

Обновление и рефакторинг кода информационной системы обучения программированию в ПГУ им. Шолом-Алейхема

Фатеенков Данила Витальевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье представлены результаты обновления сайта ITProgEdu, основной задачей которого является организация обучения программированию студентов высшего образования. Рассматриваются методы, модули и языки программирования, которые были использованы для обновления Frontend-составляющей информационной системы. Также описан процесс рефакторинга программного кода ИС на языке программирования PHP.

Ключевые слова: обновление информационной системы, рефакторинг, PHP, JavaScript, AJAX, Bootstrap

Updating and refactoring the code of the information system of programming training at Sholem Aleichem Priamursky State University

Fateenkov Danila Vitalievich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article presents the results of updating the ITProgEdu site, the main task of which is to organize programming training for students of higher education. The methods, modules and programming languages that were used to update the Frontend component of the information system are considered. The process of refactoring the IS program code in the PHP programming language is also described.

Keywords: information system update, refactoring, PHP, JavaScript, AJAX, Bootstrap

1. Введение

1.1 Актуальность

Жизненный цикл ПО зачастую предполагает после внедрения ИС в организацию с последующим началом работы пользователей в приложении также и сопровождение программы. Это значит, что разработчики продолжают дорабатывать своё ПО с целью улучшения пользовательского опыта или с целью исправления возникающих ошибок и уязвимых мест в коде, которые создают риски возникновения разного рода атак на ИС.

Этап сопровождения также важен, как и тестирование программного обеспечения как раз за счёт возможности доработки недочётов и внедрения новых особенностей в уже готовую ИС, что позволяет относительно быстро реагировать на отзывы пользователей ПО. Зачастую в этап сопровождения включают рефакторинг legacy-кода и доработку Frontend-составляющей ИС, что представляет собой очень важный этап в жизненном цикле любого ПО.

1.2 Обзор исследований

А.С. Нышанова и З.Т. Маликова в своей работе детально рассмотрели CSS фреймворк Bootstrap, который широко применяется в настоящее время в разработке веб-ориентированных информационных систем [1].

Ч.М. Касымбекова и Ж.К. Матисаков также рассмотрели фреймворк Bootstrap и описали его преимущества использования, а также привели примеры существующих ИС, работающих с использованием данного фреймворка [2].

А.К. Балаев и И.Б. Государев в своей работе рассмотрели проблему некорректной реализации рефакторинга, в процессе которого информационная система начинает вести себя некорректно [3]. Авторами определены базовые пункты, необходимые до начала рефакторинга, для осуществления безрискового изменения кода.

В.Ю. Кириллов и А.И. Костюхин в своей работе проанализировали целесообразность применения методологии разработки через тестирование для выполнения рефакторинга MVC приложения [4]. В ходе работы определена сущность методологии, основные понятия, а также на конкретном примере рассмотрены этапы и задачи разработчика.

Д.С. Дерюгин рассмотрел рефакторинг как явление в разработке и дал определение данному термину, а также цели и задачи, в которых необходимо проводить рефакторинг кода [5].

А.С. Басов рассмотрел необходимость проведения рефакторинга программного кода, а также методы и основные принципы проведения рефакторинга [6].

1.3 Цель исследования

Цель – обновить визуальное оформление информационной системы обучения программированию ITProgEdu и провести рефакторинг функциональной части системы.

2 Материалы и методы

Для решения поставленных задач применяется язык программирования PHP, а также JavaScript и фреймворк для CSS Bootstrap 5.

3 Результаты и обсуждения

Работу стоит начать с определения «рефакторинг», которое будет основной темой статьи на протяжении всего текста.

Рефакторинг – это процесс изменения внутренней структуры программного кода без изменения его внешнего поведения. Основная цель рефакторинга – улучшение читаемости, поддерживаемости и расширяемости кода, а также устранение дублирования, упрощение логики и повышение производительности разработки.

Можно выделить наиболее простые и распространённые примеры рефакторинга кода:

1. Переименование переменных, функций или классов для лучшего понимания их назначения.
2. Разделение больших функций на несколько более мелких и специализированных.
3. Применение шаблонов проектирования для улучшения архитектуры.
4. Устранение вложенных условных конструкций через использование guard clauses или полиморфизма.

Примеров, в действительности, намного больше и в статье рассматривается только часть, так как специфика проведения рефакторинга сильно связана с контекстом, в рамках которого возникает необходимость проведения рефакторинга. То есть в зависимости от архитектуры ИС могут быть рассмотрены различные задачи обновления программного кода.

На базе Приамурского государственного университета им. Шолом-Алейхема с 2023-го года работает информационная система по организации обучения программированию студентов высшего звена. Информационная система представлена в виде веб-ориентированного приложения (сайта) и написано с использованием языка программирования PHP, а также JavaScript. Информация о студентах, успеваемости, учебных группах и других объектах и сущностях хранится в базе данных, работающей под управлением СУБД MySQL последней, на момент написания статьи, версии.

Из-за сжатых сроков разработки в архитектуре ИС был допущен ряд ошибок не критического характера, но требующие доработки. Данные ошибки не влияли на работу ИС и в подавляющем большинстве случаев сайт работал стабильно, предоставляя студентам возможность изучения учебных дисциплин круглосуточно.

Помимо необходимости проведения рефакторинга программного кода была также поставлена задача обновления UI-содержимого сайта. Первая версия сайта работала с использованием фреймворка Bootstrap 4-ой версии, которая хоть и не является устаревшей, но на данный момент уступает последней (пятой) версии по удобству и количеству улучшений.

Обновление внешнего вида сайта было также направлено и на повышение удобства работы с информационной системой за счёт уменьшения количества кликов, которые должен совершить пользователь для получения необходимой информации.

На ранних этапах решения поставленной задачи была пересмотрена структура ИС и выбраны дополнительные средства для проведения рефакторинга и обновления внешнего вида сайта:

1. Одним из главных решений был выбор шаблонизатора вместо вывода содержимого страницы через PHP. Данное решение позволило отделить HTML-разметку от PHP-кода и упростить дальнейшую разработку.

2. Была обновлена версия PHP на сервере для получения доступа к актуальным модулям и функциям для дальнейшей разработки.

3. Для работы с БД была внедрена ORM RedBeanPHP, предоставляющая расширенный функционал для обработки запросов и выполнения операций над базой данных ИС.

4. Был осуществлён переход на актуальную версию Bootstrap для возможности использования новых элементов в HTML-разметке, а также для достижения большей эффективности в работе сайта.

Обновление frontend составляющей сайта проходило постепенно, и работа над страницами велась в определённом порядке. В первую очередь были изменены страница входа и главная страница информационной системы (см. рис. 2). Старый вариант предлагал достаточно функциональности для начала работы, но не был удобен для использования за счёт необходимости получения списка дисциплин переходом на отдельную страницу (см. рис. 1).

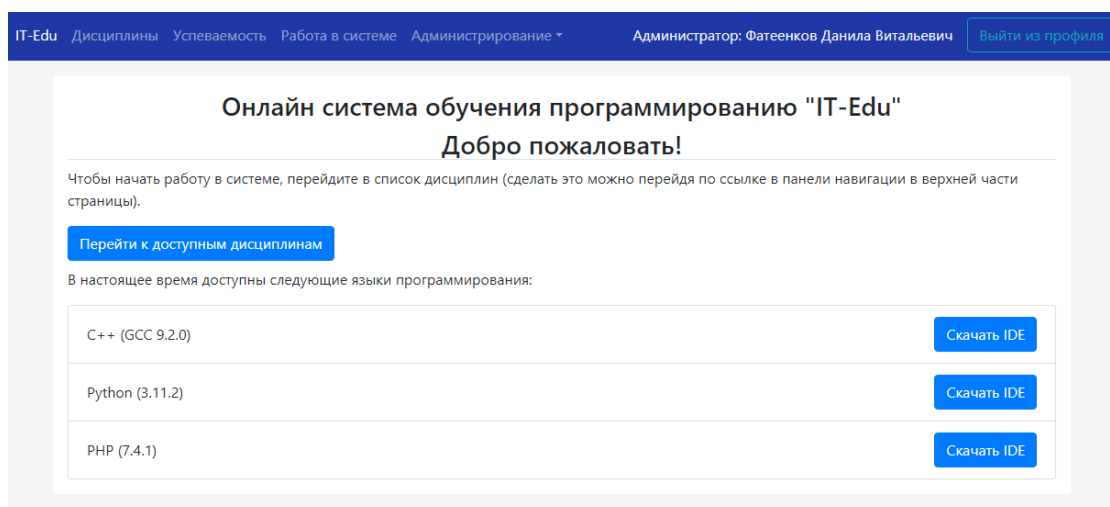


Рисунок 1. Старый вариант главной страницы информационной системы

В новой версии было принято решение убрать страницу со списком доступных дисциплин с последующим выводом всех необходимых элементов на главную страницу.

Дисциплины представлены в виде стандартного списка, где каждый элемент подсвечен одним из двух цветов: светло-голубым, если задачи в рамках дисциплины доступны для решения и красным, если преподаватель закрыл отправку решений в рамках выбранной дисциплины. В списке указаны: название дисциплины, язык программирования и количество учебных разделов.

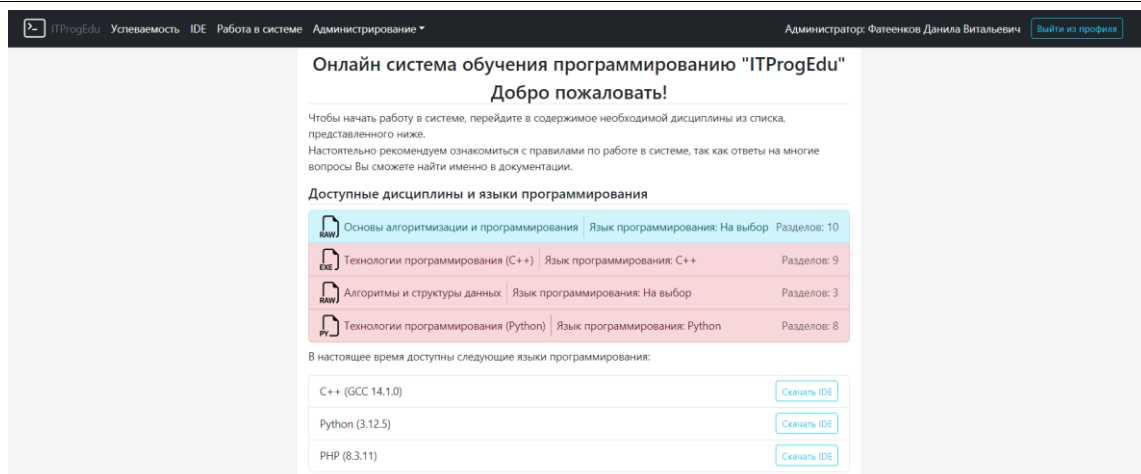


Рисунок 2. Новый вариант главной страницы информационной системы

Также на главную страницу был добавлен блок со статистикой деятельности пользователя на сайте. В этом блоке студент может получить краткую информацию о результатах всех отправленных им решений, а также ссылки на последние 5 отправок для возможности быстрого возвращения к работе (см. рис. 3).

Ваша активность на сайте

Статистика отправок		Последние 5 решений		
Вердикт	Кол-во	ID	Задача	Вердикт
Accepted	165	4526	Сложение	RE
Wrong Answer	54	4525	Сложение	OK
Time Limit Exceeded	7	4524	Сложение	RE
Memory Limit Exceeded	5	4523	Сложение	OK
Runtime Error	66	4520	1.1	OK

Рисунок 3. Блок с активностью пользователя на главной странице сайта

Многие страницы сайта структурно остались неизменными, например страница с информацией о дисциплине, на которую попадает пользователь при выборе необходимой дисциплины из представленного списка.

В старой версии студенту была доступна следующая информация: название дисциплины, краткое описание (если его указал преподаватель), информационные сообщения (например, если отправка решений закрыта преподавателем), доступный язык программирования, максимальное количество баллов, список разделов с количеством задач и баллов, а также прогресс пользователя. Также для преподавателя и администратора доступны кнопки добавления учебных разделов, редактирования дисциплины и статистики по учебным группам.

В новой версии было решено оставить структуру без изменений (см. рис. 4).

Технологии программирования (C++)			
ТП			
Доступ к дисциплине ограничен. Вы можете продолжать изучать теоретический материал, но отправка решений на данный момент приостановлена.			
Язык программирования, который можно использовать: C++ (GCC 14.1.0)			
Количество баллов, которое можно набрать, решая задачи: 80			
Всего разделов: 9			
№	Название раздела	Задач	Баллов
1	Линейные алгоритмы	9	8
2	Реализация алгоритмов ветвления	6	6
3	Работа с циклами	4	8
4	Обработка массивов	6	12
5	Работа со строковым типом данных	4	8
6	Обработка функций	3	6
7	Работа с файлами	2	6
8	Векторы	3	6
9	Контрольная работа	10	20

Рисунок 4. Структура элемента с информацией о дисциплине

Но была изменена структура страницы учебных разделов. Ранее студентам для перемещения между разделами предлагалась небольшая панель навигации в нижней части страницы, в которой были предложены кнопки для перехода к следующему и предыдущему разделу дисциплины. Такой подход был не практичным и доставлял неудобства пользователям системы, поэтому данная панель в новой версии сайта удалена, а вместо неё было решено создать отдельную боковую панель, в которой будут представлены: название дисциплины, список разделов с количеством решённых задач, а также кнопки для возвращения к информации о дисциплине и быстрого перехода к списку задач в рамках дисциплины (см. рис. 5).

Боковая панель расположена слева от учебных материалов и предлагает более удобный способ перехода между разделами выбранной пользователем дисциплины.

Технологии программирования (C++)

- Линейные алгоритмы
Решено заданий: 9 из 9.
- Реализация алгоритмов ветвления
Решено заданий: 6 из 6.
- Работа с циклами
Решено заданий: 4 из 4.
- Обработка массивов
Решено заданий: 5 из 6.
- Работа со строковым типом данных
Решено заданий: 3 из 4.
- Обработка функций
Решено заданий: 0 из 3.
- Работа с файлами
Решено заданий: 2 из 2.
- Векторы
Решено заданий: 0 из 3.
- Контрольная работа

К дисциплине К задачам

Основу языка программирования C++ составляет структурное представление кода. Приведем пример программы на языке C++, написанный для консоли (для удобства пропустим строки программы):

```

1 /* первая программа */
2 #include <iostream>
3 #include <cmath>
4 #include <windows.h>
5 #define PI 3.141592653
6 using namespace std;
7 double area (double radius);
8 int main()
9 {
10     setlocale(LC_ALL, "rus");
11     SetConsoleCP(1251);
12     SetConsoleOutputCP(1251);
13     int r, h, v;
14     cout << "\n Введите радиус и высоту цилиндра, разделенные
15     пробелами: ";
16     cin >> r >> h;
17     v = h * area(r);
18     cout << "\n Объем цилиндра радиуса " << r << " и высоты "
19     << h << " равен " << v;
20     return 0;
21 }
22 double area (double radius)
23 {
24     return PI*radius*radius;
25 }

```

В строке 1 записан комментарий. На языке C++ комментарии к программам можно оформлять двумя способами:

- с помощью комбинации `//`, в этом случае комментарий распознается до конца строки;
- с помощью парных комбинаций `/*` и `*/`, в данном случае комментарий располагается между данными парами, таким образом можно оформить многострочный комментарий.

Строки 2 определяют подключаемые библиотеки. Библиотеки подключаются с помощью директивы `#include`. Они служат для подключения функций программы, например библиотека `iostream.h` служит для обеспечения ввода и вывода текста и значений переменных (например, как в строках

Рисунок 5. Страница с содержимым раздела и боковой панелью со списком остальных разделов выбранной дисциплины

Также для большей наглядности слева от названия дисциплины отображается одна из иконок: зелёная, если решены все задания в рамках раздела и жёлтая, если решены не все соответственно.

Структура страницы с условиями задач осталась неизменной: сначала представлено название и условие задачи, затем ограничения на время исполнения программы и память, которую может задействовать программа. После ограничений также указаны количество баллов и теги к задаче (нужны для упрощения поиска задач преподавателю и для понимания основной тематики студентам). Ниже представлены также тестовые данные (только 2 из всего набора).

1.1	
Необходимо посчитать сумму $A + B$	
Входной файл: целые числа A и B	
Выходной файл: целое число	
Пример 1:	
Входной файл: 1 1	
Выходной файл: 2	
Пример 2:	
Входной файл: 3 5	
Выходной файл: 8	
Поток ввода: Стандартный ввод	
Поток вывода: Стандартный вывод	
Ограничение по времени: 1 с.	
Ограничение по памяти: 16 Мб.	
Количество баллов за правильное решение: 0	
Теги к задаче: Простая задача	
Входные данные	Выходные данные
1 1	2
3 5	8

Рисунок 6. Структура страницы с задачей

Ниже условия заданы форма для отправки решения и таблица со всеми отправками для выбранной задачи. Пользователь может выбрать интересующую его отправку и исправить код или задействовать его в другой задаче (см. рис. 7).

Если в дисциплине предусмотрена возможность выбора языка программирования, то пользователь перед отправкой может выбрать соответствующий его решению. Иначе меню выбора недоступно для студентов и преподавателей.

Также если на сайте закрыта отправка решений по техническим причинам или преподаватель закрыл отправку сам, то вместо кнопки отправки решения будет выведено соответствующее сообщение.

Таблица состоит из следующих колонок: ID попытки (представляющие собой ссылку на код решения задачи), время отправки решения с точностью до секунд, выбранный язык программирования, результат проверки.

Язык программирования

C++ (GCC 14.1.0) ▾

Отправить решение

На сайте проводятся технические работы. Проверка решений приостановлена.

В настоящее время функционал сайта обновляется, в ближайшее время работа проверяющей системы будет запущена вновь.

Результаты отправок			
ID попытки	Время отправки решения	Язык программирования	Результат
4520	23.01.25,09:57:27	C++	Принято
862	04.11.23,13:06:32	C++	Принято
861	04.11.23,13:04:26	C++	Принято
860	04.11.23,13:03:57	C++	Принято
859	04.11.23,13:02:27	C++	Неправильный ответ

Рисунок 7. Структура формы отправки решений и таблица отправок

Для студентов структура сайта осталась неизменной, но в то же время было внесено много правок в страницы, которые предназначены для преподавателей.

Одна из наиболее важных страниц на сайте: просмотр отправленных пользователями решений. В стартовой версии не хватало фильтров для ограничения списка решений, а также сама структура страницы была неудобна для использования (см. рис. 8).

Были добавлены: фильтры для просмотра решений, отдельное окно, в котором представлено решение пользователя с возможностью изменения вердикта проверки, удаления решения и копирования кода программы (см. рис. 9).

Просмотр отправленных пользователями решений

Введите ID решения:

Найти

Вывести последние 20 решений

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

Рисунок 8. Ранняя версия страницы просмотра отправленных решений

Также преподавателю представлен подробный лог проверки решения: он может просмотреть результат каждого теста. Лог представлен в виде

отдельной таблицы, в которой представлены следующие столбцы: номер теста, результат проверки, время и память, которое потребовалось для обработки теста.

Отправленные решения

Введите ID решения:

☐ Отмеченные решения

500 Все Фатеенков

Все Все дд.мм.гггг

ID	Имя пользователя	Дата	Задача	ЯП	Вердикт
☆ 4526	Фатеенков Данила Витальевич 1791a	23.01.25,13:42:17	Сложение	C++	RE
☆ 4525	Фатеенков Данила Витальевич 1791a	23.01.25,13:40:23	Сложение	Python	OK
☆ 4524	Фатеенков Данила Витальевич 1791a	23.01.25,13:40:11	Сложение	C++	RE
☆ 4523	Фатеенков Данила Витальевич 1791a	23.01.25,13:39:26	Сложение	Python	OK
☆ 4520	Фатеенков Данила Витальевич 1791a	23.01.25,09:57:27	1.1	C++	OK
☆ 4519	Фатеенков Данила Витальевич 1791a	22.01.25,12:35:35	1.0	Python	OK
★ 4518	Фатеенков Данила Витальевич 1791a	10.01.25,13:39:29	Сортиро...	Python	RE
☆ 4500	Фатеенков Данила Витальевич 1791a	27.12.24,10:57:48	4.3	C++	OK

Рисунок 9. Новая версия страницы просмотра отправленных решений

Были реализованы следующие фильтры для отправленных решений:

1. Поиск по ID отправки.
2. Просмотр «отмеченных решений».
3. Ограничение количества решений в таблице (20, 50, 100, 500).
4. Вердикт проверки решения.
5. ФИО пользователя.
6. Поиск по названию задачи.
7. Поиск по дисциплине.
8. Фильтрация по языку программирования.
9. Фильтр по дате отправки.
10. Поиск по номеру группы.

Результат проверки [\(в отдельном окне\)](#)

```
a,b = map(int,input().split())  
print(a+b)
```

ID отправки: 4525

Результат проверки: Принято

Accepted

Изменить вердикт

Скопировать решение

Удалить решение

Лог проверки решения

№ теста	Результат	Время	Память
1	Accepted	0.151 с.	13260 Кб.
2	Accepted	0.15 с.	13792 Кб.
3	Accepted	0.15 с.	13704 Кб.
4	Accepted	0.151 с.	13452 Кб.
5	Accepted	0.149 с.	13432 Кб.

Рисунок 10. Подробный лог проверки решения для одной из задач в системе

Для удобства оба блока представлены на одной странице, и преподаватель может просмотреть интересующее его решение, выбрав из предложенного списка нужный ID. Для динамической подгрузки решения используется JavaScript, который отправляет на сервер запрос с ID решения и возвращает клиенту информацию о решении. При необходимости преподаватель может открыть решение в отдельном окне, но на специальной для просмотра решений странице не предусмотрен вывод подробного лога проверки решений.

Также преподаватель может отметить «особые» решения для того, чтобы в дальнейшем он мог сослаться на них при необходимости (для решения спорных моментов, для уточнения деталей решения и т.д.). Особые решения отмечены специальным значком «звёздочка», который расположен слева от ID решения. Заполненная звёздочка является отмеченным решением, не заполненная наоборот, соответственно.

Отправленные решения

Введите ID решения:

[Найти](#) [Вывести последние 20 решений](#)

Отмеченные решения

500 Все Фатеенков Название задачи

Все Все дд.мм.гггг Номер группы

ID	Имя пользователя	Дата	Задача	ЯП	Вердикт
4526	Фатеенков Данила Витальевич	23.01.25,13:42:17	Сложение	C++	RE
4525	Фатеенков Данила Витальевич	23.01.25,13:40:23	Сложение	Python	OK
4524	Фатеенков Данила Витальевич	23.01.25,13:40:11	Сложение	C++	RE
4523	Фатеенков Данила Витальевич	23.01.25,13:39:26	Сложение	Python	OK
4520	Фатеенков Данила Витальевич	23.01.25,09:57:27	1.1	C++	OK
4519	Фатеенков Данила Витальевич	22.01.25,12:35:35	1.0	Python	OK
4518	Фатеенков Данила Витальевич	10.01.25,13:39:29	Сортиро...	Python	RE
4500	Фатеенков Данила Витальевич	27.12.24,10:57:48	4.3	C++	OK
4108	Фатеенков Данила Витальевич	21.12.24,19:46:20	5.3	Python	OK

Результат проверки (в отдельном окне)

```
a,b = map(int,input().split())
print(a+b)
```

ID отправки: 4525

Результат проверки: Принят

Accepted

[Изменить вердикт](#) [Скопировать решение](#) [Удалить решение](#)

Лог проверки решения

№ теста	Результат	Время	Память
1	Accepted	0.151 с.	13260 Kб.
2	Accepted	0.15 с.	13792 Kб.
3	Accepted	0.15 с.	13704 Kб.
4	Accepted	0.151 с.	13452 Kб.
5	Accepted	0.149 с.	13432 Kб.

Рисунок 11. Полная структура таблицы просмотра отправленных решений

При открытии решения в отдельном окне выводится отдельная страница, на которой представлен код решения задачи, а также ссылки на остальные решения и возможность копирования решения (см. рис. 12). В старой версии сайта выводилось только код решения задачи и краткая сводка результатов проверки решения.

Также используется специальный модуль PrismJS для определения подсветки синтаксиса кода решения пользователя (планируется добавления подсветки и для окна решений на странице просмотра всех решений).

Данная страница также доступна студентам: попасть на неё можно с помощью выбора нужного решения из таблицы отправок к задаче.

Результат отправленного Вами решения [Скопировать решение](#)

```
1 a,b = map(int,input().split())
2 print(a+b)
```

ID	Дата проверки	Задача	Язык программирования	Результат проверки
4526	23.01.25,13:42:17	Сложение	C++	Runtime / Compilation Error
4525	23.01.25,13:40:23	Сложение	Python	Accepted
4524	23.01.25,13:40:11	Сложение	C++	Runtime / Compilation Error
4523	23.01.25,13:39:26	Сложение	Python	Accepted
1161	06.12.23,19:09:37	Сложение	Undefined	Accepted
583	04.10.23,11:38:23	Сложение	Undefined	Runtime / Compilation Error
582	04.10.23,11:38:02	Сложение	Undefined	Accepted

Рисунок 12. Страница просмотра решений

На сайте также действует система локального антиплагиата. Преподаватель может выбрать интересующую его дисциплину, раздел и задачу, затем ИС на основе алгоритма редакционного расстояния выводит пары решений, процент совпадения структуры которых превышает 85. При этом решения, совпадающие на 100%, автоматически отмечаются на

отклонение (преподаватель может снять соответствующую галочку в поле для обозначения отклонения, если в этом есть необходимость).

В старой версии преподаватель мог выбрать только дисциплину, раздел и задачу (см. рис. 13). Информация о совпадениях включала в себя: ID решений, ФИО студентов и процент совпадения.

Локальный антиплагиат

Выберите задачу

Дисциплина

Технологии программирования (2 семестр) ▼

Раздел

3. Условные операторы в Python ▼

Задача

3.3 ▼

☐	ID1	ID2	Имя студента (ID1)	Имя студента (ID2)	%
☐	1586	1885	Александр Александрович	Александр Александрович	89.73
☐	1586	2031	Александр Александрович	Александр Александрович	87.72
☐	1586	2280	Александр Александрович	Александр Александрович	89.73
☒	1586	2321	Александр Александрович	Александр Александрович	100
☐	1765	2031	Александр Александрович	Александр Александрович	93.41
☒	1885	2280	Александр Александрович	Александр Александрович	100
☐	1885	2301	Александр Александрович	Александр Александрович	90.72
☐	1885	2321	Александр Александрович	Александр Александрович	89.73

Рисунок 13. Старая версия системы антиплагиата на сайте

Не сложно заметить, что локальный антиплагиат не проектировался с учётом того, что в одном семестре системой смогут пользоваться несколько учебных групп (а также то, что новые учебные группы смогут изучать уже загруженные дисциплины). Это создало проблему в проверке заимствований между студентами и подтвердить объективность работы антиплагиата.

В новой версии было принято решение не только изменить форму выбора необходимой задачи, но и также решить вышеописанную проблему. Соответственно был добавлен новый параметр в виде выпадающего списка в форму выбора задачи. В новой версии преподаватель также выбирает дисциплину, раздел и задачу, а также номер учебной группы. При этом рядом с номером учебной группы указывается год зачисления студентов на обучение – это нужно для случая, при котором номера учебных групп совпадут (например, студенты группы 1791, поступившие в 2019-м году, на момент работы системы уже окончили обучение, но при этом информация о их успеваемости может храниться на сайте, пока преподаватель не примет решение удалить группу и всё, что связано с успеваемостью студентов, а следующая группа с таким же номером поступит в университет в 2029-м году, что может усложнить выбор нужной группы в дальнейшем). Сама форма выбора задачи и учебной группы меньше занимает места в новой версии, что позволяет вывести на страницу больше из списка заимствований (см. рис. 14).

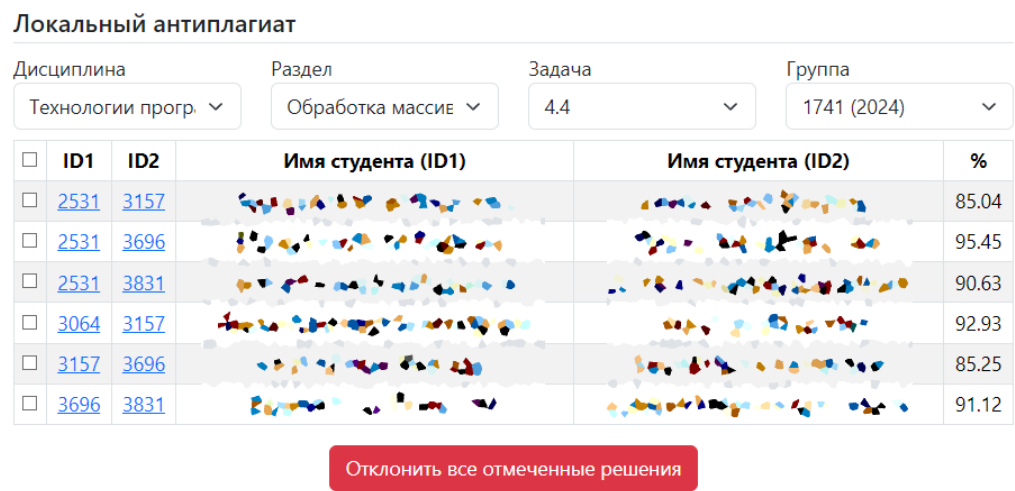


Рисунок 14. Новая система антиплагиата на сайте

Последнее, что стоит отметить в контексте функциональности ИС – формирование отчётов об успеваемости студентов. Преподаватель может сформировать отчёт и сохранить его в PDF, в котором будет отражена успеваемость студентов выбранной группы по определённой дисциплине (см. рис. 15).



Рисунок 15. Отчёт об успеваемости студентов в рамках дисциплины «Технологии программирования (Python)»

Преподаватель также может смотреть статистику и на сайте. Для этого необходимо перейти на соответствующую страницу и выбрать интересующую группу. Для отображения успеваемости задействован объект DataTable в JS, позволяющий внедрить дополнительный функционал в работу таблицы (например, сортировка по столбцам или поиск в таблице и пагинация данных по количеству выводимых строк).

Также преподаватель может выбрать отдельный раздел дисциплины для получения более подробной статистики об успеваемости группы (будет отображаться таблица с информацией о том, какие задачи решил каждый

студент). Такое нововведение призвано упростить процесс определения западающих знаний и детализации работы студентов в системе (см. рис. 16).

Статистика успеваемости по разделу дисциплины

Раздел дисциплины										
Линейные алгоритмы										
ФИО студента	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	
Иванов И.И.	-	-	-	-	-	-	-	-	-	
Петров П.П.	-	-	-	-	-	-	-	-	-	
Сидоров С.С.	-	-	-	-	-	-	-	-	-	
Кузнецов К.К.	-	-	-	-	-	-	-	-	-	
Левченко Л.Л.	+	+	+	+	+	+	+	+	+	
Михайлов М.М.	+	+	+	+	+	+	+	+	+	
Новиков Н.Н.	+	+	+	+	+	+	+	+	+	
Осипов О.О.	+	+	+	+	+	+	+	+	+	
Рябенко Р.Р.	-	-	-	-	-	-	-	-	-	
Смирнов С.С.	+	+	+	+	+	+	+	+	+	
Тихонов Т.Т.	+	+	+	+	+	+	+	+	+	
Ульянов У.У.	-	-	-	-	-	-	-	-	-	
Фролов Ф.Ф.	+	+	+	+	+	+	+	+	+	

Рисунок 16. Детализация успеваемости группы по одному из разделов в рамках дисциплины «Технологии программирования (Python)»

Помимо обновления визуальной составляющей и внедрения новых возможностей в систему путём исправления недочётов предыдущей версии, также был проведён рефакторинг всего программного кода системы и были исправлены многие недочёты, а также повышена производительность ИС.

В первую очередь стоит отметить переход с генерации содержимого страниц с помощью PHP на шаблонизатор Twig – полноценный движок обработки HTML-шаблонов, который позволяет разделить разметку страницы от PHP-скриптов.

Самый простой пример: вывод разделов в рамках выбранной дисциплины (вычисление количества задач и баллов для каждого раздела в том числе). Ниже представлен код старой версии, в котором содержимое элемента страницы генерируется с помощью PHP (SQL-запросы скрыты):

```
<tbody>
<?php
$num = 1;
foreach($sections_arr as $section) {
    $max_weight_section = 0;
    $tasks_weight = R::getAll("SELECT ...");
    foreach($tasks_weight as $weight_num) {
        $max_weight_section += $weight_num['weight'];
    }
    echo "<tr>";
    echo "<td class='text-center'><b>". $num. "</b></td>";
    echo "<td class='text-center'><a href='section.php?section_id=" . $section['id'] . "'> " . $section['name'] . " </a></td>";
    echo "<td class='text-center'>".count(R::getAll("SELECT ...")). "</td>";
```

```

        echo "<td class='text-
center'>".$max_weight_section."</td>";
        echo "</tr>";
        $num++;
    }
    ?>
</tbody>

```

Такой код сложно читать и отлаживать его работу для исправления возникающих недочётов. Для решения данной проблемы был внедрён шаблонизатор Twig. Ниже представлен PHP-скрипт, который выбирает все разделы дисциплины и возвращает результат в виде ассоциативного массива. Также для вычисления количества баллов и задач были добавлены соответствующие триггеры в БД для упрощения обработки данных:

```

function get_sections() {
    $id = (int)$_GET['id'];
    $sections_data_query = "SELECT ...";
    $sections_data = R::getAll($sections_data_query, [$id]);
    $data = [];
    foreach ($sections_data as $index => $section) {
        $data[] = [
            "id" => $section['id'],
            "num" => $index + 1,
            "name" => $section['name'],
            "tasks_number" => $section['tasks_number'],
            "points" => $section['points']
        ];
    }
    return $data;
}

```

Полученный массив достаточно передать в Twig на этапе генерации страницы на основе заранее заготовленного шаблона. Шаблонизатор автоматически подставит содержимое массива в выбранный элемент:

```

<tbody>
    {% for section in sections %}
    <tr>
        <td class="text-center"><b>{{ section.num
}}</b></td>
        <td class="text-center"><a class="link-offset-2
link-offset-3-hover link-underline link-underline-opacity-0
link-underline-opacity-75-hover" href="section.php?section_id={{
section.id }}">{{ section.name }}</a></td>
        <td class="text-center">{{ section.tasks_number
}}</td>
        <td class="text-center">{{ section.points }}</td>
    </tr>
    {% endfor %}

```

</tbody>

Twig в некотором смысле заимствует классические конструкции из алгоритмизации, а именно: операторы ветвления, циклы, тернарные операторы и другие. Такой подход делает данный движок удобным для работы и предлагает разработчику лёгкий синтаксис, который можно освоить за короткий промежуток времени.

Синтаксис Twig представляет собой в некотором роде гибрид между Python и PHP, в нём также представлены встроенные агрегирующие функции, позволяющие проводить вычисления на стороне шаблонизатора.

Например, можно рассмотреть функцию нахождения длины массива – такая используется на странице задач для вывода тегов (если тегов нет, то всё содержимое данной характеристики будет проигнорировано во время генерации страницы). Для нахождения длины массива нужно задействовать функцию `length` со следующим синтаксисом: `array|length` (где на первой позиции сам массив, а на второй функция). Соответствующий шаблон в коде разметки выглядит следующим образом:

```
{% if tags|length != 0 %}
    <p><em>Теги к задаче: </em>{% for tag in tags %} <span
class="badge text-bg-secondary">{{ tag.tag_name }}</span> {%
endfor %}</p>
{% endif %}
```

Также предусмотрена работа с тернарными операторами. На той же странице с задачей выводится информация о потоках ввода и вывода данных в программе (либо стандартный, либо через файлы). В ограничениях это явным образом указывается. В старой версии генерация элемента с потоками ввода и вывода выглядит следующим образом:

```
<tr>
    <td>Поток ввода:</td>
    <td><?php if ($task_data[0]['filestream'] == 0) { echo
"Стандартный ввод"; }
        else echo "input.txt";?></td>
</tr>
<tr>
    <td>Поток вывода:</td>
    <td><?php if ($task_data[0]['filestream'] == 0) { echo
"Стандартный вывод"; }
        else echo "output.txt";?></td>
</tr>
```

У данного варианта вывода есть много проблем (как минимум можно было задействовать тернарный оператор на стороне PHP вместо стандартного ветвления), но главный недостаток – загроможденность конструкциями, из-за чего

код тяжело читать. В Twig шаблоне данная секция выглядит компактнее и читать такой код намного легче:

```
<tr>
    <td>Поток ввода: {{ task_data.filestream == 1 ?

```

Помимо внедрения шаблонизатора Twig также были переработаны многие SQL-запросы для достижения большей эффективности обработки данных (в том числе и для уменьшения времени обработки запросов). Многие запросы были переписаны с использованием операторов объединений JOIN. Например, в старой версии был представлен следующий ряд запросов на нахождение списка дисциплин для учебной (названия полей изменены):

```
$link_array_query = "SELECT did FROM gdl WHERE gid = ? ORDER
BY did";
$link_array = R::getAll($link_array_query, [$gid]);
for($i=0;$i<count($link_array);$i++) {
    $linkid = (int) $link_array[$i]['discipline_id'];
    $discipline_query = "SELECT * FROM d WHERE id =
'$linkid'";
    $discipline = R::getAll($discipline_query);
    $disciplineid = (int) $discipline[0]['id'];
    $sections_query = "SELECT id FROM sect_data WHERE did =
'$disciplineid'";
}
```

В данной части скрипта выполняется 3 запроса: извлечение всех ID дисциплин, к которым привязаны учебные группы; получение дисциплины по значениям из массива ID, который был получен ранее; получение id раздела из соответствующей таблицы по связи между двумя таблицами.

Этот ряд запросов можно сократить до одного, который представлен ниже:

```
SELECT d.id,d.name, d.pid, d.sq, d.ist
FROM d
JOIN gdl ON gdl.did = d.id
WHERE gdl.group_id = ?
```

По такому же принципу были обновлены практически все запросы в коде информационной системы, что позволило реструктурировать программный код и облегчить логику работы ИС.

Также были закрыты дыры в безопасности ИС и возможности допуска SQL-инъекций или XSS-атак сведены к минимуму (при этом продолжается анализ кода программы с целью выявления потенциальных мест, откуда может произойти атака).

4. Выводы

Результатом данной работы стала обновлённая (фактически написанная с нуля с использованием новых инструментов, но при этом с соблюдением технического задания оригинальной версии) версия сайта обучения программированию в ПГУ им. Шолом-Алейхема. Был переписан весь код с целью улучшения производительности и читаемости кода, а также переработан внешний вид ИС.

Студенты очной формы обучения уже в предстоящем семестре смогут ознакомиться с новой версией сайта и оставить свои отзывы для внесения дальнейших корректировок в работу и оформление сайта.

Библиографический список

1. Нышанова А.С., Маликова З.Т. Использование фреймворка Bootstrap в создании динамических сайтов // Наука. Образование. Техника. - 2019. - № 1 (64). - С. 24-30.
2. Касымбекова Ч.М., Матисаков Ж.К. Использование Bootstrap для создания веб-приложений и сайтов // В сборнике: Физик: ученый, педагог, наставник. сборник научных трудов. Саратовский национальный исследовательский государственный университет имени Н.Г. Чернышевского. Саратов. 2023. С. 176-179.
3. Балаев А.К., Государев И.Б. Методологии рефакторинга унаследованного кода. Подготовительная фаза рефакторинга // В сборнике: Альманах научных работ молодых ученых университета ИТМО. XLVIII научная и учебно-методическая конференция Университета ИТМО. 2019. С. 18-19.
4. Кириллов В.Ю., Костюхин А.И. Использование методологии разработки через тестирования для рефакторинга MVC приложения // Электронный журнал: наука, техника и образование. 2019. № 4 (27). С. 35-43.
5. Дерюгин Д.С. Рефакторинг, как метод обеспечения качества продукта // Синергия Наук. 2020. № 43. С. 409-415.
6. Басов А.С. Методы рефакторинга кода // Вестник науки. 2020. Т.2. № 8 (29). С. 102-106.