образовательной программы по дисциплине.

4. Деятельность в системе осуществляют: преподаватели, студенты, а также программное обеспечение.

Также была проведена декомпозиция первого уровня для уточнения работы главного процесса (см. рис. 2).

Можно выделить следующие процессы на первом уровне декомпозиции:

1. Добавление студентов в дисциплину: формируется группа студентов на основе загруженных списков преподавателем. Эту группу в дальнейшем можно привязать к необходимой дисциплине.

2. Добавление материалов курса: преподавателем создаётся дисциплина, добавляются необходимые материалы для изучения и задачи для самостоятельной подготовки по пройденной лекции.

3. Организация и проведение обучающего процесса: студенты изучают добавленные материалы и выполняют задания.

4. Печать отчётов: преподаватель может получить отчёт об успеваемости группы в рамках изучаемой дисциплины.

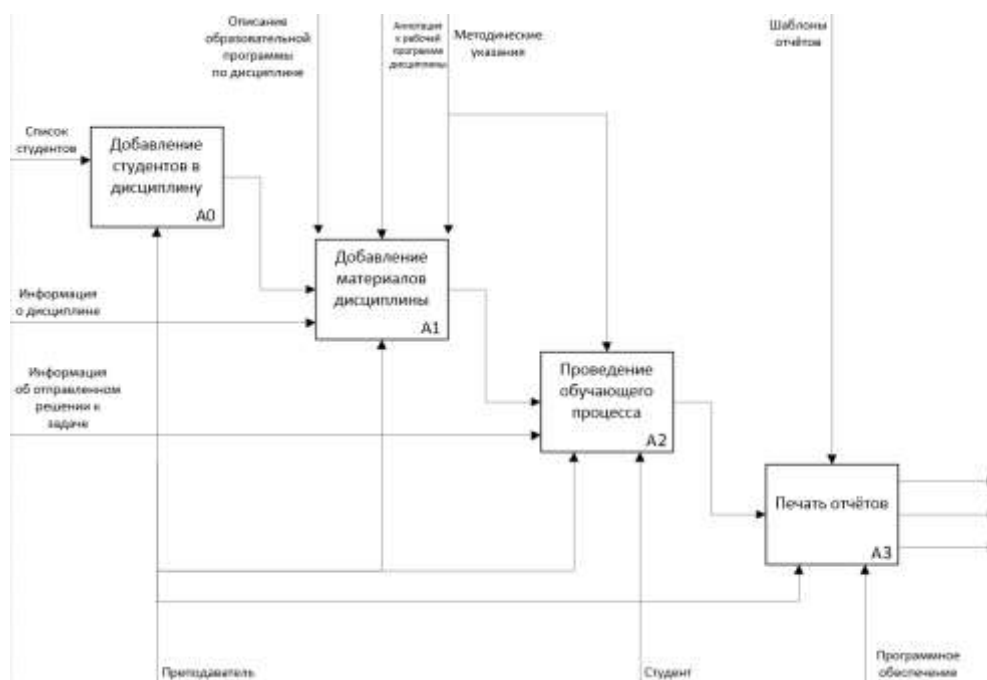


Рисунок 2. Декомпозиция первого уровня главного процесса ИС

Есть много различных методологий, по которым можно провести анализ разрабатываемой ИС и построить будущую архитектуру, но IDEF0 является самой распространённой в настоящее время.

На главной странице представлены следующие элементы: панель навигации (см. рис. 3), ссылки для скачивания IDE используемых языков программирования в системе, а также приветствующая пользователя информация.



Рисунок 3. Панель навигации в системе

Панель навигации позволяет получить доступ к следующим страницам сайта:

1. Дисциплины – на странице представлен список доступных пользователю дисциплин (см. рис. 4).
2. Успеваемость – на странице представлена статистика о прохождении дисциплин, а также памятка о выставлении оценки за прохождение дисциплины на основе полученных баллов.
3. Работа в системе – документация по работе в системе.
4. Администрирование (не доступно для студентов) – содержит список ссылок, ведущих на страницы по работе с содержимым ИС.

На странице с дисциплинами пользователю представлен список доступных для прохождения учебных курсов. Список выполнен в виде

таблицы, элементами которой являются дисциплины. Элементы представлены в виде прямоугольных блоков, которые разделены на 3 части: название учебного курса, краткое описание (опционально при создании) и кнопка для перехода к материалам курса.

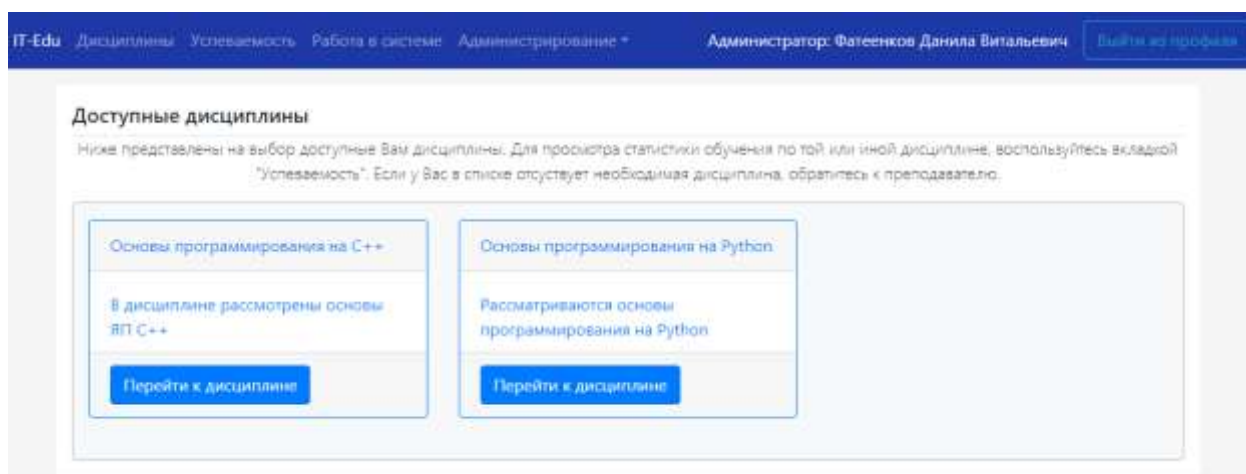


Рисунок 4. Страница с доступными дисциплинами

На странице с успеваемостью пользователь может узнать количество полученных баллов в рамках учебного курса, который он изучает. Количество баллов также представлено в процентном соотношении (сделано это для того, чтобы установить соответствие университетской рейтинговой системе).

Дисциплины на данной странице представлены в виде раскрывающихся элементов. Внутри таких элементов хранится информация о всех разделах дисциплины, количестве задач для каждого раздела и количестве решённых задач для каждого раздела, а также количество баллов. Под таблицей расположена кнопка, которая осуществляет переход на страницу материалов дисциплины (см. рис. 5).

**Ваша успеваемость**

Нижче 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**

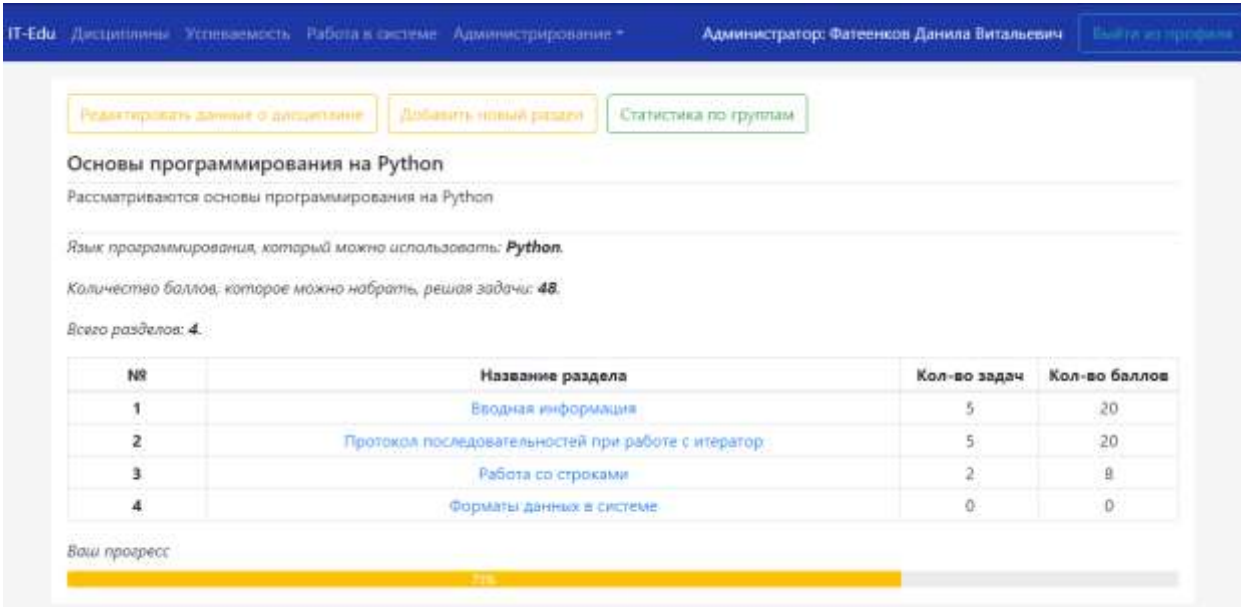
Компиляция программ на C++	Введение в Python	Общий обзор языка PHP	Количество баллов	Количество баллов (в %)
3/3	1/2	0/1	40/60	67%

[Перейти к дисциплине](#)

Рисунок 5. Страница с успеваемостью студента

Перейдя на страницу дисциплины (см. рис. 6), пользователь может ознакомиться со следующей информацией: краткое описание, используемый в рамках дисциплины ЯП, максимальное количество баллов за решение задач и список дисциплин, по которым можно перейти и получить доступ к учебным материалам и задачам. Также представлена шкала прогресса, которая заполняется по мере выполнения заданий в рамках дисциплины.

Для преподавателей также доступны элементы управления дисциплиной: кнопка редактирования данных о дисциплине, кнопка добавления новых учебных модулей и кнопка для формирования статистики об успеваемости группы студентов.



IT-Edu Дисциплины Успеваемость Работа в системе Администрирование + Администратор: Фатеенков Данила Витальевич Войти из профиля

Редактировать данные о дисциплине Добавить новый раздел Статистика по группам

### Основы программирования на Python

Рассматриваются основы программирования на Python

Язык программирования, который можно использовать: **Python**.

Количество баллов, которое можно набрать, решая задачи: **48**.

Всего разделов: **4**.

№	Название раздела	Кол-во задач	Кол-во баллов
1	Вводная информация	5	20
2	Протокол последовательностей при работе с итератор	5	20
3	Работа со строками	2	8
4	Форматы данных в системе	0	0

Ваш прогресс

75%

Рисунок 6. Страница дисциплин в ИС

На странице с учебными материалами представлена информация, которую предварительно загрузил преподаватель (см. рис. 7). Информация может быть представлена в различных форматах: текст, изображения, таблицы, видео и аудио.

В самом верху страницы (но под панелью навигации) представлены элементы управления учебным разделом: редактирование содержимого и добавление задач. Кнопка «Добавить задачу» ведёт преподавателя на новую страницу, в которой представлена форма для добавления данных о практических заданиях.

После учебных материалов представлен список задач (см. рис. 8).

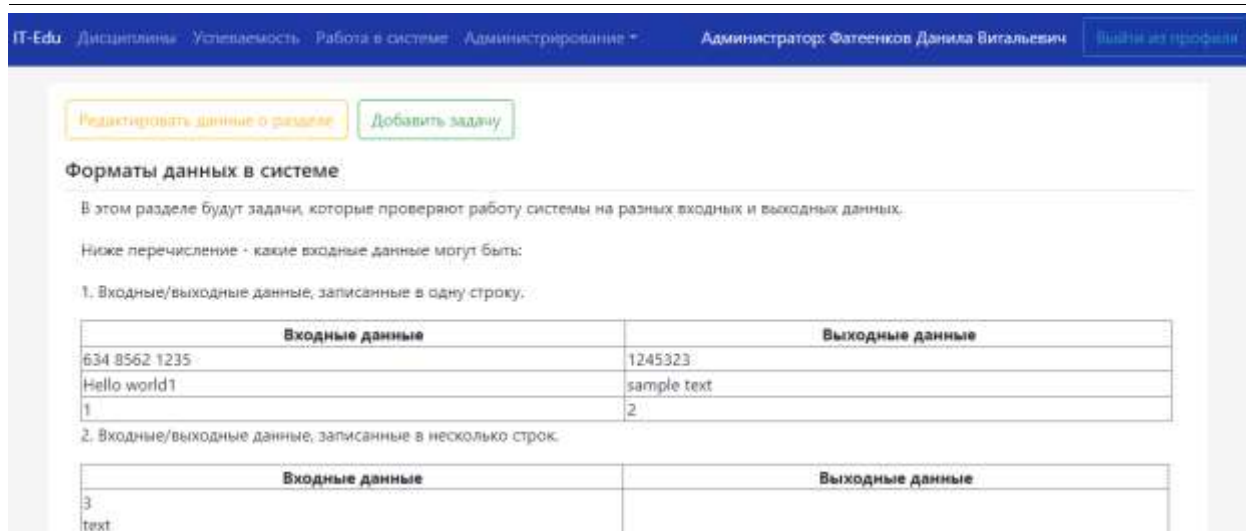


Рисунок 7. Страница с учебными материалами в рамках выбранной дисциплины

Список задач после учебных материалов содержит следующую информацию, которая будет полезной для пользователя: название задания (и ссылка ведущая на страницу с задачей), количество баллов за правильное решение и ссылка на последнее отправленное пользователем решение.

Строка с задачей в списке окрашивается в зелёный цвет, если пользователь правильно её решил. Это сделано для того, чтобы облегчить навигацию пользователя в системе.

Задачи для самостоятельной подготовки:

Название задачи	Последнее решение	Количество баллов
Глухой телефон	Код	10
Монетки	-	10

Рисунок 8. Список задач в рамках выбранного учебного раздела

На странице выбранной задачи представлено загруженное преподавателем условие к заданию, тестовые данные (используемый формат на входе и выходе). Условие преподаватель загружает также через специальную форму.

Также слева от блока с условием представлен список всех задач, добавленных в учебный раздел. Сделано это для облегчения навигации пользователя в системе. Через этот список также можно вернуться к учебным материалам выбранного раздела.

Также для преподавателей вынесены следующие элементы управления: кнопка для редактирования задачи и кнопка, ведущая пользователя на страницу со списком всех отправленных пользователями (в том числе и преподавателями) решений (см. рис. 9).

В условии к заданию также указываются ограничения по времени и памяти, которые также задаёт преподаватель.

IT-Edu Дисциплины Успеваемость Работа в системе Администрирование \* Администратор: Фатеев Данила Витальевич Выход из профиля

Расшифровка строки

Симпатичный узор

Пустая задача

**Перевернуть и сложить**

Код к задаче

Редактировать задачу

Статистика отправок по данной задаче

### Перевернуть и сложить

Операция «Перевернуть и сложить» (англ. Reverse-and-add) заключается в том, чтобы сложить исходное число с его «перевернутой» копией, то есть числом, цифры которого записаны в обратном порядке. Например,  $56 + 65 = 121$ ,  $521 + 125 = 646$ . Эта операция может быть применена к любому натуральному числу.

Ваша задача заключается в том, чтобы посчитать количество таких операций для числа  $n$ , которые нужно совершить, чтобы получить палиндром.

**Входные данные**

Одно число -  $n$  ( $n > 100$ ).

**Выходные данные**

Одно число - количество операций "перевернуть" и "сложить", которое нужно провести, чтобы получить палиндром.

Ограничение по времени: 1 с.

Ограничение по памяти: 16 МБ.

Рисунок 9. Страница с информацией о выбранной задаче

После условия задачи на странице выводится следующий важный элемент: форма отправки решения к задаче. В зависимости от выбранного языка при создании (его можно оставить на усмотрение студента, либо выбрать конкретный для данной дисциплины) форма может варьироваться (см. рис. 10).

Также под формой выводится таблица отправок решений, в которой представлена следующая информация: номер попытки (номер строки в таблице), ID попытки (нужен для упрощения поиска решения, если есть необходимость), время отправки решения, результат проверки и ссылка на страницу с кодом решения.

Язык программирования

C++

Отправить решение

Отправить

#### Результаты отправок

Номер попытки	ID попытки	Время отправки решения	Результат	Код решения
3	106	16.06.23.13:46:18	Принято	<a href="#">Код</a>
2	105	16.06.23.13:46:03	Ошибка компиляции	<a href="#">Код</a>

Рисунок 10. Форма отправки решения к задаче

Перейдя на страницу с кодом отправленного решения, пользователь может получить следующую информацию (см. рис. 11): отправленный пользователем код, ID решения, ссылка на задачу, а также результат отправки.

Результат отправки преподаватель может изменить, если в этом есть необходимость.



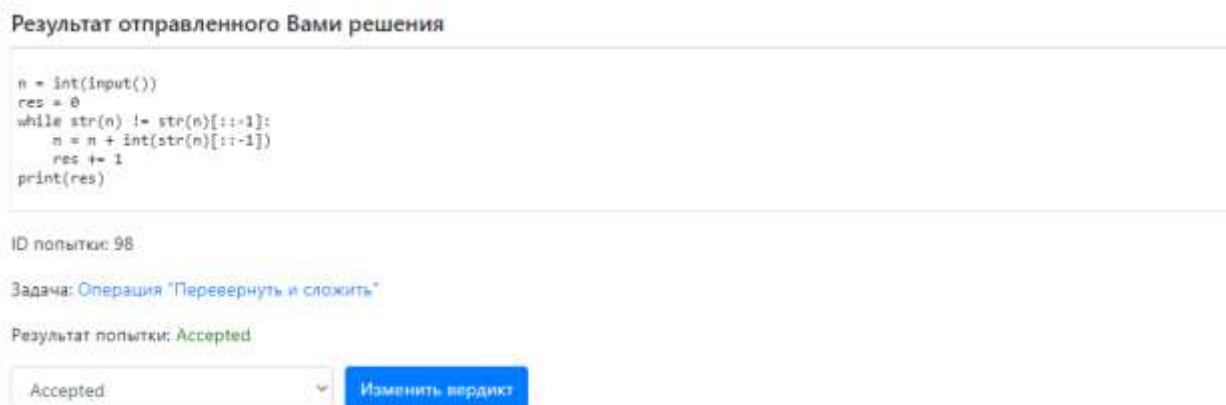


Рисунок 11. Страница просмотра отправленных решений

Весь описанный функционал доступен всем пользователям, включая студентов. Но есть страницы, которые могут просматривать только преподаватели и администраторы. В основном это страницы по управлению содержимым ИС.

Первым этапом в организации обучения это добавление дисциплины. Сделать это можно с помощью специальной формы (см. рис. 12). Преподаватель должен указать название дисциплины, выбрать язык программирования и вписать краткое описание дисциплины (это опционально).

Также под формой добавления новых дисциплин выводится таблица с уже созданными дисциплинами и следующими элементами управления: кнопка для редактирования созданной дисциплины, кнопка для управления доступом к учебным материалам (запрещает пользователям изучать дисциплину, но при этом не удаляет её) и кнопка для удаления дисциплины.

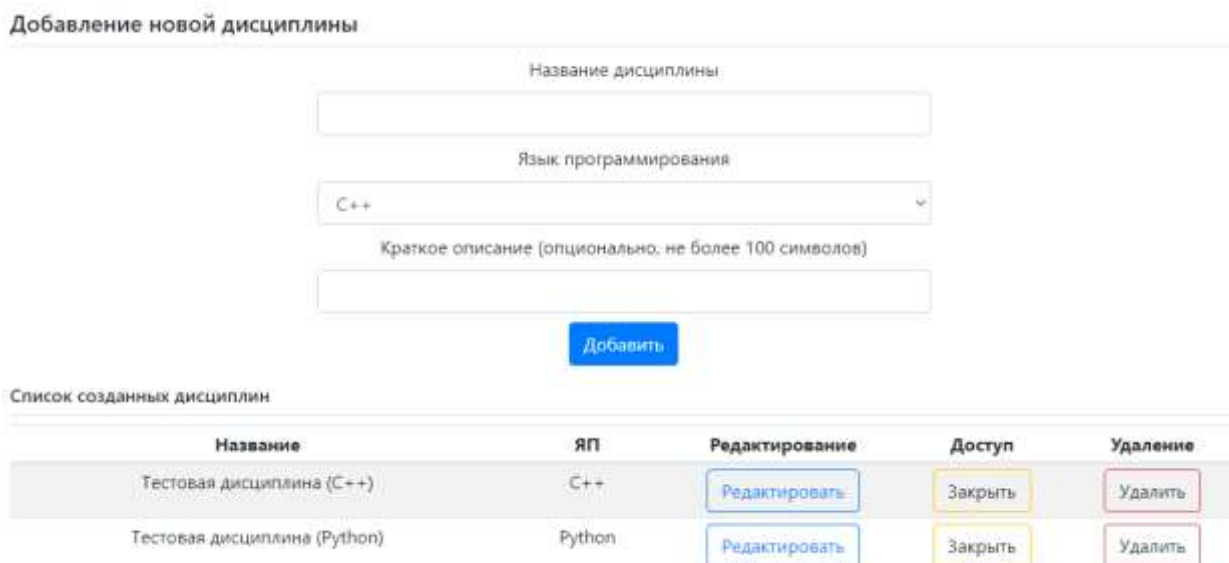


Рисунок 12. Страница добавления новых дисциплин

Форма для добавления учебных материалов состоит из двух элементов: строка для ввода названия раздела и ввода содержимого раздела (см. рис. 13).

Поле для ввода содержимого выполнено по технологии WYSIWYG (“What You See Is What You Get”), то есть в этом поле отображается содержимое в том виде, в котором оно будет представлено на странице учебного раздела. Поддерживается форматирование текста, добавление изображений, видео, ссылок и таблиц.

Добавление нового раздела для дисциплины Основы программирования на Python

Название раздела

Содержимое раздела

Файл Редактировать Вид Вставить Формат Инструменты Таблица

↶ ↷ Абзац ▾ **B** *I*

СЛОВ: 0 ПРИ ПОДДЕРЖКЕ TINY

Добавить

Рисунок 13. Форма добавления учебного раздела для дисциплины

В форме для добавления задач представлены следующие элементы: строка для названия задачи, поле для содержимого условия задачи, а также поля для ввода количества баллов за правильное решение, ограничения по времени и памяти. Также преподаватель может внести тестовые данные – для этого ему нужно указать, что поступает программе на вход и что должно быть на выходе (см. рис. 14).

Добавление задачи

Название задачи

Содержимое задачи

Файл Редактировать Вид Вставить Формат

↶ ↷ **B** *I*

СЛОВ: 0 ПРИ ПОДДЕРЖКЕ TINY

Количество баллов

Ограничения по времени

Ограничения по памяти

Входные данные

Тестовые значения

Выходные данные

Добавить

Добавить тестовые значения

Рисунок 14. Форма для добавления задач в выбранный раздел

Преподаватели могут просматривать отправленные пользователями решения. Для этого в системе присутствует специальная страница, где можно просмотреть последние 20 отправленных пользователями решений или найти нужное решение по ID (см. рис. 15). Важно также отметить, что преподаватели могут просматривать решения всех пользователей в системе, но студенты не могут смотреть решения друг друга.

Также преподаватель может просмотреть решения в рамках выбранной им задачи. Для этого на странице с условием задачи необходимо перейти в список решений и будут выбраны только те, которые относятся к открытой задаче.

#### Просмотр отправленных пользователями решений

Введите ID решения:

Найти
Вывести последние 20 решений

Номер отправки	Имя пользователя	Код	Вердикт
80	Александр Александрович	Код решения	Accepted
79	Александр Александрович	Код решения	Runtime Error / Compilation Error
78	Александр Александрович	Код решения	Memory Limit
77	Александр Александрович	Код решения	Memory Limit
76	Александр Александрович	Код решения	Runtime Error / Compilation Error
75	Александр Александрович	Код решения	Runtime Error / Compilation Error
74	Александр Александрович	Код решения	Runtime Error / Compilation Error
73	Александр Александрович	Код решения	Accepted
72	Александр Александрович	Код решения	Runtime Error / Compilation Error
71	Александр Александрович	Код решения	Accepted

Рисунок 15. Страница просмотра решений

Также преподаватели могут просматривать статистику по изучению дисциплины среди студентов в рамках интересующей группы. После выбора группы выводится таблица, в которой перечислены все студенты группы и все разделы учебной дисциплины. В ячейках таблицы представлено количество решённых студентом задач и общее количество задач в разделе. В конце таблицы выводится набранной количество баллов студентом (см. рис. 16).

#### Статистика группы 9999t

ФИО студента	Основные концепции Python	Простые сортировки	Сортировки, не основанные на сравнениях	Алгоритмы поиска	Строковые алгоритмы	Баллы
Александр А.А.	1/1	2/2	1/2	2/2	0/1	44
Владим В.В.	1/1	2/2	2/2	2/2	1/1	50
Семён С.С.	1/1	1/2	1/2	0/2	0/1	21
Евгений Е.Е.	0/1	0/2	0/2	0/2	0/1	0

Рисунок 16. Статистика успеваемости группы в рамках выбранной дисциплины

Преподаватель также может сохранить отчёт об успеваемости студентов для дальнейшей печати. Сохраняется таблица, как и на сайте, но добавляется дополнительная информация о дисциплине: название, язык программирования и т.д.

Также преподаватель может загружать новых пользователей в систему. Для этого создана отдельная форма, в которой можно загрузить файл со списком студентов. После загрузки списка, формируются логины и пароли для студентов и выводится таблица с данными для входа (можно также распечатать).

Преподаватель, после создания дисциплины и добавления учебной группы, должен установить связь между группой и дисциплиной. Для этого есть отдельная форма (см. рис. 17). Добавление связи необходимо для того, чтобы открыть доступ к дисциплине студентам. При этом достигается разграничение доступа и учебная группа, которая не связана с дисциплиной, не сможет получить доступ к учебным материалам.

#### Добавление связи между группами и дисциплинами

Дисциплина	Номер группы	Удаление связи
Тестовая дисциплина (C++)	1791a	Удалить
Тестовая дисциплина (Python)	1791a	Удалить
Тестовая дисциплина (Python)	9999t	Удалить

Рисунок 17. Добавление связи между группами и дисциплинами

Также важным элементом подобной системы является локальный антиплагиат. Данный функционал необходим, чтобы студенты не копировали решения у друг друга. Была добавлена страница, на которой преподаватель может отслеживать скопированные решения и при необходимости обнулять результат. Важно отметить, что учитываются только правильные решения (сделано это для того, чтобы не было сильной нагрузки на систему во время проверки решений).

На странице выводится следующая информация, если обнаружен плагиат: пользователь, кто решил задачу первым и пользователь, кто решил задачу вторым. Также выводится код обоих решений и предложение обнулить результат решений (см. рис. 18). Используется ступенчатый метод проверки исходного кода на оригинал.

## Локальный антиплагиат

Обнаружен возможный плагиат в задаче: Алгоритм сортировки слиянием

Внимание: 2 решения от разных пользователей совпали!

Первым задачу решил: [Иванов Иван Александрович](#)

Второе решение отправлено пользователем: [Фатеенков Данила Витальевич](#)

[Обнулить результат решения](#)

<a href="#">Иванов Иван Александрович</a>	<a href="#">Фатеенков Данила Витальевич</a>
<a href="#">Код решения</a>	<a href="#">Код решения</a>
<pre>#include &lt;iostream&gt; #include &lt;vector&gt; using namespace std; //Сортировка слиянием vector&lt;int&gt; merge_array(vector&lt;int&gt;&amp; arr1, vector&lt;int&gt;&amp; arr2) {     int i=0,j=0;     int n=arr1.size(),m=arr2.size();     vector&lt;int&gt; new_arr;</pre>	<pre>#include &lt;iostream&gt; #include &lt;vector&gt; using namespace std; vector&lt;int&gt; merge_array(vector&lt;int&gt;&amp; arr1, vector&lt;int&gt;&amp; arr2) {     int i=0,j=0;     int n=arr1.size(),m=arr2.size();     vector&lt;int&gt; new_arr;</pre>

Рисунок 18. Пример работы локального антиплагиата в ИС

Таким образом была разработана система, позволяющая организовать обучение программированию с использованием веб-технологий. Система проста и удобна в использовании, а также обладает всем необходимым функционалом для обеспечения работы системы.

### Библиографический список

1. 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 участием. Науч. и отв. редактор Д.Ю. Карандеев. Абакан. 2022. С. 122-124.
2. Данилова И.И., Полицын С.А. Применение автоматизированных тестов и инструментов статического анализа в информационной системе проверки программ в рамках обучения программированию // В сборнике: Информатика: проблемы, методология, технологии. Сборник материалов XIX международной научно-методической конференции. Под ред. Д.Н. Борисова. 2019. С. 921-926.
3. 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.