





**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-  
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»**  
(ФГБОУ ВО «ЮУрГГПУ»)

**ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ**  
**КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И  
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ**

**Разработка автоматизированной системы проверки задач по программированию**  
**Выпускная квалификационная работа**  
**по направлению 09.03.02 Информационные системы и технологии**  
**Направленность программы бакалавриата**  
**«Информационные технологии в образовании»**

Проверка на объем заимствований:  
\_\_\_\_\_ % авторского текста

Работа \_\_\_\_\_ к защите  
\_\_\_\_\_ рекомендована/не рекомендована

« \_\_\_ » \_\_\_\_\_ 20\_\_ г.  
зав. кафедрой И, ИТ и МОИ

\_\_\_\_\_ Рузаков А.А.

Выполнил:  
Студент группы ОФ-413/095-4-1  
Шагеев Радмир Рифкатович

Научный руководитель:  
к.п.н, доцент кафедры И, ИТ и МОИ  
Паршукова Наталья Борисовна

**Челябинск**  
**2019**

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
Глава 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И НАЗНАЧЕНИЕ ПРОЕКТИРУЕМОЙ ПОДСИСТЕМЫ.....	5
1.1. Характеристика объекта автоматизации .....	5
1.2. Описание проектируемой системы .....	13
1.3. Сравнительный анализ аналогичных программных продуктов 15	
Вывод по главе 1 .....	17
Глава 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «TEST-IN»       18	
2.1. Выбор архитектуры информационной среды и инструментальных средств реализации.....	18
2.2. Описание модели данных .....	20
2.3. Описание реализации основных функций системы .....	24
Вывод по главе 2 .....	25
Глава 3. ДОКУМЕНТИРОВАНИЕ ПОДСИСТЕМЫ.....	26
3.1. Испытание информационной системы .....	26
3.2. Руководство пользователя .....	27
3.3. Технико-экономическое обоснование .....	38
Вывод по главе 3 .....	42
ЗАКЛЮЧЕНИЕ .....	43
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	44

## ВВЕДЕНИЕ

В современном обществе трудно представить жизнь без информационных технологий, несмотря на то что в недалеком прошлом человечество прекрасно обходилось и без них, сейчас они играют немаловажную роль в нашей жизни упрощая ее и позволяя использовать наше время более продуктивно. Процесс информатизации коснулся всех сфер человеческой жизни в том числе и обучения. Процесс обучения программированию с приходом информационных технологий стал проще и быстрее. Учебные заведения используют сложные методики организации тестирования работ по программированию, однако не каждый вуз предоставляет возможность преподавателю провести тестирование алгоритма, написанного студентом легко и удобно. Немаловажным является получение быстрого доступа студента к результатам тестирования его алгоритма. Сегодня для получения подобной информации студенту необходимо лично явиться на консультацию, которая проходит раз в неделю, или же связаться с преподавателем по электронной почте, а потом дожидаться результатов тестирования, при этом отнимая время преподавателя и нагружая его еще большей работой. Автоматизация этого процесса позволит студентам проводить тесты написанной им программы из любого места и позволит сэкономить время как студенту так преподавателю.

Таким образом, **целью** работы является разработка информационной системы, которая позволит студенту упростить процедуру сдачи работы и в автоматическом режиме получить оценку за выполненное им задание, а также сократить преподавателю время на проверку работ студентов.

### **Задачи проекта:**

- описать структуру и бизнес-процессы, действующие при проверке работ студентов по программированию;
- проанализировать аналоги информационной системы;
- описать и утвердить техническое задание на разработку автоматизированной информационной системы;

- изучить документацию необходимую для создания информационной системы;
- разработать дизайн макет системы;
- выполнить верстку по готовому макету и реализовать логику;
- провести тестирование системы на группе студентов;
- внедрить систему в инфраструктуру организации;

Практическая значимость заключается в разработке автоматизированной информационной системы, позволяющей упростить процесс проверки и оценивания работ по программированию студентов, а также снять часть нагрузки с преподавателя.

# **Глава 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И НАЗНАЧЕНИЕ ПРОЕКТИРУЕМОЙ ПОДСИСТЕМЫ**

## **1.1. Характеристика объекта автоматизации**

### **1.1.1. Описание образовательной организации**

Заказчиком является кафедра «Информатики, информационных технологий и методики обучения информатике» (ИИТиМОИ), федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Южно-Уральский государственный гуманитарно-педагогический университет» (ЮУрГГПУ).

На кафедре ведется подготовка по следующим дисциплинам:

- технологии программирования;
- web-дизайн;
- инструментальные средства информационных систем;
- управление данными;
- архитектура информационных систем;
- и многие другие.

Так же кафедре ведется научная деятельность по следующим направлениям:

- педагогические системы с искусственным интеллектом;
- менеджмент качества в общем образовании;
- информатика в профильной школе.

Преподавателями кафедры в издательстве БИНОМ выпущены учебные и методические пособия по элективным курсам: «Компьютерные сети»; «Основы компьютерной алгебры»; «Основы криптографии».

С 2012 по 2014 год велась научно-исследовательская работа в рамках государственного задания Министерства образования и науки РФ по проекту «Компьютерные технологии в профильном обучении» (руководитель Г. Б. Поднебесова).

На кафедре открыта аспирантура по специальности теория и методика обучения и воспитания (информатика уровень общего и профессионального образования). Разработаны программы повышения квалификации научно-педагогических работников высших учебных заведений.

С 2014 года кафедра является участником «Программы по обучению студентов программированию мобильных приложений для ios» компании Apple [5].

### 1.1.2. Описание структуры организации заказчика системы и основных бизнес-процессов

Кафедра «Информатики, информационных технологий и методики обучения информатике» ведет подготовку по дисциплинам, связанных с изучением программирования на различных языках: Pascal, C#, Python и другие [5].

Процесс изучения проходит следующим образом. Преподаватель проводит лекцию, далее преподаватель со студентами разбирают несколько примеров, после чего студентам необходимо самостоятельно решить индивидуальную задачу по варианту и сдать ее преподавателю. Данный процесс имеет множество сложностей, связанных с тем, что в данный момент для связи преподавателя и студента существует два пути.

Первый – это связь студента и преподавателя по электронной почте. Данный способ не требует от студента физических затрат, но приводит к потере времени так как не каждый преподаватель может себе позволить мгновенно проверять решенные студентом индивидуальные задания, отправленные им на почту.

Второй – это когда студент приходит на консультации к преподавателю. Данный способ потребует от студента определенные физические затраты, но сэкономит время на проверку индивидуального задания. Однако, данный способ несет в себе огромный недостаток, заключающийся в том, что консультации проводятся раз в неделю.

Однако и первый и второй вариант несет в себе для преподавателя проблему того, что ему необходимо провести N количество тестировании программного алгоритма, написанного студентом, что отнимает очень много времени.

Для формального понимания тонкостей существующего процесса сдачи индивидуальных заданий рассмотрим соответствующие бизнес процессы.

Бизнес-процесс – комплекс мероприятий, совокупность разных видов деятельности, направленных на создание максимально ценного для потребителя продукта (услуги), а также оптимизации работы организации в целом.

Методология моделирования IDEF были разработаны в рамках реализации программы интегрированной компьютерной поддержки производства ICAM в США в середине 70-х годов с целью улучшения взаимодействия специалистов, занимающихся интеграцией существующих производственных и организационных систем.

Методология IDEF три частных методологии моделирования, основанные на графическом представлении системы: IDEF0, IDEF1, IDEF2.

Контекстная диаграмма – это модель, представляющая систему как набор иерархических действий, в которой каждое действие преобразует некоторый объект или набор объектов. Высшее действие иерархии называется действием контекста – это самый высокий уровень, который непосредственно описывает систему. Уровни ниже называются порожденными декомпозициями и представляют подпроцессы родительского действия.

Затем проводится функциональная декомпозиция – разбиение описанной системы на наиболее крупные фрагменты (диаграммы декомпозиции). Декомпозиция каждого из фрагментов системы проводится до достижения необходимого уровня детализации. Бизнес-процесс сдачи индивидуальных заданий по программированию (рисунок 1).

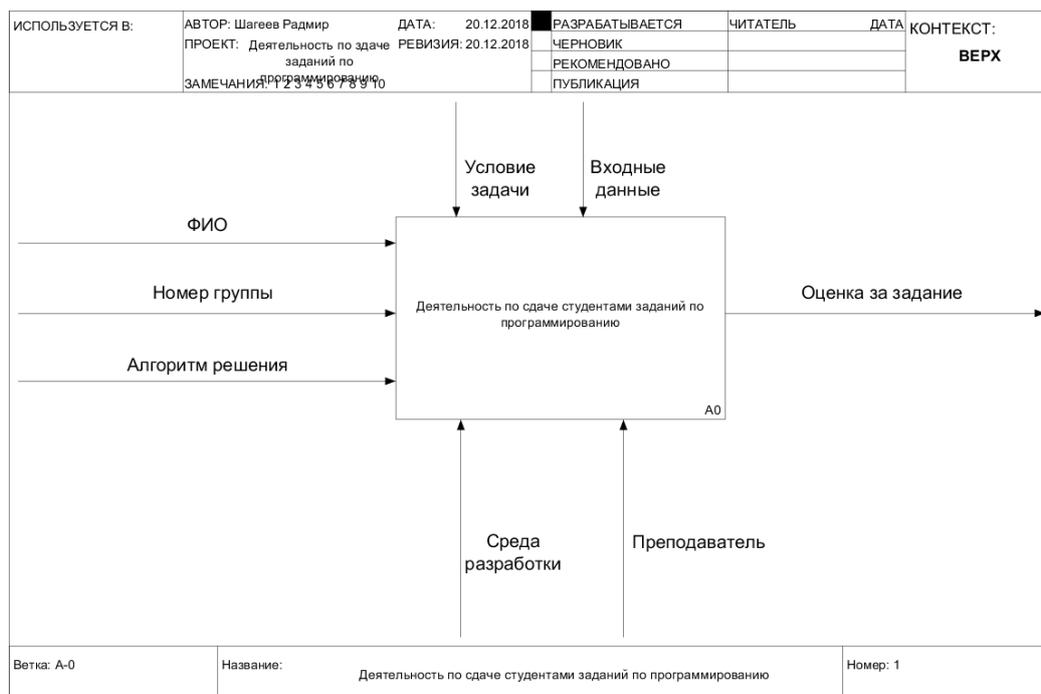


Рис. 1. Контекстная диаграмма «Бизнес-процесс сдачи заданий по программированию»

Описание стрелок контекстной диаграммы:

- **Вход (Input)** – материалы либо информация, используемые для получения результата (выхода). В данном случае на вход поступают ФИО студента, номер группы, алгоритм решения.
- **Выход (Output)** – материалы либо информация, производимые работой. Результатом проверки задания является оценка о выполнении.
- **Управление (Control)** – стандарты, стратегии и правила, которыми руководствуется работа. Выполнение заданий осуществляется по условию задачи с использованием входных данных.
- **Механизм (Mechanism)** – ресурсы, выполняющие работу. В данном случае механизмами можно считать преподавателей и программное обеспечение.

Диаграммы декомпозиции предназначены для детализации функций и получаются при разбиении контекстной диаграммы на крупные подсистемы и описывающие подсистемы, и их взаимодействие. Единственная функция, представленная на контекстной диаграмме верхнего уровня, может быть разложена на основные подфункции посредством создания дочерней

диаграммы. В свою очередь, каждая из этих подфункций может быть разложена на составные части посредством создания дочерней диаграммы следующего, более низкого уровня, на которой некоторые или все функции также могут быть разложены на составные части. Каждая дочерняя диаграмма содержит дочерние блоки и стрелки, обеспечивающие дополнительную детализацию родительского блока.

Декомпозиция контекстной диаграммы представлена на рисунке 2.

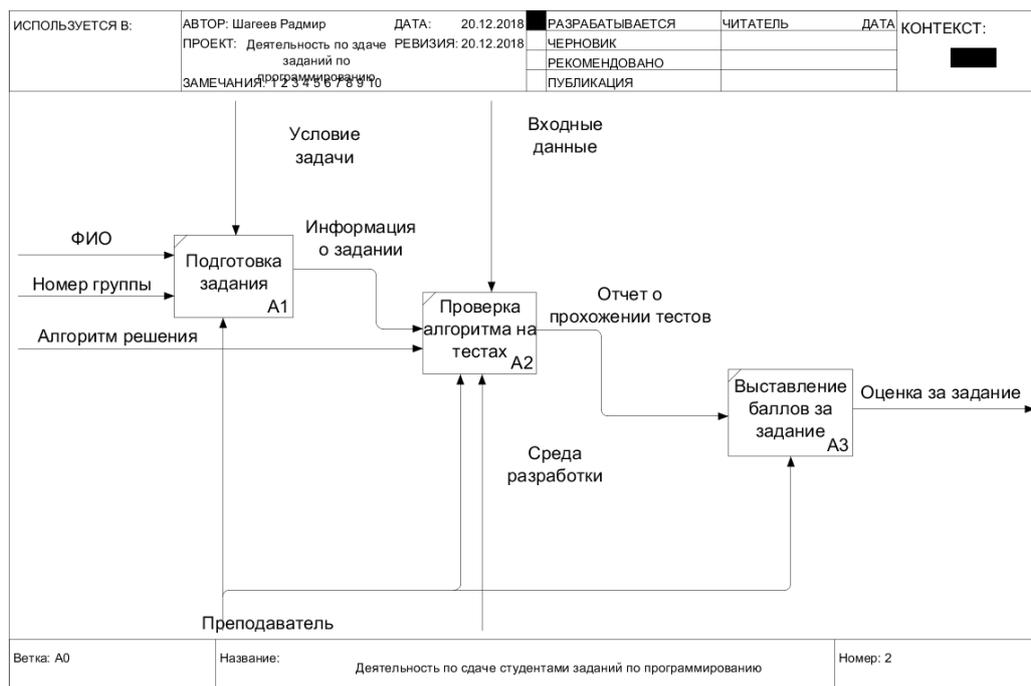


Рис. 2. Декомпозиция контекстной диаграммы

После декомпозиции контекстной диаграммы проводится декомпозиция каждого блока диаграммы A0 на более мелкие фрагменты и так далее, до достижения нужного уровня подробности описания.

Выше был рассмотрен бизнес-процесс сдачи индивидуальных заданий по программированию. Данный процесс можно было бы ускорить и упростить если бы в университете использовалась система автоматизированной проверки заданий по программированию.

Рассмотрим процесс сдачи заданий по программированию с использованием автоматизированной системы рисунок 3.

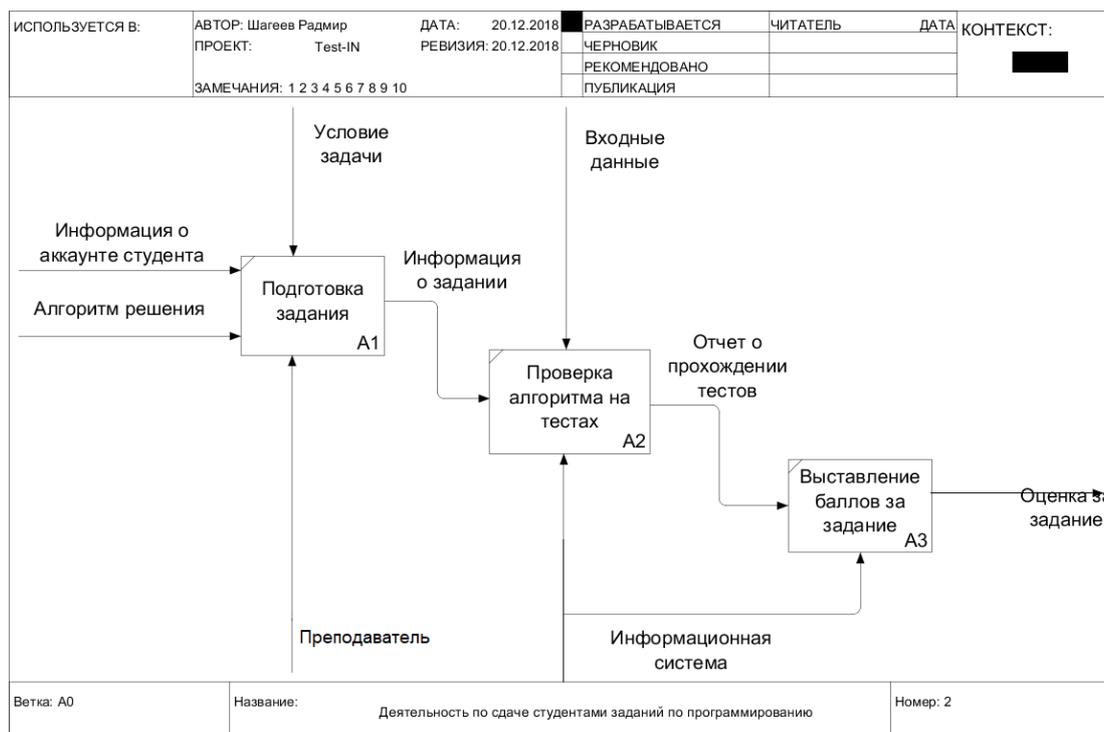


Рис. 3. Декомпозиция контекстной диаграммы

Исходя из выше приведенного рисунка, мы можем увидеть, что используя автоматизированную информационную систему мы снимаем нагрузку с преподавателя, передавая рутинную работу такую как проверка алгоритмов и выставление оценки автоматизированной информационной системе.

### 1.1.3. Описание средств автоматизации работы организации заказчика

На кафедре имеются компьютерные аудитории, которые имеют следующие возможности:

- рабочие компьютеры, через которых студенты используют сеть Интернет, а также внутреннюю сеть организации;
- сетевое соединение со скоростью 1 Гбит/с;
- сервер, который управляет внутренней сетью организации.

Структура сети рисунок 4, является многоуровневой, входит в домен csru.ru, связана между собой узлами коммутации и маршрутизации, также имеет подключение к сети Интернет. Структура сети имеет форму «звезды» - единственной топология сети с явно выделенным центром, к которому подключаются все остальные абоненты [1].

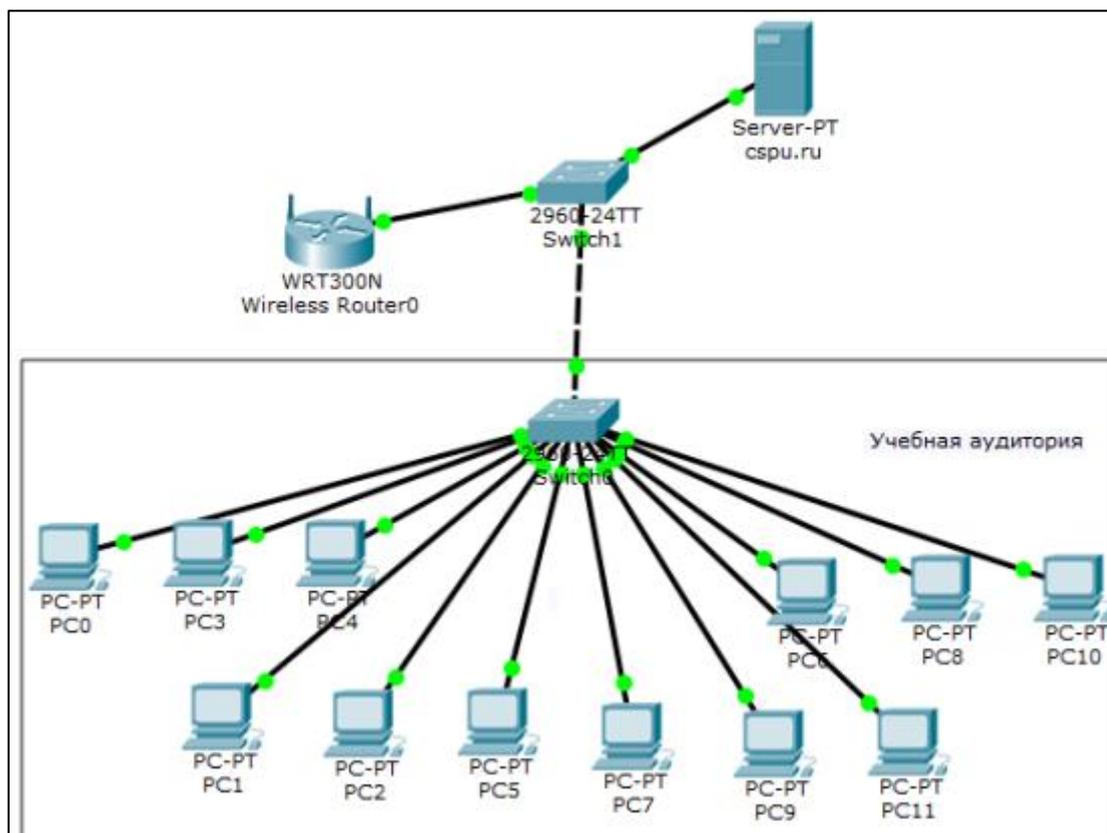


Рис. 4. Структура сети ЮУрГГПУ

Кафедра «Информатики, информационных технологий и методики обучения информатике» используя информационные технологии и автоматизацию пытается по максимуму упростить учебный процесс. На кафедре используется SharePoint что позволяет студентам быстро получить доступ к учебным и методическим материалам во время и вне учебных занятий, что позволяет студенту быть готовым к лабораторной работе предстоящего занятия рисунок 5.

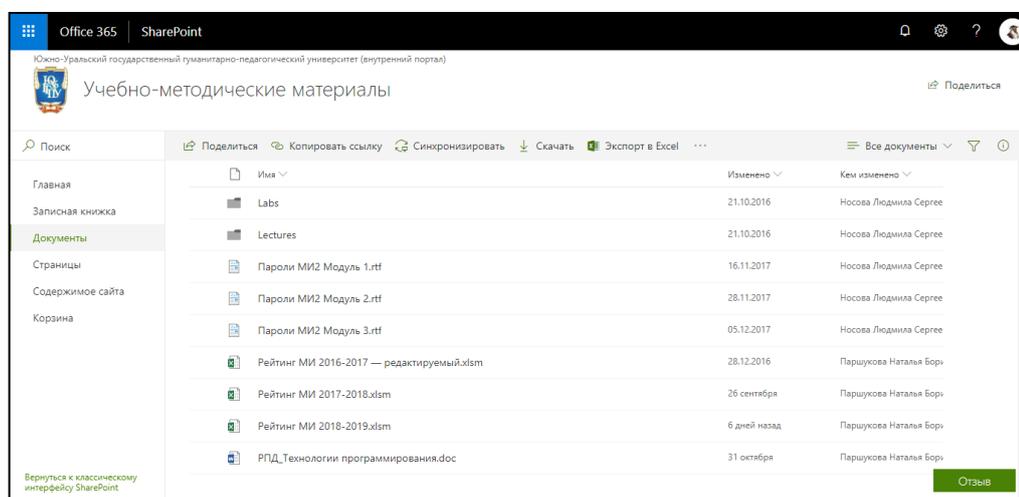


Рис. 5. Учебно-методические материалы.

Так же на портале студент может получить актуальную информацию о рейтинге по той или иной дисциплине рисунок 6.

№	Фамилия Имя	Текущий контроль							Промежуточная аттестация							Индивидуальный рейтинг по дисциплине, %	Оценка по дисциплине				
		Тест 1	Простые типы данных	Тест 2	Условный оператор	Циклы	Простые типы данных	Условный оператор	Циклы	Повторяющийся коэффициент	Понижающий коэффициент	Текущий контроль, %	1 этап					2 этап			
													I уровень Знать (Тест 3)	II уровень Уметь (Процедуры)	III уровень Владеть (Функции)			Коэффициент сформированности компетенции	За 1 этап, %	Зачет	За 2 этап, %
Максимальный балл							Максимальное количество заданий (операций)			Максимум											
1	Назаренко	5	9	5	15	10	10	0,1			52,8%				0,0	0%		0,0%	52,8%	незачтено	18,2
2	Комарова	7	5	6	5	8	10	0,2			53,7%	8			0,2	0%		0,0%	53,7%	незачтено	
3	Романишина	7	5	11	5	15	10	0,1			69,6%	9			0,2	0%		0,0%	69,6%	зачтено	5,16
4	Смирнов	9	5	12	5	15	10	0,3			93,6%	9			0,2	0%		0,0%	93,6%	зачтено	
5	Фикин АКАДЕМ										0,0%				0,0	0%		0,0%	0,0%	незачтено	
6	Черняева	8	5	10	5	15	10	0,2			82,5%	10			0,2	0%		0,0%	82,5%	зачтено	1,15
7	Шелекето	9	5	12	5	10	10	0,2			66,8%	3			0,1	0%		0,0%	66,8%	зачтено	
8	Шумаков	11	5	12	5	15	10	0,3			96,4%	9			0,2	0%		0,0%	96,4%	зачтено	
9	Нажипова										0,0%				0,0	0%		0,0%	0,0%	незачтено	
10	Окищева	11	5	12	5	15	10	0,3			96,4%	8			0,2	0%		0,0%	96,4%	зачтено	

Рис. 6. Рейтинг студентов по дисциплине.

Исходя из написанного выше мы можем сделать вывод о том, что у заказчика есть все необходимое оборудование и программное обеспечение, а так же высокоскоростное интернет-соединение для внедрения автоматизированной информационной системы.

## 1.2. Описание проектируемой системы

### 1.2.1. Назначение системы

Информационная система «Test-IN» является web-приложением, предназначенным как для студента, так и для преподавателя ускорить процесс сдачи проверки и доступа к заданиям. Используя данную систему преподавателю, необходимо лишь создать курс, добавить в него задания и установить баллы за каждое задание. Студенту же необходимо зайти на сайт информационной системы решить задачу и отправить алгоритм решения на автоматическое тестирование после прогонки трех тестов студент получит балл за решение задачи который выставит ему система. Использование данной системы позволит преподавателю сконцентрировать свое внимание на других не менее важных задачах и сократить время на прием заданий студентов.

### 1.2.2. Цели создания системы

Основными целями системы является:

- добавление заданий по программированию;
- редактирование заданий по программированию;
- проверка заданий по программированию;
- выставление баллов за решенное задание.

### 1.2.3. Требования к структуре и функционированию системы

Структурная схема изображена на рисунке 7.

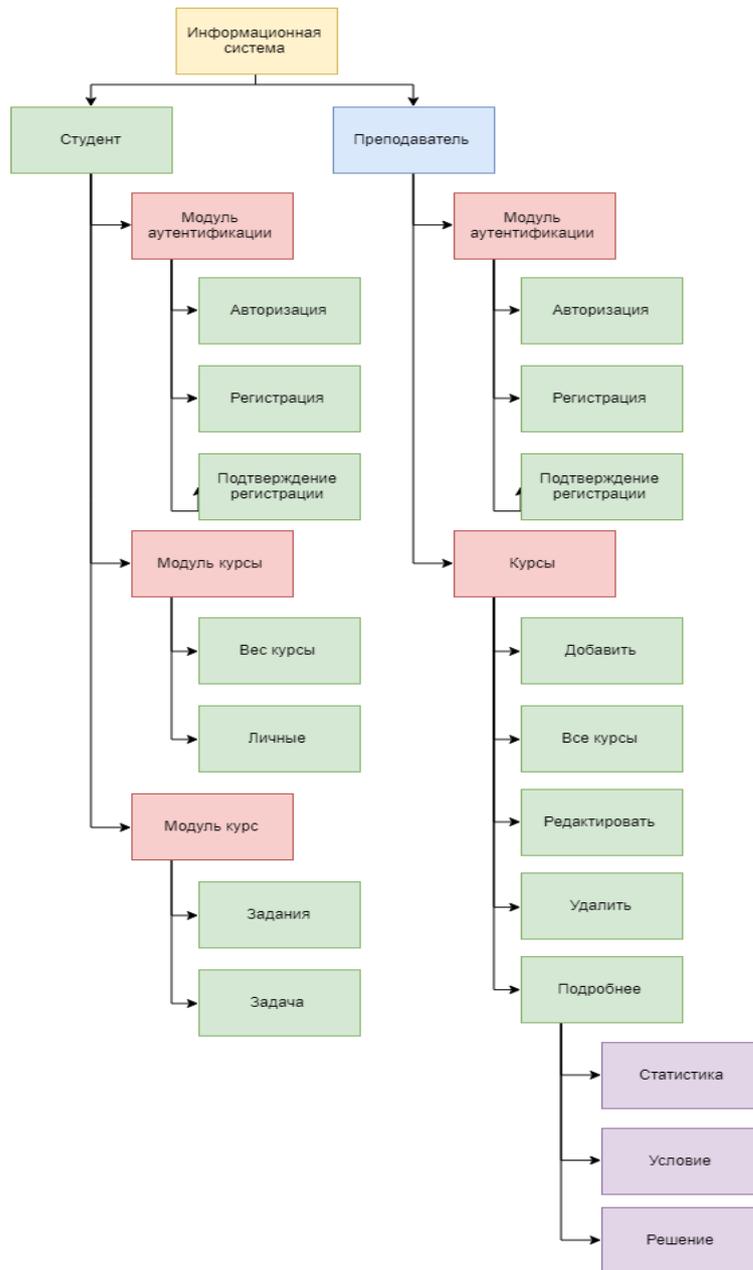


Рис. 7. Структурная схема АИС

В АИС имеются следующие функциональные подсистемы:

- подсистема пользователи;
- подсистема курсы;
- подсистема проверки заданий.

#### Подсистема пользователи

Позволяет регистрировать, авторизовать пользователей двух типов студент и преподаватель.

## Подсистема курсы

Позволяет преподавателю добавлять, удалять, редактировать, просматривать статистику по курсу. Так же отображает курсы и задания студентам.

## Подсистема проверки заданий

Проверяет алгоритм, написанный студентом на соответствие всем тестам, выставляет баллы за задания собирает статистику.

### **Основные требования к системе:**

- система не должна требовать установки;
- должна присутствовать поддержка различных операционных систем;
- система не должна быть ресурсоемкой.

## **1.3. Сравнительный анализ аналогичных программных продуктов**

### **1.3.1. «Питонтьютор»**

Автоматизированная система проверки алгоритмов, написанных на языке Python.

#### **Плюсы:**

- сайт с удобным и понятным интерфейсом;
- лекционный материал по теме задания;
- подробно расписанные задания;
- удобная форма написания и разработки кода;
- выполнение программы по шагам;
- удобные для восприятия тесты алгоритма.

#### **Минусы:**

- возможность исполнять только один язык программирования;
- фиксированное количество заданий, без возможности сторонним авторам добавлять свои задания.

### 1.3.2. Информатикс

Автоматизированная система проверки алгоритмов, работает на CMS moodle для автоматизированной проверки использует модуль contestester.

#### **Плюсы:**

- большое количество языков программирования на выбор;
- возможность создавать курсы и добавлять задания сторонним авторам.

#### **Минусы:**

- отсутствие формы ввода и компиляции алгоритма;
- отсутствие доступа к результатам теста написанного алгоритма со стороны ученика;
- Отсутствие UX.

## **Вывод по главе 1**

В данной главе был описан анализ организационной структуры заказчика – кафедры «Информатики, информационных технологий и методики обучения информатике» (ИИТиМОИ), федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Южно-Уральский государственный гуманитарно-педагогический университет». Это позволило больше узнать о его работе со студентами и определить необходимость внедрения систем автоматизации.

Была составлена контекстная диаграмма деятельности по проверке индивидуальных заданий студентов. Представлена схема локальной сети. Это показало возможность автоматизации процесса решения и сдачи индивидуальных заданий по программированию.

## Глава 2. ПРОЕКСИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «TEST-IN»

### 2.1. Выбор архитектуры информационной среды и инструментальных средств реализации

Главной задачей работы является создание системы, которая позволит в автоматизированном режиме проверять задачи по программированию. Для компиляции кода необходим компилятор, было решено использовать онлайн сервис и тут два варианта Rextester либо Sphere Engine, выбор пал на онлайн сервис Rextester [14] так как Sphere Engine [17] имеет сложное API, а так же является платным. В связи с этим, целесообразно будет реализовать систему в виде web-приложения. В качестве серверного языка было решено выбрать PHP. Для хранения информации о пользователях, курсов, заданий и достижений нам понадобится база данных, решено было использовать СУБД MySQL.

Использование сервиса Rextester позволит избежать необходимости создания собственного компилятора и разгрузить систему переключая всю работу на сторонний сервис. Rextester является онлайн инструментом для компиляции написанных вами алгоритмов. Rextester является open source проектом, те исходный код доступен широкой публике. Причиной выбора данного сервиса послужило то, что данный инструмент поддерживает более 50 языков программирования, среди них «C#, VB.NET, F#, Java, Python, C (gcc), C++ (gcc), PHP, Pascal, Objective-C, Haskell, Ruby, Perl, Lua, Nasm, Sql Server, Javascript», а также он является бесплатным. Онлайн компилятор Rextester имеет встроенные API работающие по типу отправить POST, принять JSON с результатом временем компиляции и затрачиваемой памятью [14].

Создание веб приложения позволит избежать необходимости установки системы на устройство пользователя, а также позволит охватить большой спектр разнообразных устройств независимо от их операционной системы. Веб-приложения – это специальный вид приложений, которые

работают в глобальной сети интернет по протоколу HTTP. Как правило, веб-приложения не требуют установки дополнительного программного обеспечения на стороне клиента, а вся логика, в основном, выполняется на стороне сервера. Для отображения пользовательского интерфейса используется браузер – программа, способная распознавать язык разметки HTML и сопутствующие технологии – таблицы стилей CSS, клиентский скриптовый язык программирования JavaScript и т.д. Браузер обычно принято называть "тонким клиентом", т.е. клиентом, который содержит минимальное количество бизнес-логики [3].

На сегодняшний день PHP является наиболее распространенным языком веб-программирования. Подавляющее большинство сайтов и веб-сервисов в интернете написано с помощью PHP. По некоторым оценкам PHP применяется более чем на 80% сайтов, среди которых такие сервисы, как facebook.com, vk.com, baidu.com и другие [11]. И такая популярность не удивительна. Простота языка позволяет быстро и легко создавать сайты и порталы различной сложности. Одной из главных причин выбора данного языка программирования заключается в его «эффективности» при программировании многопользовательских сред что не маловажно так как с системой одновременно может работать огромное количество пользователей [7].

Хранение всей информации в базе данных позволит web – приложению оставаться гибким, безусловно это качественно структурирует информацию, делает ее более упорядоченной. Сайт становится более гибким и благодаря тому, что можно извлекать контент так как нам угодно мы можем его структурировать по-разному, выводя где-либо отдельными блоками. Удобно формировать список материалов, и это поможет в дальнейшем избежать бесконечного роста страниц. Для этих целей наилучшим выбором будет СУБД MySQL, она является одной из самых быстрых, высокопроизводительных и относительно простых СУБД, которую значительно проще установить и администрировать, чем многие другие

большие системы, также система является бесплатной. MySQL представляет систему управления реляционными базами данных [2]. На сегодняшний день это одна из самых популярных систем управления базами данных по итогам опроса сайта tagline.ru (сформирован на основе анкетирования, которое проводилось с августа 2014 по апрель 2016 года, 390 digital-агентств с продакшном и/или клиентским офисом в России) [10].

Для взаимодействия с базой данных из языка PHP было решено использовать библиотеку PDO. Она представляет собой расширение языка PHP для быстрого доступа к базам данных, доступно с PHP 5.1. PDO определяет простой и согласованный интерфейс для доступа к базам данных в PHP. Каждый драйвер базы данных, в котором реализован этот интерфейс, может представить специфичный для базы данных функционал в виде стандартных функций расширения. Достоинством данной библиотеки является простота составления запроса, а также расширение может поддерживать любую систему управления базами данных, для которой существует PDO-драйвер что позволяет менее болезненно заменить СУБД MySQL на любую другую при необходимости [12].

В качестве среды разработки использовалась Visual Studio Code. Данный редактор кода является легковесным создан компанией Microsoft. Основным достоинством является то, что это легковесная среда разработки с широким функционалом и практически не требующая настройки IDE прямо из коробки. Для полноценной работы необходимо установить лишь пару расширений такие как:

- PHP IntelliSense;
- PHP Debug.

PHP Debug является одним из самых полезных расширений позволяющее проводить поэтапное тестирование кода (по шагам).

## **2.2. Описание модели данных**

Для создания системы нам потребуется база данных.

Схема на рисунок 8 отражает модель данных, реализованную в разрабатываемом веб приложении. В таблицах 1 – 5 показаны свойства сущностей модели данных.

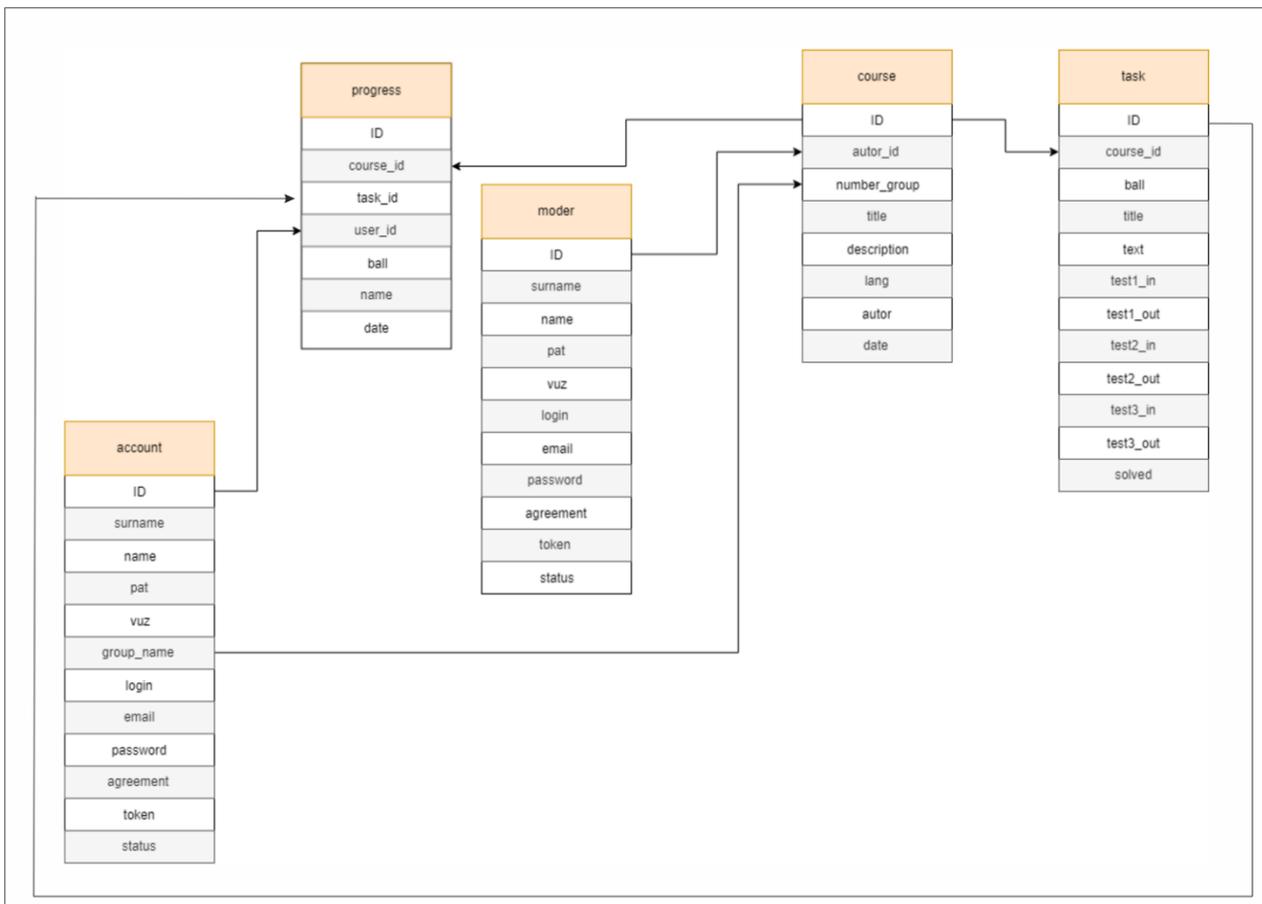


Рис. 8. Схема модели данных.

Для хранения информации о студентах нам понадобится таблица

1.

Свойства полей «account»

Таблица 1

Имя поля	Тип данных
ID	AUTO INCREMENT
surname	text
name	text
pat	text
vuz	text
group_name	text

login	text
email	text
password	text
agreement	boolean
token	text
status	boolean

Для хранения информации о преподавателях нам понадобится таблица 2.

Свойства полей «moder»

Таблица 2

<b>Имя поля</b>	<b>Тип данных</b>
ID	AUTO INCREMENT
surname	text
name	text
pat	text
vuz	text
login	text
email	text
password	text
agreement	boolean
token	text
status	boolean

Также необходимо будет хранить список курсов таблица 3.

Свойства полей «course»

Таблица 3

<b>Имя поля</b>	<b>Тип данных</b>
ID	AUTO INCREMENT

autor_id	int
number_group	text
title	text
description	text
lang	text
autor	text
date	datetime

Для каждого курса необходимо будет хранить список заданий для этого используем таблицу 4.

#### Свойства полей «task»

Таблица 4

<b>Имя поля</b>	<b>Тип данных</b>
ID	AUTO INCREMENT
course_id	int
ball	double
title	text
text	text
test1_in	text
test1_out	text
test2_in	text
test2_out	text
test3_in	text
test3_out	text
solved	int

Что бы отслеживать прогресс по каждому курсу и заданию в этом курсе необходимо создать таблицу 5.

#### Свойства полей «progress»

Таблица 5

<b>Имя поля</b>	<b>Тип данных</b>
ID	AUTO INCREMENT
course_id	int
task_id	int
user_id	int
ball	text
name	text
date	datetime

### **2.3. Описание реализации основных функций системы**

Разрабатываемая информационная система представляет собой web – приложение с базой данных MySQL. В качестве логики использовался PHP Framework разработанный по паттерну MVC. Для компиляции кода используется онлайн компилятор Rextester, а для подцветки синтаксиса, JS модуль CodeMirror [8, 15]. В качестве средства связи базы данных и PHP Framework используется PHP модуль PDO. Для разработки пользовательского интерфейса используется язык разметки HTML [13], а для оформления внешнего вида страниц используется CSS, все это работает на базе HTML, CSS и JS фреймворка Bootstrap разработанного компанией Twitted, что позволяет писать адаптивную кроссбраузерную верстку [6].

## **Вывод по главе 2**

Во второй главе было представлено обоснование инструментальных средств для реализации системы. В качестве инструментов реализации были использованы MVC Framework написанный на языке PHP, для сервера баз данных использовался MySQL, для взаимодействия с базой данных из PHP была выбрана библиотека PDO, так же для компиляции кода используется API сервиса Rextester, а для динамической отправки данных на сервер и динамики сайта был использован JavaScript. В качестве среды разработки был использован редактор исходного кода Visual Studio Code.

Было представлено описание модели баз данных, описание таблиц и свойства полей. Также была представлена схема основных функций системы.

## **Глава 3. ДОКУМЕНТИРОВАНИЕ ПОДСИСТЕМЫ**

### **3.1. Испытание информационной системы**

После завершения создания основных функций системы необходимо провести ее полное тестирование. Данный этап необходим для выявления всех недостатков системы. Тестирование – это, оценка разрабатываемого программного продукта, чтобы проверить его возможности, способности и соответствие ожидаемым результатам.

Испытание системы проходило в два этапа.

Первый этап функциональное тестирование. Под функциональным тестированием понимается проверка соответствия программного продукта функциональным требованиям, указанным в техническом задании на создание этого продукта. Если говорить проще, то при функциональном тестировании проверяется выполняет ли программный продукт все функции, которые должен. Было проведено тестирование отдельных компонентов программного продукта, сфокусированное на их специфике, назначении и функциональных особенностях. Проверялась стабильность работы всех моделей контроллеров, проверялись основные алгоритмы такие как регистрация студентов и преподавателей, добавление редактирование и удаление курсов, стабильность работы API интерфейса компилятора, отображение информации о статистике по курсу, редактирование пользователя, а также лояльность алгоритма при проверке заданий по программированию. По результатам тестирования было выявлено полное соответствие заявленным требованиям, а также полная стабильность работы системы.

Вторым этапом было нефункциональное тестирование. Оно оценивает такие качества программного продукта, как, например, эргономику или производительность. Тестирование проводилось на малой группе студентов (7 человек), проверялась скорость работы алгоритмов при нагрузке в несколько пользователей, а также удобство интерфейса. В результате тестирования не было выявлено просадок в скорости работы при нагрузке в

несколько пользователей, по результатам опроса был получен положительный результат об удобстве интерфейса [9].

### 3.2. Руководство пользователя

Для начала работы понадобится интернет-соединение и интернет браузер, в адресной строке необходимо будет ввести <http://b95178vy.beget.tech>, после загрузки страницы не авторизованный пользователь попадет на главную страницу с информацией о сайте рисунок 9

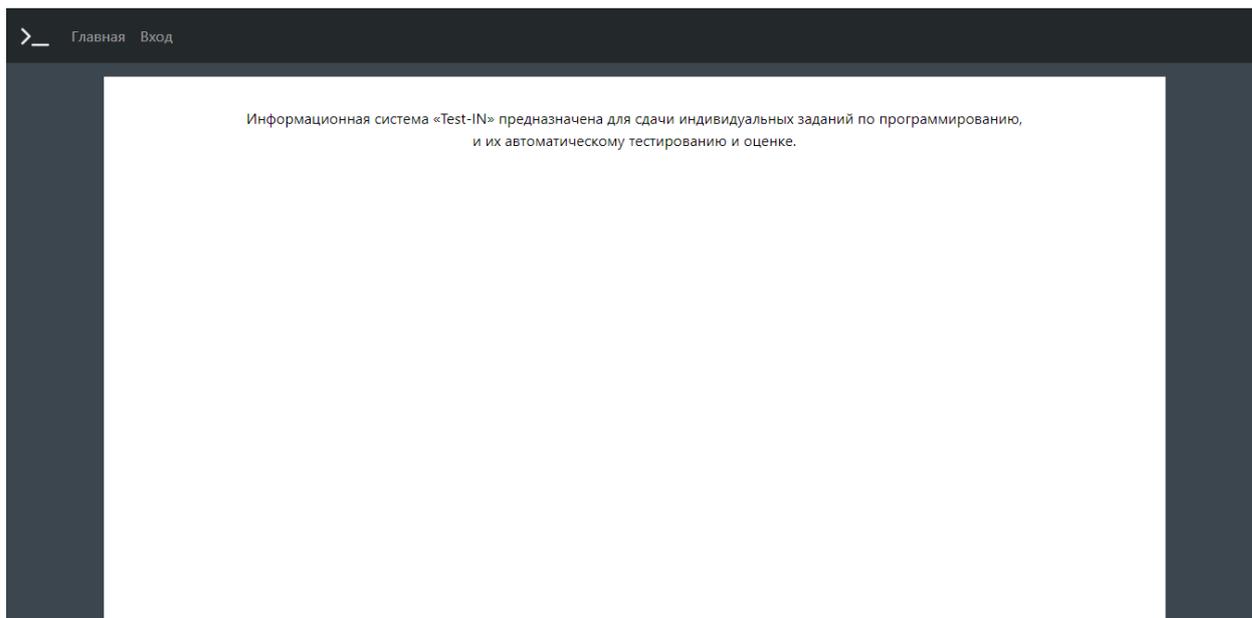
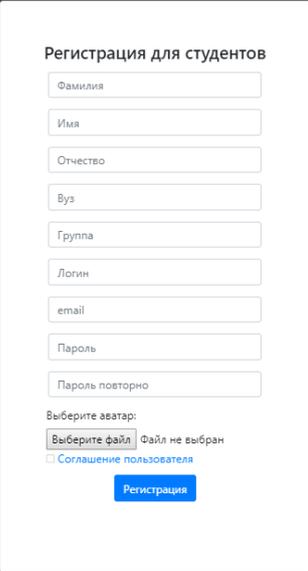


Рис. 9. Главная страница информационной системы

#### 3.2.1. Руководство пользователя для роли студент

Для доступа ко всем функциям информационной системы студенту необходимо пройти авторизацию. Если студент еще не зарегистрирован в системе ему необходимо пройти регистрацию заполнив форму на рисунке 10.



Регистрация для студентов

Фамилия

Имя

Отчество

Вуз

Группа

Логин

email

Пароль

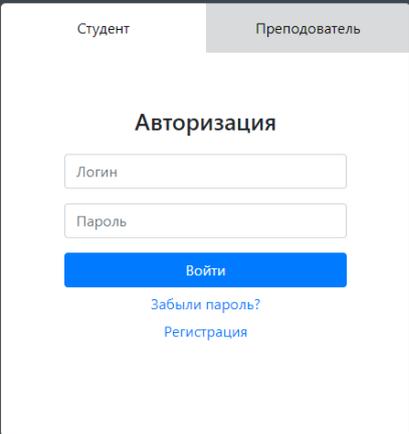
Пароль повторно

Выберите аватар:

[Соглашение пользователя](#)

Рис. 10. Форма регистрации студентов

Заполнив все поля и согласившись с пользовательским соглашением студент пройдет регистрацию и получит на почту письмо для подтверждения регистрации. Пройдя по ссылке студент попадет на страницу авторизации рисунок 11.



Студент  Преподаватель

Авторизация

Логин

Пароль

[Забыли пароль?](#)

[Регистрация](#)

Рис. 11. Форма авторизации для студентов

Пройдя авторизацию студент попадает на страницу с личными курсами, доступными студенту и всем его одногруппникам рисунок 12.

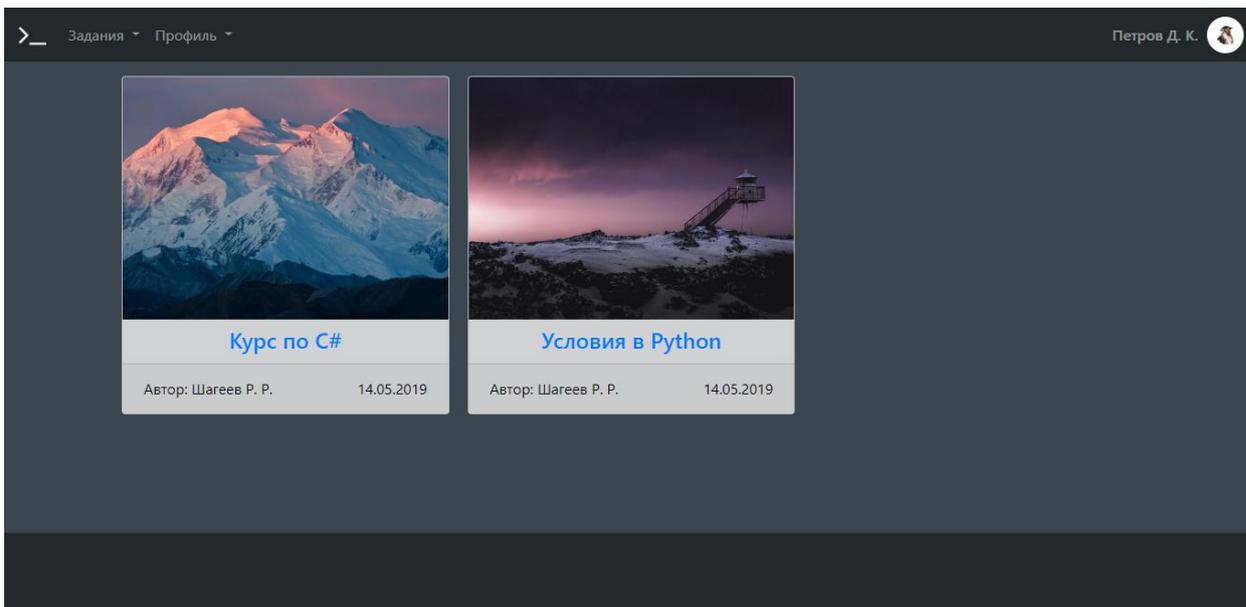


Рис. 12. Страница личных курсов

После авторизации пользователю становятся доступны новые пункты меню, такие как «Все курсы» содержит список всех курсов доступных на сайте рисунок 13. Так же раздел «Профиль» рисунок 14, в нем содержится информация о студенте, общий прогресс по всем личным курсам и по каждому курсу соответственно, также в разделе профиль доступна функция редактирования личной информации рисунок 15.

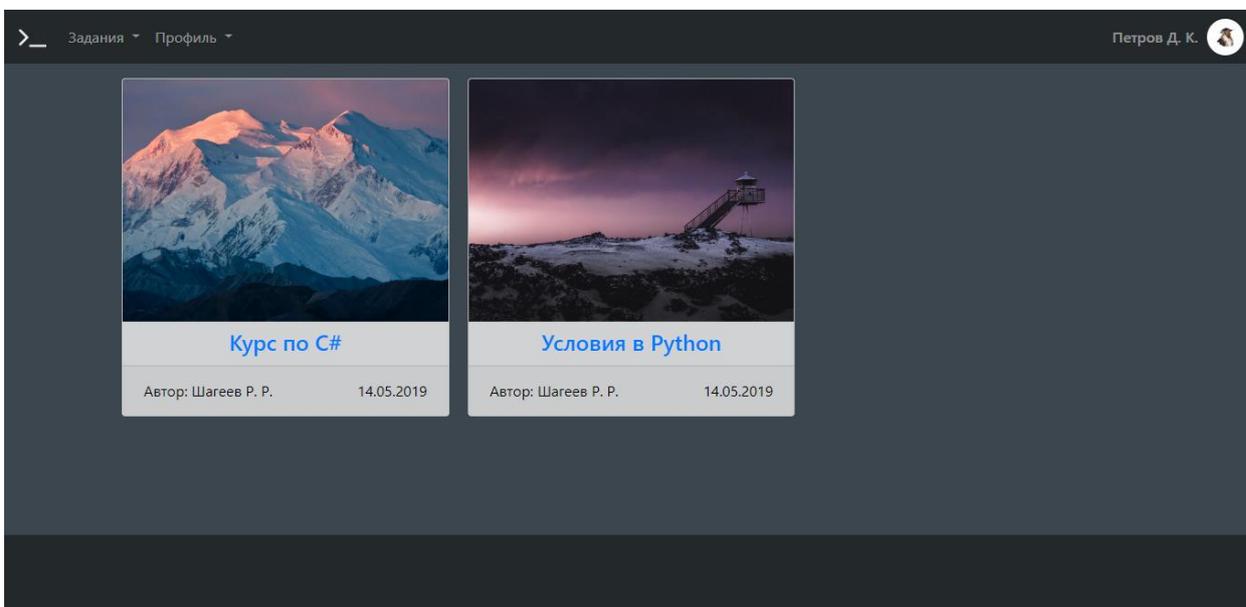


Рис. 13. Страница всех курсов

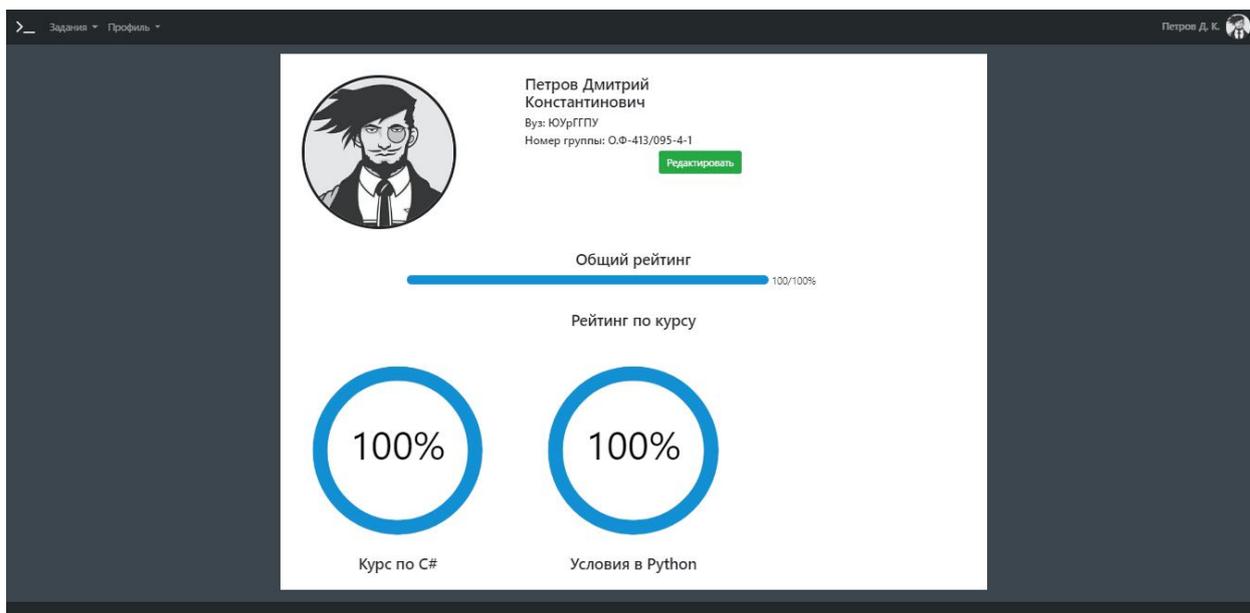


Рис. 14. Страница профиль студента

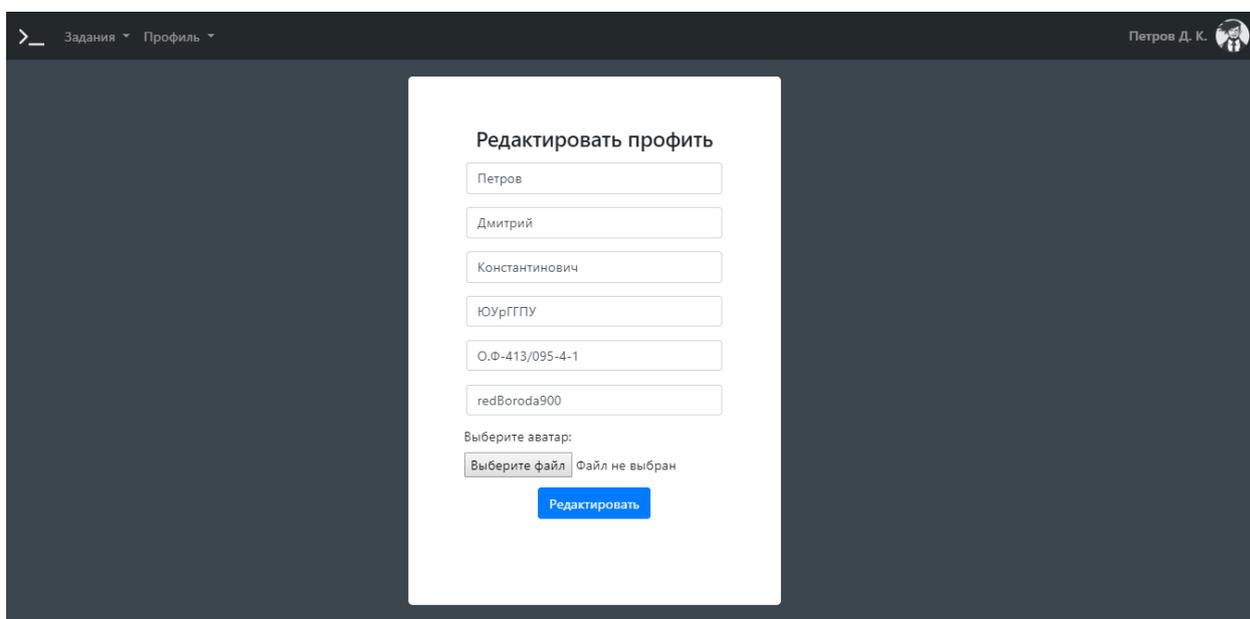


Рис. 15. Страница редактирования профиля

На вкладке задания в разделе личные студент может открыть один из курсов рисунок 16, в этом разделе студент познакомится с описанием курса, со списком всех заданий по курсу.

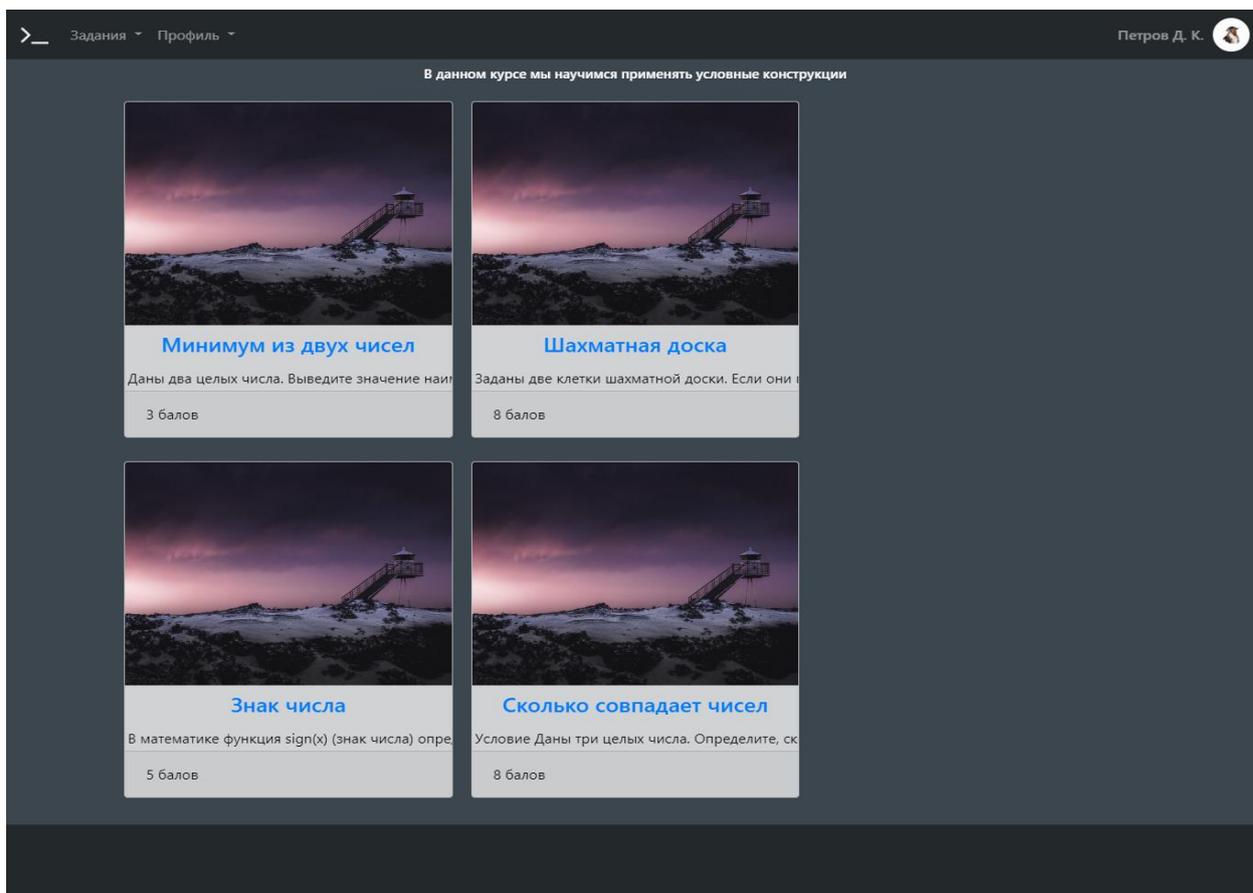


Рис. 16. Страница список заданий по курсу

Выбрав одно из заданий, студент попадает на страницу задачи рисунок 17, данная страница делится на три раздела. Первый это раздел с информацией о задаче (название, условие). Второй это окно редактора кода. Третий это список тестов.

### Задача «Минимум из двух чисел»

Условие:

Даны два целых числа. Выведите значение наименьшего из них. Во всех задачах считывайте входные данные через `input()` и выводите ответ через `print()`.

```
1 #python 3.5.2
2
3 x = int(input())
4 y = int(input())
5 if x > y:
6     print(y)
7 else:
8     print(x)
```

Входные данные:

Выходные данные:

Список ошибок

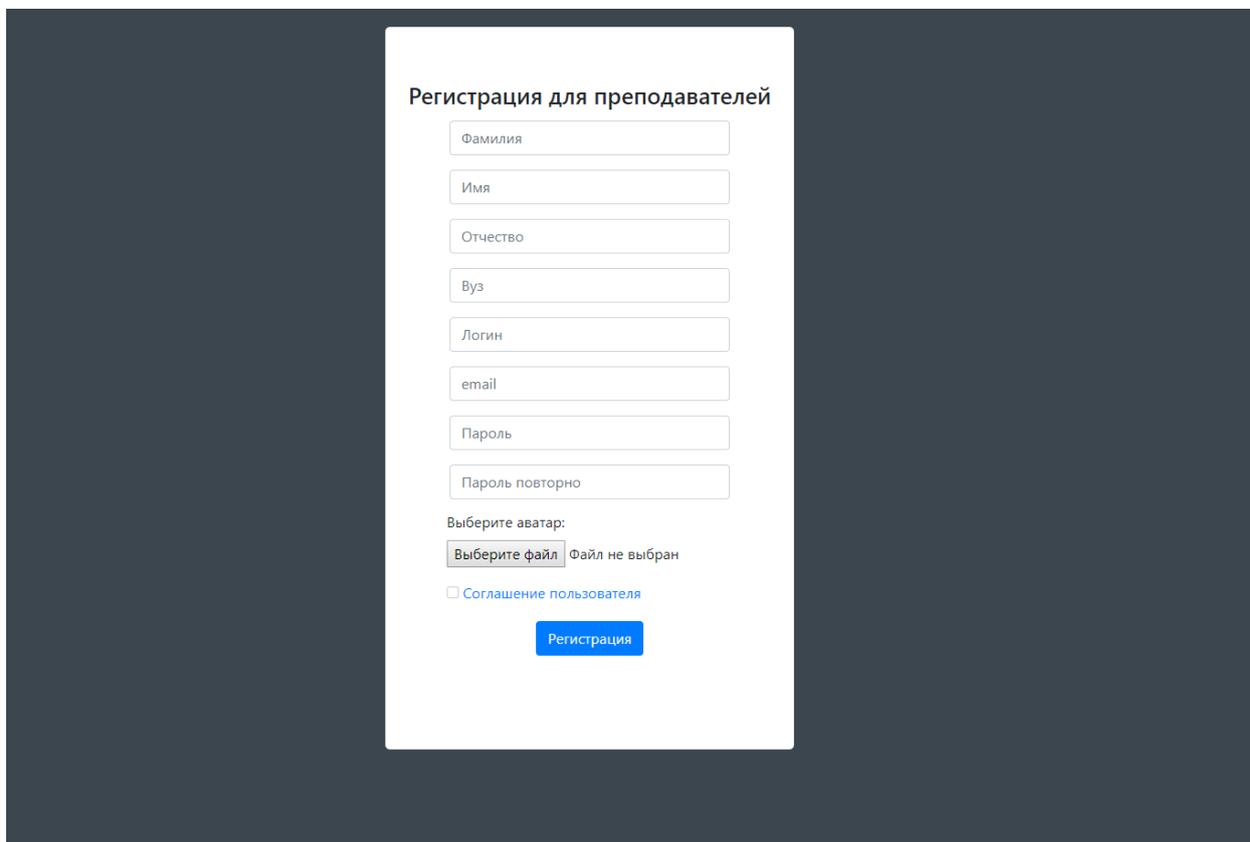
Проверить решение на всех тестах

№	Входные данные	Что вывела программа	Правильный ответ
Тест 1	3 7		3
Тест 2	2 2		2
Тест 3	-8 15		-8

Рис. 17. Страница задания

### 3.2.2. Руководство пользователя для роли преподаватель

Для доступа ко всем функциям информационной системы преподавателю необходимо пройти авторизацию. Если преподаватель еще не зарегистрирован в системе ему необходимо пройти регистрацию заполнив форму рисунок 18.



The image shows a registration form titled "Регистрация для преподавателей" (Registration for teachers). The form is centered on a dark background. It contains the following fields and elements:

- Фамилия (Surname)
- Имя (Name)
- Отчество (Patronymic)
- Вуз (University)
- Логин (Login)
- email
- Пароль (Password)
- Пароль повторно (Repeat password)
- Выберите аватар: (Choose avatar:)
- Выберите файл (Choose file) | Файл не выбран (File not selected)
- Соглашение пользователя (User agreement)
- Регистрация (Registration) button

Рис. 18. Форма регистрации преподавателей

Заполнив все поля и согласившись с пользовательским соглашением преподаватель пройдет регистрацию и получит на почту письмо для подтверждения регистрации. Пройдя по ссылке преподаватель попадет на страницу авторизации рисунок 19.

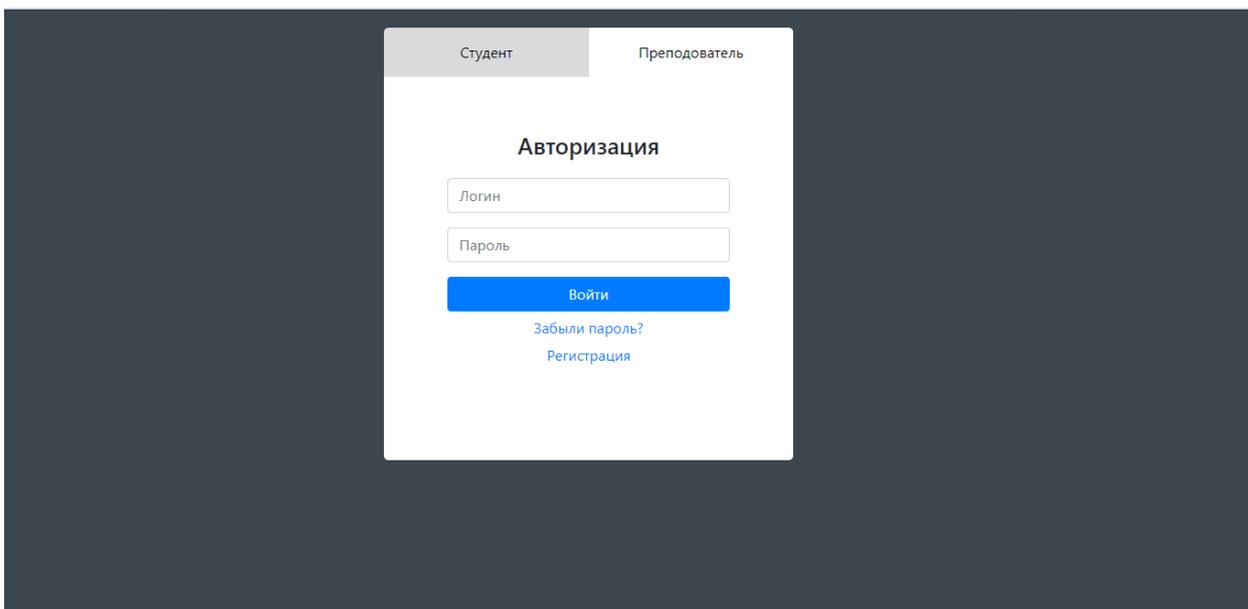


Рис. 19. Форма авторизации преподавателей

Пройдя авторизацию преподаватель попадает на страницу «Все курсы» где размещены все добавленные им курсы рисунок 20.

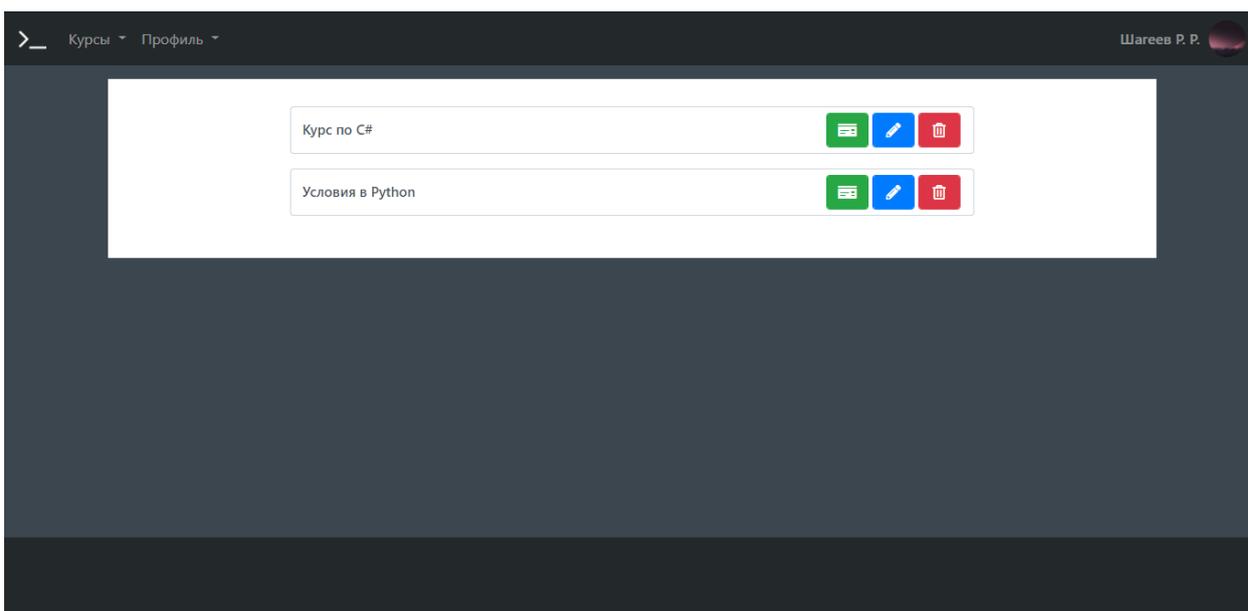


Рис. 20. Страница со списком всех курсов

Рядом с названиями курсов можно увидеть три кнопки (подробнее, редактировать, удалить). Нажав на кнопку подробнее, преподаватель попадает на страницу «статистика по задачам» рисунок 21. Кнопка редактировать открывает страницу редактирования курса рисунок 22. кнопка удалить удаляет курс.

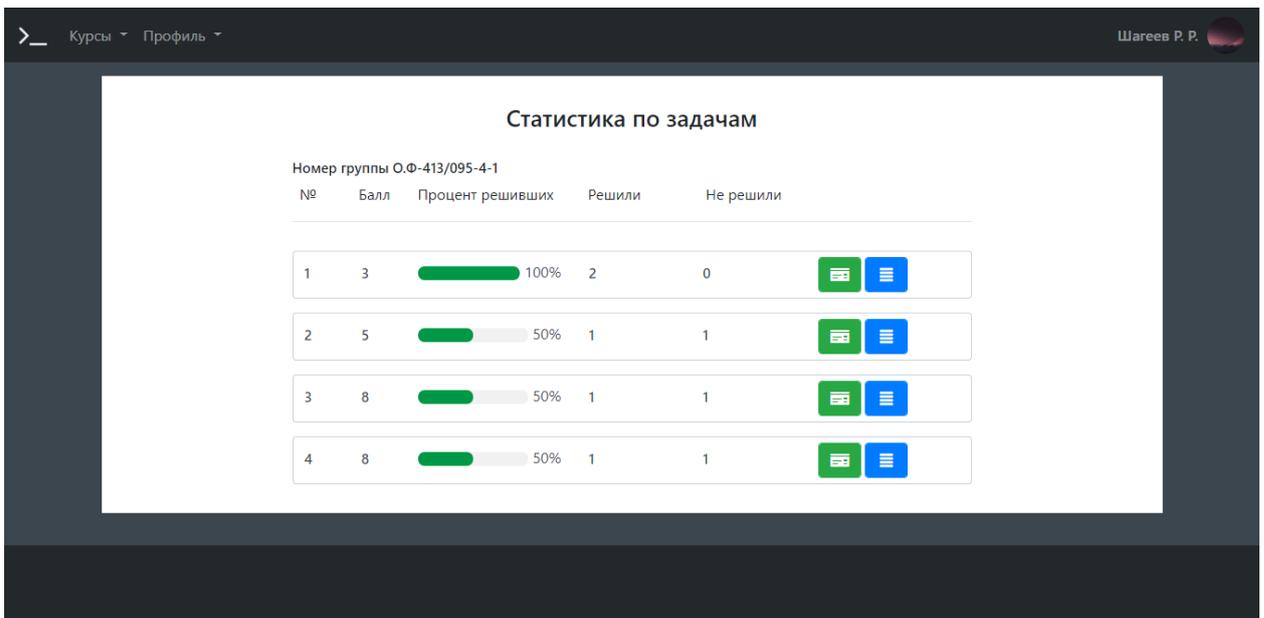


Рис. 21. Страница статистика по задачам

**Редактировать курс**

О.Ф-413/095-4-1

Курс по С#

Основы программирования на С#

Выберите язык

Выберите заставку:

Выберите файл | Файл не выбран

**Задание 1**

5

авыа

авывав

Выходные данные необходимо вводить через ENTER

**Тест 1**

Hello | Hello

**Тест 2**

Hello | Hello

**Тест 3**

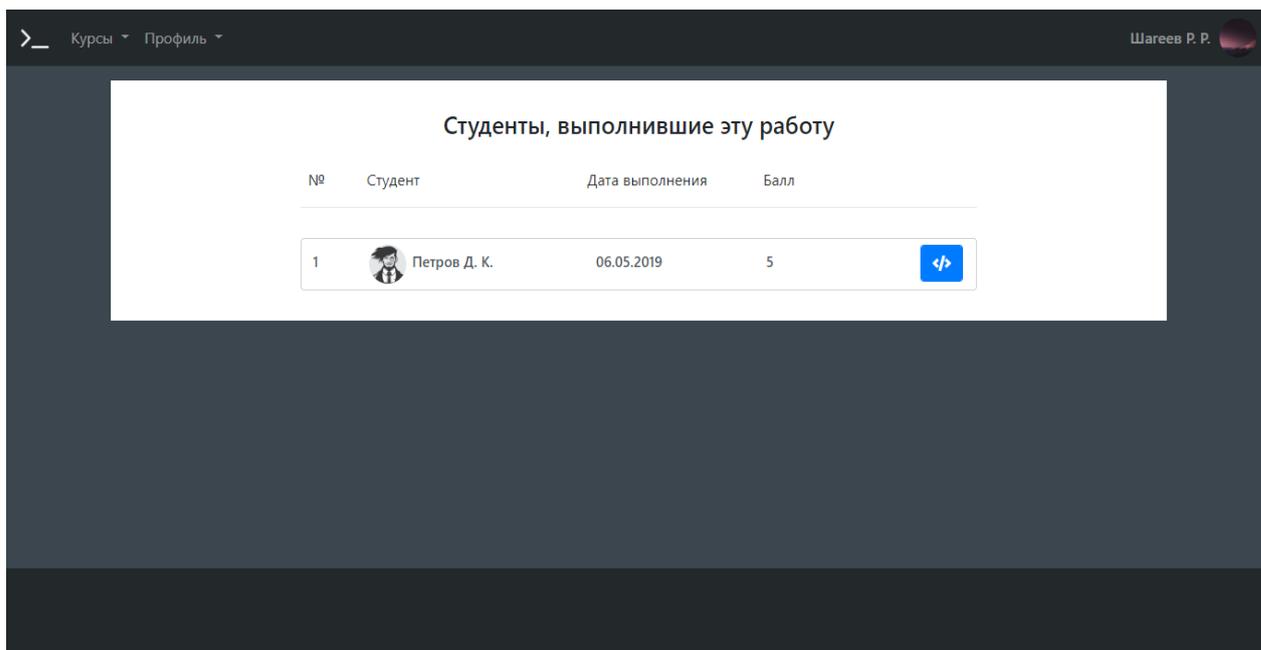
Hello | Hello

+ | 🗑️

Редактировать

Рис. 22. Страница редактировать курс

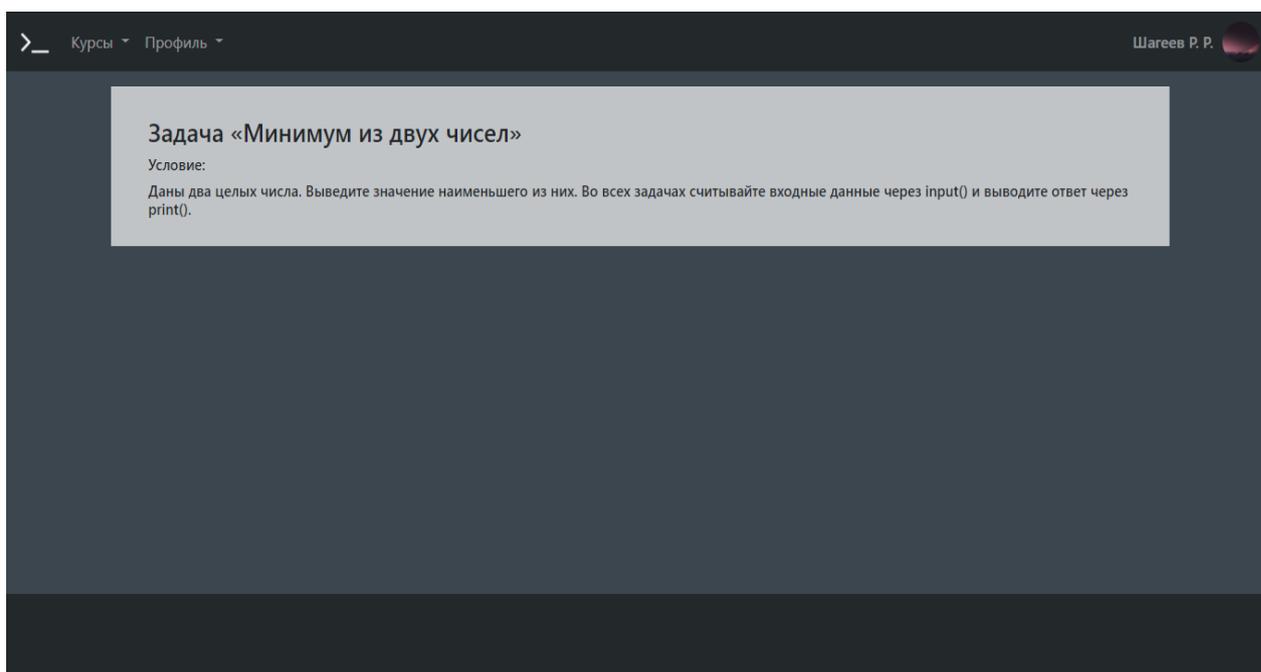
Перейдя на страницу «Статистика по задачам» преподаватель получает информацию о номере группы к которой прикреплен этот курс, номер задания, его балл, процент решивших студентов, количество решивших и не решивших, а также у каждого задания можно посмотреть список студентов которые решили это задание рисунок 23 и условие рисунок 24, нажав на соответствующую кнопку.



The screenshot shows a web interface with a dark header. On the left, there are navigation links: a home icon, 'Курсы', and 'Профиль'. On the right, the user's name 'Шареев Р. Р.' is displayed next to a profile picture. The main content area is titled 'Студенты, выполнившие эту работу'. Below the title is a table with the following data:

№	Студент	Дата выполнения	Балл	
1	 Петров Д. К.	06.05.2019	5	

Рис. 23. Страница студенты, выполнившие эту работу



The screenshot shows a web interface with a dark header. On the left, there are navigation links: a home icon, 'Курсы', and 'Профиль'. On the right, the user's name 'Шареев Р. Р.' is displayed next to a profile picture. The main content area is titled 'Задача «Минимум из двух чисел»'. Below the title, the condition is written:

Условие:  
Даны два целых числа. Выведите значение наименьшего из них. Во всех задачах считывайте входные данные через `input()` и выводите ответ через `print()`.

Рис. 24. Страница условие задачи

Перейдя на страницу со списком студентов, решивших задание, преподаватель получает информацию о фамилии студента дате решения и бал, который он получил, нажав на кнопку решение, можно посмотреть подробное решение задачи от студента рисунок 25.



```
1 #python 3.5.2
2
3 x = int(input())
4 y = int(input())
5 if x > y:
6     print(y)
7 else:
8     print(x)
```

Рис. 25. Страница решение

Нажав на кнопку редактирования курса рисунок 22, преподаватель может отредактировать курс.

Перейдя на вкладку курсы и выбрав пункт добавить преподаватель попадает на страницу добавления курса рисунок 26. Для добавления курса необходимо ввести номер группы, название, описание и выбрать язык программирования. Количество заданий в курсе не ограничено, добавить задание в курс можно нажав кнопку «+» удалить нажав кнопку удаления.

Курсы ▾ Профиль ▾ Шагеев Р. Р.

### Добавить курс

Номер группы

Название

Описание

Выберите язык ▾

Выберите заставку:

Выберите файл | Файл не выбран

### Задание 1

Вес задания

Название

Текст задания

Выходные данные необходимо вводить через ENTER

Тест 1

Входные данные | Выходные данные

Тест 2

Входные данные | Выходные данные

Тест 3

Входные данные | Выходные данные

+ | 🗑️

Добавить

Рис. 26. Страница добавить курс

### 3.3. Техничко-экономическое обоснование

Целью выполнения техничко-экономического обоснования является расчет затрат на разработку программного продукта. Разработка программного продукта требует трудовых затрат, приобретение расходных материалов, и ряда других прямых затрат. Расчет затрат на оплату труда представлен в таблице 6.

Таблица 6

№	Показатель	Единица измерения	Величина затрат
1	Затраты времени исполнителя на разработку	дни	30
2	Ставка дневной заработной платы исполнителя	Руб.	1300
3	Величина заработной платы исполнителя за выполнение разработки	Руб.	39000
4	Страховые взносы на обязательное пенсионное страхование (22%)	Руб.	8580
5	Страховые взносы на обязательное социальное страхование (ОСС) на случай временной нетрудоспособности и в связи с материнством (2,9%)	Руб.	1131
6	Страховые взносы на обязательное медицинское страхование (ОМС) (5,1%)	Руб.	1989
Суммарные затраты на оплату труда с учетом страховых взносов:			50700

Расчет затрат на материалы, приобретаемые для выполнения разработки представлен в таблице 7

Таблица 7

№	Показатель	Единица измерения	Сумма затрат,
---	------------	-------------------	---------------

			<b>руб</b>
1	Бумага	Руб.	300
2	Ручка шариковая	Руб.	12
3	Скоросшиватель	Руб.	60
Итого:			372

Расчет прочих прямых затрат представлен в таблице 8.

Таблица 8

<b>№</b>	<b>Показатель</b>	<b>Единица измерения</b>	<b>Количество</b>	<b>Сумма затрат на единицу</b>	<b>Сумма затрат за месяц</b>
1	Интернет	Месяц	2	430	860
Итого:					860

Расчет полной себестоимости на разработку программного продукта представлен в таблице 9

Таблица 9

<b>№</b>	<b>Показатель</b>	<b>Единица измерения</b>	<b>Сумма затрат</b>
1	Суммарные затраты на оплату труда с учетом страховых взносов	Руб.	50700
2	Расходы на материалы	Руб.	372
3	Прочие затраты	Руб.	860
4	Итого себестоимость	Руб.	51932
5	Средний уровень прибыльности (рентабельности) проектов разработки программных продуктов	%	25

6	Планируемая прибыль	Руб.	12983
7	Цена разработки программного продукта	Руб.	64915

Таким образом, цена разработки программного продукта составила 64915 рублей.

### **Вывод по главе 3**

В результате выполнения работ была спроектирована и реализована информационная система «Test-in», которая может использоваться на кафедре «Информатики, информационных технологий и методики обучения информатике» (ИИТиМОИ). Были приведены и описаны испытания информационной системы, в ходе тестирования удалось проверить корректность алгоритмов, отвечающих за регистрацию, авторизацию и работу в автоматизированной информационной системе.

Также для автоматизированной информационной системе было разработано подробное иллюстрированное руководство пользователя, в котором подробно расписаны все основные процессы АИС.

В рамках данной главы было приведено технико-экономическое обоснование, которое достаточно подробно демонстрирует расчеты по затратам и прибыли, позволяющие оценить финансовые расходы на разработку приложения в коммерческой среде.

## ЗАКЛЮЧЕНИЕ

Главным результатом квалификационной работы является разработанная ИС «Test-in» обеспечивающая быстрый доступ и проверку заданий по программированию.

В ходе выполнения работы был проведен анализ других продуктов на предмет выявления аналогичной системы, однако не одна из проанализированных систем не удовлетворяет всем требованиям.

В ходе выполнения работ была проанализирована и описана организационная структура сдачи заданий по программированию на кафедре «Информатики, информационных технологий и методики обучения информатике» (ИИТиМОИ). Для описания бизнес – процессов использовалась методология моделирования IDEF0.

Были обозначены цели создаваемой информационной системы. Также было представлено обоснование выбора средств разработки. Было решено использовать собственный Framework написанный на языке PHP, а также API сервиса Rextester для проведения компиляции алгоритмов, написанных студентами. В качестве базы данных использовался MySQL.

В ходе выполнения работы поставленные цели достигнуты, задачи выполнены в полном объеме.

## БИБЛИОГРОФИЧЕСКИЙ СПИСОК

1. Алексеев, В. А. Коммутируемые локальные сети Ethernet [Электронный ресурс] : методические указания к проведению лабораторных работ по курсу «Сети ЭВМ и телекоммуникации» / В. А. Алексеев. — Электрон. текстовые данные. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2010. — 31 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/17714.html>
2. Введение в СУБД MySQL [Электронный ресурс] / . — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 228 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73650.html>
3. Грекул, В. И. Проектирование информационных систем. Курс лекций [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по специальностям в области информационных технологий / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. — Электрон. текстовые данные. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 303 с. — 978-5-4487-0089-7. — Режим доступа: <http://www.iprbookshop.ru/67376.html>
4. Зудилова, Т. В. Web-программирование JavaScript [Электронный ресурс] / Т. В. Зудилова, М. Л. Буркова. — Электрон. текстовые данные. — СПб. : Университет ИТМО, 2012. — 68 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/65749.html>
5. Кафедра информатики, информационных технологий и методики обучения информатике [Электронный ресурс] // Режим доступа: [http://www.cspu.ru/o\\_cspu/kafedri/fakultetskie/informatiki-informatsionnykh-tekhnologiy-i-metodiki-obucheniya-informatike/](http://www.cspu.ru/o_cspu/kafedri/fakultetskie/informatiki-informatsionnykh-tekhnologiy-i-metodiki-obucheniya-informatike/)
6. Кириченко, А. В. Динамические сайты на HTML, CSS, Javascript и Bootstrap. Практика, практика и только практика [Электронный ресурс] / А. В. Кириченко, Е. В. Дубовик. — Электрон. текстовые данные. — СПб. :

Наука и Техника, 2018. — 272 с. — 978-5-94387-763-6. — Режим доступа: <http://www.iprbookshop.ru/77578.html>

7. Кисленко, Н. П. Интернет-программирование на PHP [Электронный ресурс] : учебное пособие / Н. П. Кисленко. — Электрон. текстовые данные. — Новосибирск : Новосибирский государственный архитектурно-строительный университет (Сибстрин), ЭБС АСВ, 2015. — 177 с. — 978-5-7795-0745-5. — Режим доступа: <http://www.iprbookshop.ru/68769.html>

8. Перепелица, Ф. А. Разработка интерактивных сайтов с использованием jQuery [Электронный ресурс] / Ф. А. Перепелица. — Электрон. текстовые данные. — СПб. : Университет ИТМО, 2015. — 144 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/68076.html>

9. Поляков, Е. А. Web-дизайн [Электронный ресурс] : учебное пособие / Е. А. Поляков. — Электрон. текстовые данные. — Саратов : Вузовское образование, 2019. — 188 с. — 978-5-4487-0489-5. — Режим доступа: <http://www.iprbookshop.ru/81868.html>

10. Рейтинг систем управления базами данных (СУБД) 2016 [Электронный ресурс] // Режим доступа: <https://tagline.ru/database-management-systems-rating/>

11. Рейтинг языков программирования 2018 [Электронный ресурс] // Режим доступа: <https://tagline.ru/programming-languages-rating/>

12. Руководство по PHP [Электронный ресурс] // Режим доступа: <https://www.php.net/manual/ru/>

13. Савельев, А. О. HTML 5. Основы клиентской разработки [Электронный ресурс] / А. О. Савельев, А. А. Алексеев. — Электрон. текстовые данные. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 286 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/57369.html>

14. Rextester API [Электронный ресурс] // Режим доступа: <https://rextester.com/main>

15. CodeMirror [Электронный ресурс] // Режим доступа:  
<https://codemirror.net>

16. jQuery API [Электронный ресурс] // Режим доступа:  
<https://api.jquery.com>

17. Unleash the power behind Ideone. Ideone.com - Online Compiler and  
IDE [Электронный ресурс] // Режим доступа: <https://ideone.com/sphere-engine>