

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«**Приамурский государственный университет имени Шолом-Алейхема**»

ФАКУЛЬТЕТ Математики, информационных технологий и техники
КАФЕДРА Информационных систем, математики и правовой информатики
Направление 44.03.05 Педагогическое образование: Информатика и математика

ВКР ДОПУЩЕНА К ЗАЩИТЕ
Заведующий кафедрой
_____ Р.И. Баженов
« ____ » . _____ .2020 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)
на тему «Разработка автоматизированной системы по обучению
программированию»

Студент 5 курса
А.О. Кизянов
Научный руководитель
к.п.н., доцент Баженов Р.И.

ВКР защищена с оценкой _____
Технический секретарь ГАК
_____ П.А. Козич

г. Биробиджан, 2020 год

Оглавление

Введение.....	3
1. Методы обучения программированию	6
1.1 Традиционные методы обучения	6
1.2 Автоматизированные системы обучения	12
1.3 Обзор используемых технологий.....	19
2. Создание автоматизированной системы обучения программированию.....	21
2.1 Структура системы	21
2.2 Разработка структуры базы данных.....	22
2.3 Руководство пользователя	28
2.4 Структура курса по изучению языка программирования C++ в рамках дистанционного обучения.....	42
2.5 Педагогический эксперимент по использованию автоматизированной системы обучению программирования	47
Заключение	54
Библиографический список	55

Введение

На данный момент нет ни одной деятельности, где бы ни использовали ЭВМ с её многочисленными функциями и возможностями. Образование она также не обошла. Современная система образования с помощью информационных технологий является открытой и доступной для получения образовательных услуг. Новые информационные технологии предоставляют средства для более эффективного планирования учебного процесса, использования различных источников и видов информации, мобильности и открытости содержания обучения.

Образовательная деятельность, как и прочие сферы деятельности человека, подвержена изменениям, современным тенденциям. Одной из таких тенденций является стремление полного или частичного замещения человека автоматикой. Высокие технологии внедряются, расширяют спектр возможностей донесения учебного материала, форм и методов контроля знаний, увеличивают эффективность процесса проверки заданий учащихся. Одним из сравнительно новых методов контроля знаний стал машинный контроль – тестирование с автоматизированной системой проверки результатов. Однако классическое тестирование не предполагает проверки знаний и умений за рамками формы вопрос-ответ или выбора ответа из предложенных вариантов. Классическое тестирование, даже перенесенное в электронную среду, до недавнего времени не было способно оценить профессиональные навыки и умения по предмету тестирования. Однако для будущих специалистов в сфере программирования такая возможность есть – это системы автоматизированного тестирования решений задач по программированию.

В этой работе будет создана автоматическая система проверки заданий по программированию, в которой преподаватель сможет формировать курсы, разделы и задания, а студент проходить их дистанционно с автоматической проверкой заданий.

Актуальность задачи определяем в автоматизации процесса проверки

заданий по программированию, решения которых могут отличаться друг от друга, что не дает возможности простого сравнения с эталонным решением. Например, одна и та же задача может быть решена на разных языках программирования и разными способами, так что единственным решением будет «прогон» решения задания по заготовленным тестам для выявления правильности решения. Также, студентам будет приятно наблюдать за своим прогрессом.

Цель работы – разработать автоматизированную систему обучению программированию.

Объектом исследования является обучение программированию.

Предметом исследования является информационная система автоматизированного обучения программированию.

Для реализации поставленной цели необходимо выполнить следующие задачи:

1. Изучить методы традиционного обучения программированию.
2. Анализировать статьи, учебники и существующие среды автоматической проверки заданий по программированию.
3. Разработать информационную систему автоматизированного обучения программированию.
4. Провести педагогический эксперимент по апробации разработанной системы.

Методы и средства решения поставленной цели:

- изучение литературы по теме исследования;
- проектирование разработка программных систем;
- моделирование;
- педагогический эксперимент.

Основными практическими результатами являются интерактивная методика по обучению языкам программирования с автоматизированной проверкой заданий и возможностью создавать курсы с теоретической частью

и практической. Такую систему можно применять на уроках по обучению программированию.

Данная выпускная квалификационная работа состоит из трех частей. Первая часть аналитическая, где рассматриваются: традиционные методы преподавания программирования и анализ существующих систем.

Вторая глава проектная, где рассматриваются: разработка концепции построения информационной модели, разработка информационной системы, разработка структуры базы данных.

1. Методы обучения программированию

1.1 Традиционные методы обучения

В базовом курсе информатики термин программирование опирается на умения и навыки применения алгоритмизации.

Цели обучения программированию в базисном курсе информатики предполагает собой.

1. Понимание и представление автоматической обработке информации и данных.

2. Понимание структур программ и организации хранения данных в них.

Содержание такого курса опирается на следующие понятия.

1. Типы алгоритмических конструкций языка (линейная, разветвляющаяся, циклическая) и работа с ними на языке программирования.

2. Линейные программы. Понятия переменной, её имени и значения.

3. Разветвляющиеся программы. Понятие условия.

4. Графика в языке программирования. Графические компоненты такие как, точка, прямая, окружность и т.д.

5. Циклические алгоритмы. Понятие цикла. Построение программ на множественное повторение одних действий в зависимости от условия.

6. Вложенные циклы. Решение комбинаторных задач, игровые задачи.

7. Функциональные алгоритмы. Возможность множественного использования одного кода.

8. Операции со строками. Алгоритмы замены, вставки, обрезки и т.д.

9. Массивы. Возможность хранения структурированной информации и работа с ней.

Традиционная методика обучению программированию построена на решении по большей части математических задач средствами компьютера. Разделяют следующие этапы решения таких задач.

1. Изучение исходных данных.

2. Найти алгоритм решения задачи.

3. Подготовка языка программирования к работе с данными задачи.
4. Создание алгоритма на языке программирования и его отладка.
5. Анализ полученных результатов.

Главная цель при обучении решения задач с использованием языка программирования научить использовать универсальные приемы построения решения при различных ситуациях.

Основные требования к системе задач.

1. Для объяснения нового материала лучше подходят практические и интересные задачи. При решении новых задач следует использовать новые приемы и конструкции языка программирования. Они вводятся в том объеме, который необходим для решения данной задачи.

2. Для закрепления должны использоваться уже изученные конструкции языка, но иногда в модифицированном виде заодно отражающие новые аспекты своего применения. Тем самым показывая, что их использование может быть универсальным.

3. Для изучения типовых методов решения чаще задачи формулируются нечетко, и уже в беседе с учеником проясняются все нужные компоненты.

4. Для самостоятельного решения задачи подбираются разного уровня для закрепления и изучения материала.

Основные этапы работы над задачей.

1. Знакомство с текстом и обсуждение (выявление проблемы, выявление ассоциаций в практической деятельности).

2. Обсуждение различных подходов к решению задачи, разбиение на подзадачи, выбор методов решения, формализованное описание действий в соответствии с возможностью компьютера и языка программирования.

3. Построение первого варианта решения – частично формализованного и не обязательно верного (на естественном или алгоритмическом языке).

4. Определение, какие данные (переменные, константы, какого типа) потребуются. Введение обозначений. Описание основных и промежуточных переменных.

5. Запись на языке программирования фрагментов решения (наиболее трудные – обсуждаются фронтально учителем, остальные – учащиеся).

6. Сборка программы в целом.

7. Ручная отладка программы – проигрывание различных ситуаций, отлавливание явных ошибок.

8. Компьютерное тестирование программы (составить тесты – наборы исходных данных). Анализ результатов и коррекция.

9. Обсуждение возможностей применимости и модификации.

Написание программы решения задачи требует разработки модели её решения, учет различных условий протекания программируемого процесса – формируется системный подход к анализу процесса.

Поиск путей решения задачи в условиях ограничений (средства языка программирования, системы программирования, ресурсов компьютера) – развивает гибкость мышления и аналитические способности.

Запись алгоритма с помощью средств языка программирования – развивает навыки формализованного описания решения (формализации) и формально-логическое мышление.

Редко, когда первый вариант программы оказывается идеальным. Многочисленные синтаксические, семантические и логические ошибки требуют организации системной деятельности по поиску и устранению ошибок – формируются навыки рефлексивной деятельности (компьютер помогает конкретизировать мышление).

Научение программированию изменяет способ, с помощью которого дети изучают учебные предметы, изменяет понятие задачи, подходы к решению задач.

Изучение программирования остается «классикой» информатики, однако, основным является не программа как таковая, а задача, которая решается с компьютером этой программой.

И здесь основным является модельный подход, поскольку алгоритм и программа – это модели деятельности субъекта по решению задачи.

Язык программирования и система программирования – это средства общения с компьютером (исполнителем), которому мы поручаем решение задачи (средства автоматизации информационного процесса обработки информации). В процессе обучения программированию удастся организовать поисковую, исследовательскую деятельность по решению проблем.

Для анализа учебников по информатике на то, как они вводят тему «Алгоритмизация и программирование» были отобраны три учебника.

Первый учебник «Информатика 9кл. А.В. Горячев, С.Л. Островский» [4]. Учебник представлен в 2 частях, и первая часть сразу начинается с темы «Алгоритмизация и программирование» первые шаги были с системы счисления и работы с ней, далее была работа с символьным типом данных на языке программирования Паскаль. Составление программ и работа с функциями ввода и вывода. В целом обучение направлено на решение заданных проблем. Другими словами, дается проблема и с помощью языка программирования решается.

Следующий учебник – это «Информатика и ИКТ 9кл Л.Л. Босова, А.Ю. Босова» [2]. Учебник представлен также в 2 частях и в 1 части впервые идет тема «Основы алгоритмизации». Первыми темами идет понятие алгоритма, свойства алгоритма. В отличие от предыдущего учебника дальше идут темы исполнителя и характеристики исполнителя.

И последний учебник «Информатика 10 класс Н.Д. Угринович, Базовый уровень» [3]. В данном учебнике тема «Алгоритмизация» начинается с 4 главы и сразу рекомендует установить такие системы как Lazarus и Microsoft Visual Studio. Начитается, как и в предыдущем учебнике с понятия алгоритма, но в отличии от него сразу переходит к блок схемам. У этого

раздела есть некая особенность, он сразу наполнен и понятием, и свойствами алгоритма, блок схемами описывающие эти алгоритмы и небольшие конструкции языка программирования Microsoft Basic .NET.

Из анализа учебников можно сделать вывод, что у каждого автора есть свое представление правильной подачи нового материала, кто-то начинает понятий и терминов, кто-то смешивает их сразу с блок схемами, а кто-то добавляет ещё туда конструкций языка.

Среди научных трудов есть статья Е.А. Конова и Г.А. Поллак про «Обучение программированию с использованием метода кейсов» [5]. В данной статье описывается сценарии работы с кейсами при обучении программированию. Подчеркивается наличие индивидуального подхода к каждому студенту с целью формирования у выпускника ряда общекультурных и профессиональных компетенций, предусмотренных общеобразовательным стандартом. Описывается методы обучения программированию, один из которых быстрое погружение студента в процесс составления программ: знакомство с основными объектами и концентрации языка программирования. Второй описываемый метод это ориентацией на алгоритмы предполагает первоначальное знакомство студентов с рядом понятий и конструкций теории алгоритмов, не зависящих от конкретного языка программирования и без учета эффективного исполнения программного кода. Студенты учатся анализировать предметную область с целью формализации поставленной задачи и выбора алгоритма ее решения.

Следующая статья М.С. Орлова «Модели смешанного обучения и их применение при обучении программированию» [6]. В ней описываются всевозможные подходы к определению смешенного обучения, большинство из которых носит описательный характер. Первый подход связан с пониманием смешанного обучения, как некоего формата учебных курсов, при котором в дистанционные курсы встраиваются активные методы обучения. В этом случае основной материал излагается в рамках

дистанционного курса, который предполагает самостоятельную работу учащегося; закрепление и отработка материала проходят на очных занятиях, реализуемых с использованием активных методов обучения. Второй подход рассматривает смешанное обучение, как модель использования распределенных информационно-образовательных ресурсов в очном обучении с применением элементов асинхронного и синхронного дистанционного обучения.

Элементы модели смешанного обучения, при их интеграции на уровне компонентов, должны размещаться в единой образовательной среде, одним из существенных свойств которой является предоставление обучающимся доступа к огромным массивам информации вместо препарированного знания учебников.

Третьей статьей является В.П. Гладков «Методы обучения программированию» [1]. В ней рассматриваются методы выработки качеств необходимых программисту. По словам автора, самые распространенные ошибки в обучении программированию являются: Обучение конкретному языку программирования, малое число решенных задач, отказ от знакомства с профессиональными работами для опыта. Также автор выделил набор качеств, которыми должен обладать программист:

1. Терпение.
2. Дисциплинированность.
3. Добросовестность.
4. Умение работать в коллективе.
5. Способность разобраться в сложной ситуации.
6. Аккуратность.
7. Усидчивость.

По словам автора, следование всем приведенным приемам в статье способствует хорошим результатам обучения, что подтверждает практика обучения профессиональных программистов в Пермском государственном техническом университете.

1.2 Автоматизированные системы обучения

Поскольку сейчас век современных технологий, то и обучение стало переходить на онлайн версию. Появляется множество онлайн школ по обучению чему угодно, в том числе и программированию. Несколько таких школ и проанализируем.

В основной массе такие школы представляет собой платформу с возможностью записаться (бесплатно или за деньги) на курс, и проходить его дистанционно либо через записанные лекции, либо через онлайн трансляции. После каждого крупного раздела существует домашняя работа, для каждого курса это может быть что-то своё, тесты, проекты, творческие задания и т.д. Некоторые из них проверяет сам преподаватель курса, а некоторые, такие как решение задачи на программирование можно проверить и автоматически.

Для сравнения платформ возьмем следующие критерии: доступность на русском языке, наличие бесплатных курсов, возможность создать свой собственный курс, возможность автоматической проверки заданий, сертификат о прохождении учебы, обратная связь.

Первая площадка для сравнения пойдет [udemy.com](https://www.udemy.com) [13]. Udemy – крупнейшая в мире платформа онлайн-курсов. Внешний вид главной страницы представлен на рисунке 1.1.

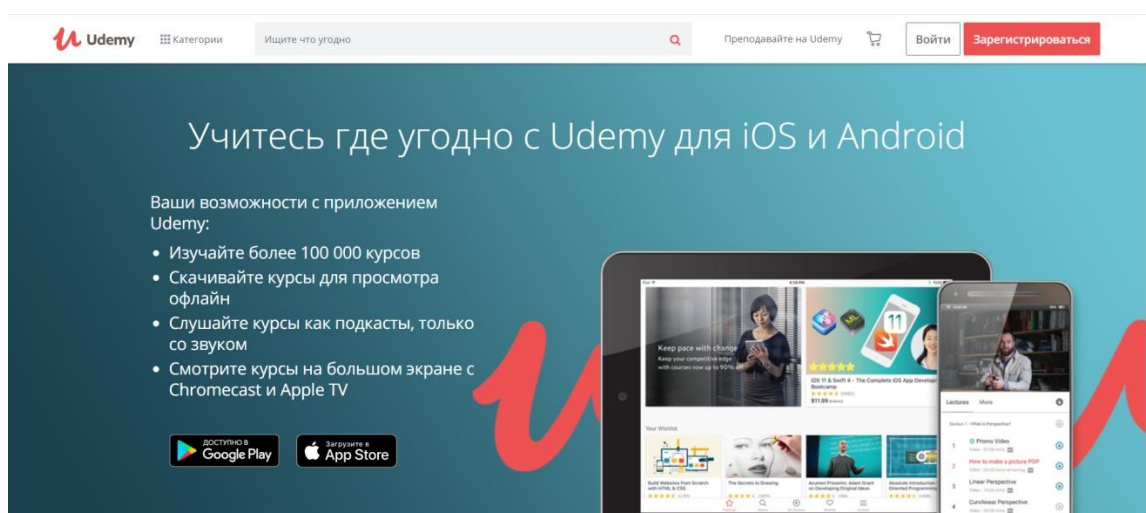


Рис. 1.1. Первая страница платформы Udemy.com

У платформы есть поддержка русского языка, нет автоматической проверки домашних заданий, по выпуску некоторые курсы дают сертификат, удобная обратная связь с преподавателем.

Следующая платформа, smotriuchis.ru [11], в отличие от предыдущей, создана в России и, следовательно, имеет русский язык по умолчанию. Вид первой страницы представлен на рисунке 1.2.

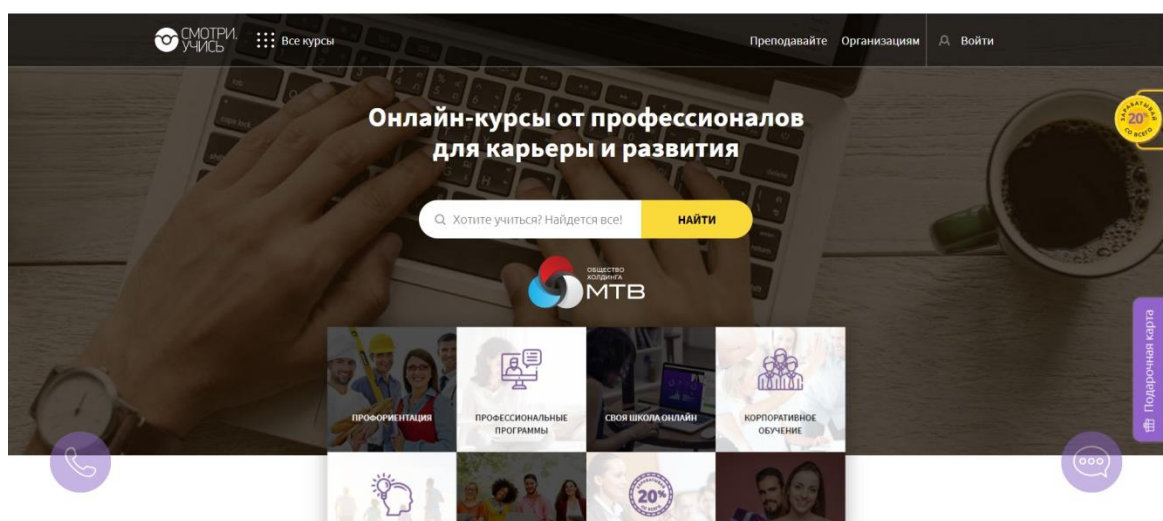


Рис. 1.2. Главная страница платформы smotriuchis.ru

Но также, как и предыдущая не имеет автоматической проверки домашних заданий, по выпуску сертификаты не даются, но есть удобная обратная связь с преподавателем.

Третья платформа это stepik.org [12]. Она, как и предыдущая создана в России и имеет поддержку русского языка изначально, но эта платформа позволяет переводить сайт ещё на несколько языков таких как английский, украинский и другие. Главная страница выглядит как на рисунке 1.3.

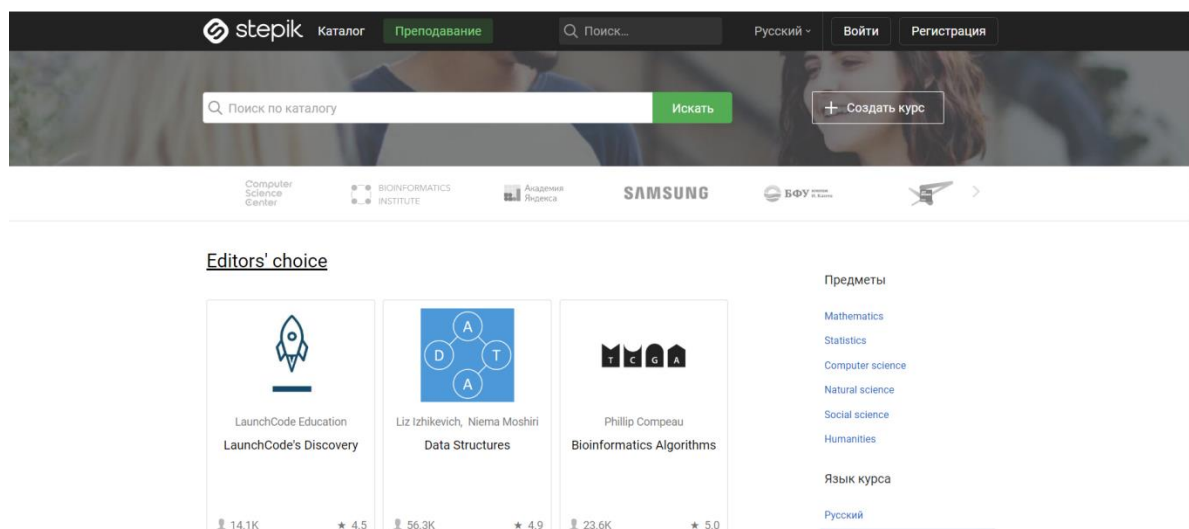


Рис. 1.3. Главная страница stepik.org

В отличие от других платформ, эта больше специализируется на информационных технологиях, поэтому здесь чаще можно увидеть курсы, посвященные языкам программирования. Возможно, как-раз из-за этого они сделали себе автоматическую систему проверки заданий. Платформа поддерживает практически все известные языки. Также преподаватели могут давать сертификаты по прохождении курсов. Обратная связь формируется через несколько каналов связи с преподавателем.

На рисунке 1.4 представлена главная страница проекта hexlet.

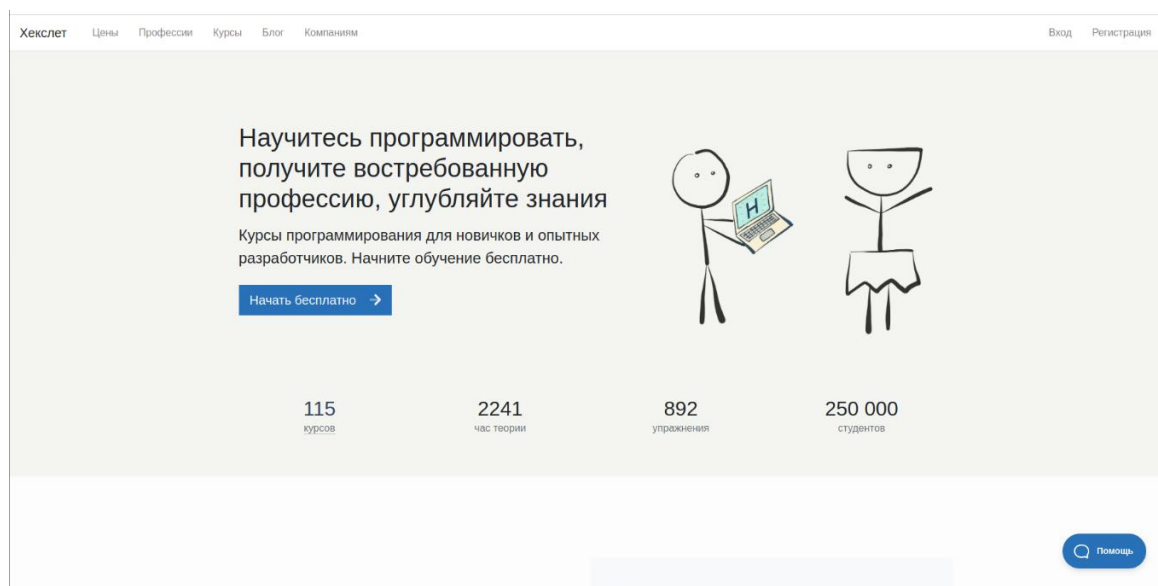


Рис. 1.4. Главная страница hexlet

Создатель платформы hexlet [10] считает, что настоящий программист должен понимать, как работает компьютер, и обладать вычислительным мышлением. Он должен видеть проблему, а не задание. Должен уметь анализировать и рассуждать на уровне проблемы и выше, а не только на уровне кода.

При сегодняшнем обилии учебных материалов, курсов и книг, главный вопрос, который встает перед новичком, это не «где учиться?», а «что и в каком порядке учить?». Существует множество мнений по этому поводу. Кто-то советует начинать с математики, кто-то советует конкретные языки и технологии.

Теория. Короткая лекция в виде текста или видео. Она основана на задаче, так что в отличие от традиционной академической теории, мы делаем упор на специфическую задачу, которая уже решена умными инженерами и программистами, а затем следуем по их шагам, чтобы понять решение.

Тест. Проверка вашего понимания. Нам не важно запоминание фактов, поэтому вопросы теста нацелены на понимание концепции, а не на тренировку памяти.

Практика. Упражнения по программированию в реальной среде разработки, доступны через браузер. Не в виде симуляции, не в виде игрушки, а на настоящей машине с базами данных, фреймворками, серверами и другими инструментами.

В процессе обучения вы сможете получать помощь ментора, а также других членов нашего сообщества: учеников, выпускников, менторов и преподавателей.

На рисунке 1.5 представлена главная страница проекта GeekBrains.

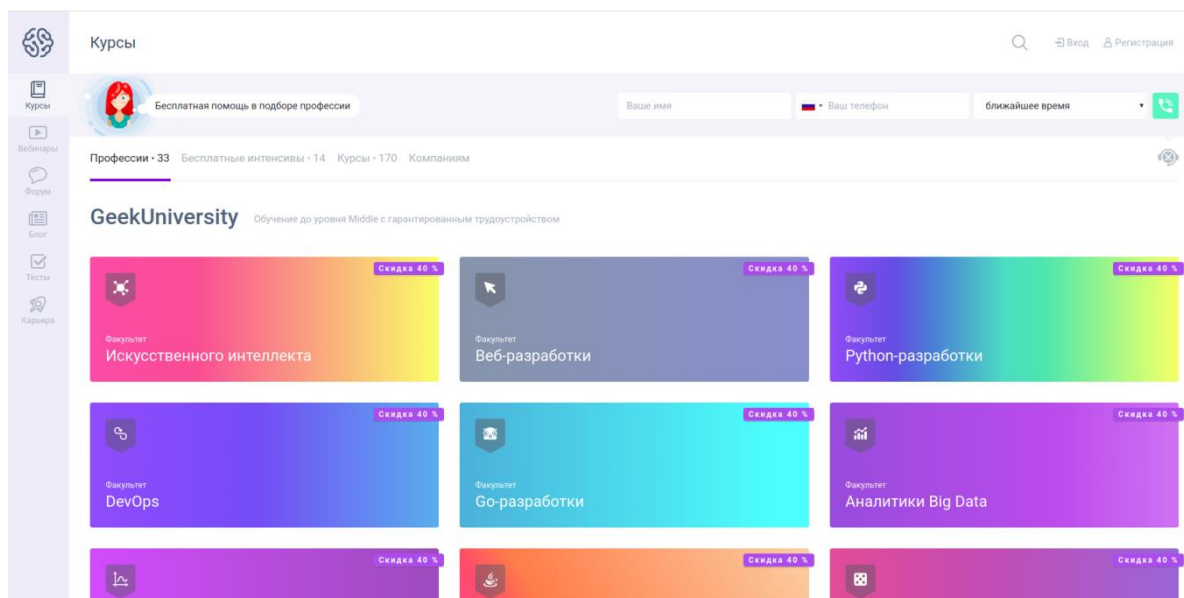


Рис. 1.5. Главная страница Geekbrains

Geekbrains [9] появился при наблюдении оторванности современной системы образования от реальных потребностей рынка труда. Так в 2010г. появился проект «Школа программирования», которые впоследствии вырос в сообщество GeekBrains, а с августа 2016 года вошел в состав компании Mail.ru Group. Сегодня уже более 3 млн. пользователей, и с каждым новым участником проект становится еще более полезнее.

По заверению портала они создают современную систему образования. Верят, что когда-нибудь получение полноценной профессии через интернет станет новой реальностью для человечества. Ведь это открывает безграничные возможности для амбициозных и энергичных людей.

На рисунке 1.6 представлена главная страница проекта coursera.

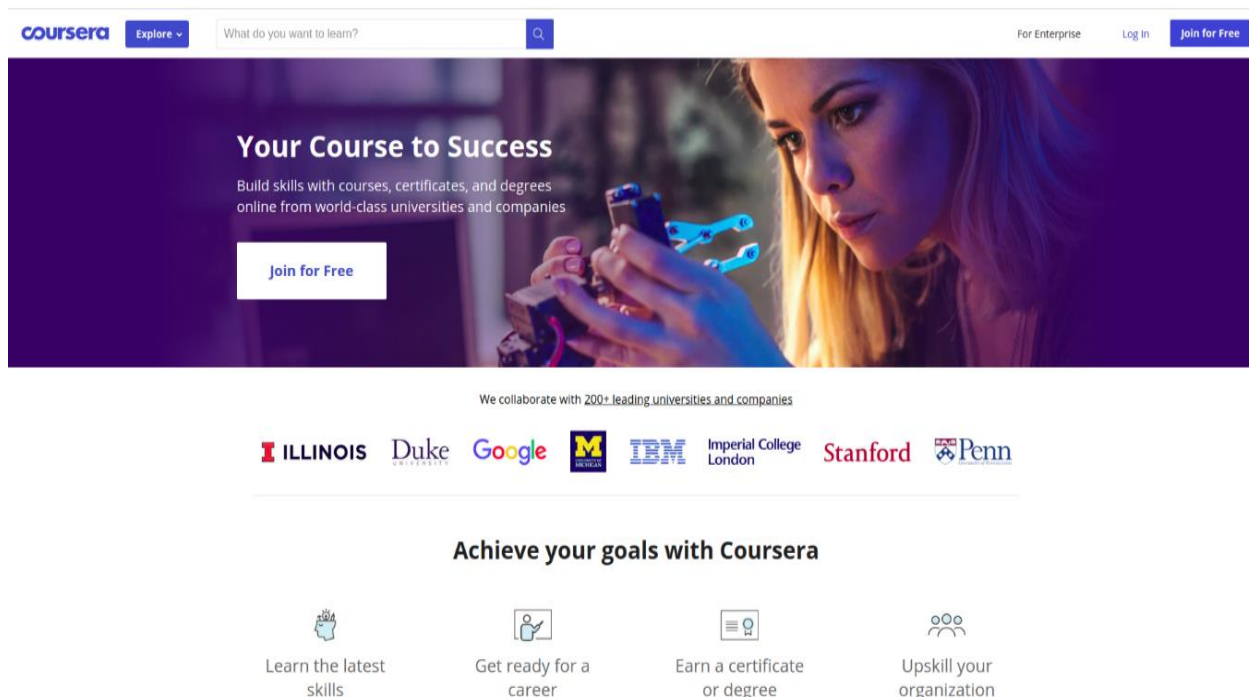


Рис. 1.6. Главная страница Coursera

Каждый курс Coursera [8] преподается лучшими преподавателями из университетов и компаний мирового уровня, так что вы можете узнать что-то новое в любое время и в любом месте. Сотни бесплатных курсов дают вам доступ к видео-лекциям по запросу, домашним заданиям и дискуссионным форумам сообщества. Платные курсы предоставляют дополнительные тесты и проекты, а также разделяемый сертификат курса по завершении.

После регистрации буквально через два часа вы сможете приступить к реализации своего проекта имея навыки, полученные во время обучения. Проекты являются самостоятельными, требуют меньших временных затрат и обеспечивают практическое использование инструментов в реальных сценариях, поэтому вы можете получить необходимые рабочие навыки именно тогда, когда они вам нужны.

На рисунке 1.7 представлена главная страница проекта Codecademy.

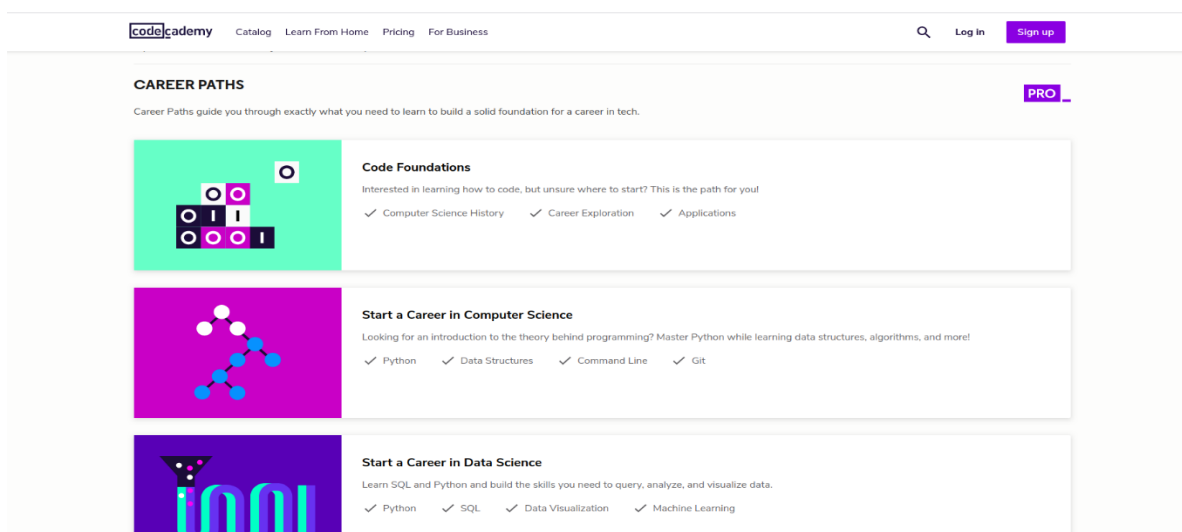


Рис. 1.7. Главная страница Codecademy

Codecademy [7] – образовательная компания. Они стремятся создать лучший опыт обучения внутри и снаружи, делая Codecademy лучшим местом для обучения, преподавания и создания опыта онлайн-обучения в будущем.

Нынешняя система государственных школ в США восходит к 19 веку и не предназначена для того, чтобы масштабировать ее так, как сейчас. Многие компании работают над тем, чтобы «нарушить» образование, изменив методы работы в классе и подключив класс к интернету. Они переосмысливают образование снизу вверх. Интернет переосмыслил практически все – коммерцию, социальные сети, здравоохранение и многое другое. Строят образование, в котором нуждается мир.

В таблице 1.1 ниже разобраны основные пункты сравнения образовательных площадок.

Таблица 1.1 – Сравнение образовательных площадок

	hexlet	stepik	codecademy	coursera	smotriuchis	udemy	geekbrains
Наличие бесплатных курсов	+	+	+	+	+	+	+
Поддержка русского языка	+	+	-	+	+	+	+
Выдача бесплатного сертификата	+	+	-	+	+	-	+
Большое количество курсов (>500)	-	+	-	+	+	+	-
Возможность создать свой курс	-	-	-	+/-	+	+	-
Автоматическая проверка домашнего задания	+	+	+	-	-	-	-

1.3 Обзор используемых технологий

Для создания платформы по автоматизированному обучению программированию долго подбирались технологии способные это осуществить. Первой проблемой стала изоляция кода. Для правильного обеспечения работы системы необходимо чтобы код, который будет выполняться внутри, и проверяться по тестам не навредил основной системе, которая его запустила. Поэтому первым делом разработалась концепция изоляции кода. Изолировать код получилось благодаря Docker контейнерам. Они работают следующим образом, они изолируют один процесс в системе и не позволяют ему взаимодействовать с ядром системы. В нем запускают необходимые операции и завершают процесс. В чем преимущество Docker контейнеров перед виртуальными машинами. Первое это скорость запуска,

так как пока операционная система в виртуальном окружении запустится пройдет много времени, а процесс стартует за считанные секунды и сразу приступает к выполнению заданий. Таким образом для системы были разработаны несколько Docker контейнеров под каждый язык, так как есть компилируемые языки, есть интерпретируемые, а некоторые требуют ещё специальных библиотек. Список поддерживаемых языков, следующий Python, C#, Go, Java, PascalABC.NET, C++. Это стандартный набор популярных языков, кроме PascalABC.NET, он добавлен только из-за того, что в российских школах его еще преподают.

Платформа выполнена в виде сайта и написана на веб Фреймворке Django. На рисунке 1.8 изображена схема взаимосвязи внутри проекта. Есть компонент Internet, через который приходят запросы, Nginx их обрабатывает передает через uvicorn и gunicorn к Django. Когда пользователь пытается получить доступ к сайту, он сначала попадает на обратный Nginx прокси, который разбирает web запрос. После отдает его процессу gunicorn который держит в памяти сразу несколько экземпляров Django и самому простаивающему передает запрос. А Django дальше на основании своих алгоритмов разбирает запрос и отправляет ответ по той же схеме только в обратном направлении.

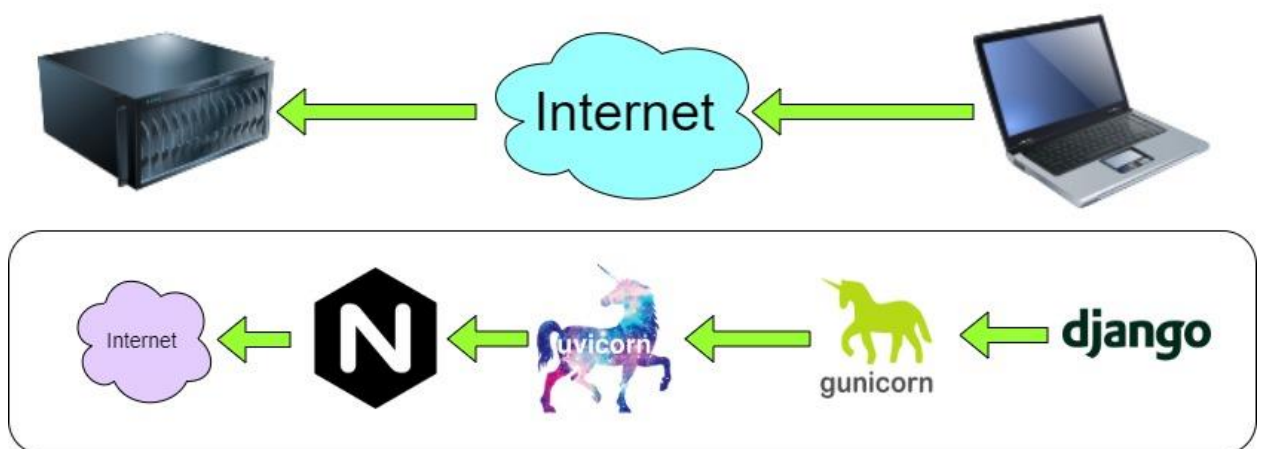


Рис. 1.8. Схема связи компонентов платформы.

2. Создание автоматизированной системы обучения программированию

2.1 Структура системы

Основной задачей разрабатываемой информационной веб-системы является автоматизация процессов проверки правильности выполнения заданий по программированию.

К разрабатываемой системе были предъявлены следующие требования:

1. Приятный дизайн и удобный интерфейс.
2. Регистрация пользователей.
3. Заполнение и редактирование записей в базе данных.
4. Просмотр статистики.
5. Добавление групп.
6. Добавление курсов.
7. Добавление разделов.
8. Добавление заданий к разделам.

Исходя из предъявленных требований, в системе существуют несколько типов пользователей: «Администратор», «Преподаватель», «Студент».

Администратору не требуется регистрация и вход, он выполняет свою работу в СУБД и поддерживает работоспособность всей веб-системы.

Преподавателя регистрирует администратор. В его возможности входит создание курсов, разделов, тестов и заданий, а также добавление и редактирование групп.

Студентов регистрирует преподаватель, если их ещё нет в системе, если есть, то может просто добавить целую группу на свой курс.

Студенты могут авторизовываться и видеть только те курсы, на которые их группу добавили.

Предполагаемая схема возможных действий в системе приведена на рисунке 2.1.

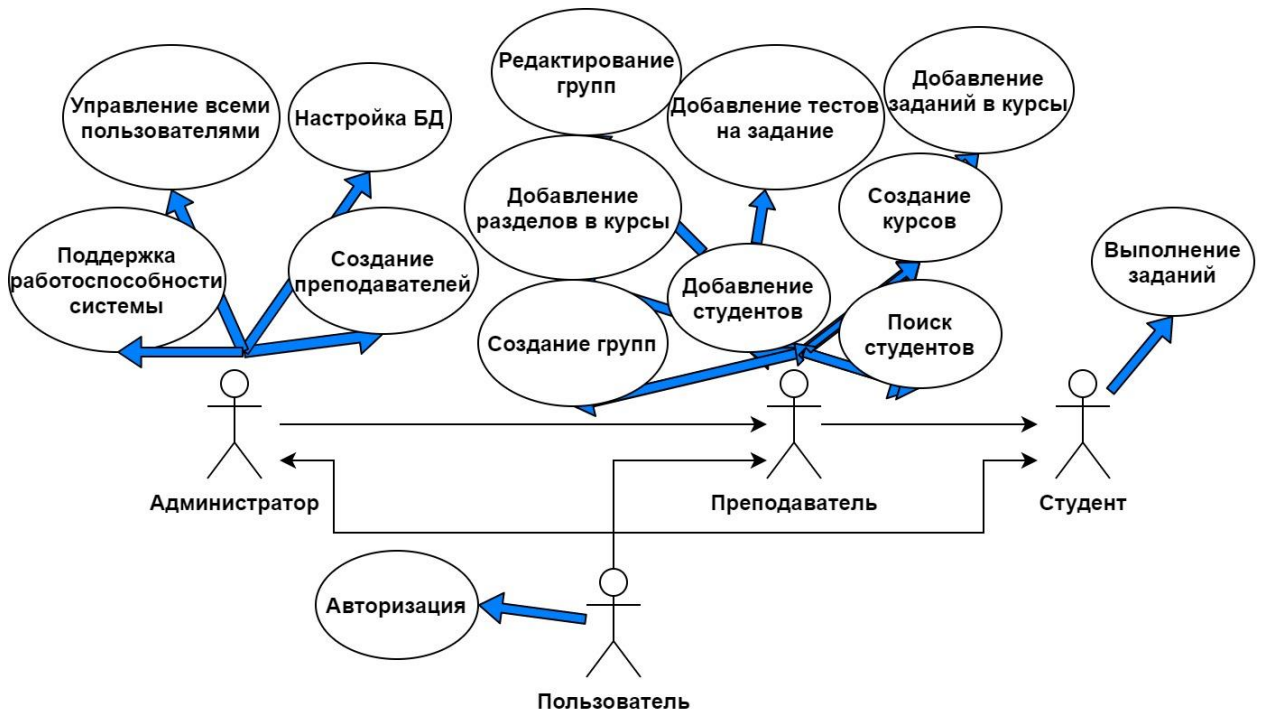


Рис. 2.1. Диаграмма вариантов использования

2.2 Разработка структуры базы данных

Существует множество средств для создания баз данных, но для этого проекта было решено выбрать систему управления PostgreSQL.

Разработка структуры базы данных началась с создания логической модели, в которой были созданы связи между таблицами. Были выявлены сущности модели и определены связи между ними (см. рис. 2.2).

Далее была сформирована физическая модель базы данных из 12 таблиц. Атрибутам были присвоены определенные типы данных, а также проведены связи между таблицами (см. рис. 2.2).

Теперь, чтобы подробнее рассмотреть атрибуты данной БД, структурируем информацию о них в виде оформленных таблиц.

Таблица «Пользователи» содержит подробную информацию о пользователях (таблица 2.1).

Таблица 2.1 – Описание атрибутов таблицы «User»

Название атрибута	Описание	Тип
username	Логин пользователя	CharField
email	Почта пользователя	EmailField
first_name	Фамилия пользователя	CharField
is_staff	Флаг модераторства	BooleanField
is_active	Флаг активности	BooleanField
middle_name	Отчество пользователя	CharField
last_name	Имя пользователя	CharField
is_teacher	Пользователь учитель?	BooleanField
password_unsafe	Хранение пароля для студентов	CharField
date_joined	Дата регистрации	DateTimeField

Таблица «UserGroup» содержит названия групп студентов (таблица 2.2).

Таблица 2.2 – Описание атрибутов таблицы «UserGroup»

Название атрибута	Описание	Тип
name	Название группы	CharField
users	Связь с таблицей User	ManyToManyField

Таблица «Course» содержит всю информацию про курс и группы, которые могут его посещать (таблица 2.3).

Таблица 2.3 – Описание атрибутов таблицы «Course»

Название атрибута	Описание	Тип
user	Связь с автором курса	ManyToManyField
groups	Список групп студентов	ManyToManyField
name	Название курса	CharField
description	Описание курса	TextField

Таблица «Section» содержит информацию разделах, которые входят в состав курса (таблица 2.4).

Таблица 2.4 – Описание атрибутов таблицы «Section»

Название атрибута	Описание	Тип
name	Название раздела	CharField
text	Полный текст раздела	TextField
priority	Приоритет в списке	SmallIntegerField
course	Связь с курсом	ForeignKey

Таблица «Exercise» содержит информацию о заданиях (таблица 2.5).

Таблица 2.5 – Описание атрибутов таблицы «Exercise»

Название атрибута	Описание	Тип
name	Название задания	CharField
text	Описание задания	TextField
languages	Язык программирования	ForeignKey
section	Связь с разделом	ForeignKey
priority	Приоритет в списке	SmallIntegerField
cpuctime	Предел по CPU	SmallIntegerField
memory	Предел по памяти	SmallIntegerField
point	Кол-во баллов за задание	SmallIntegerField

Таблица «Try» содержит информацию о попытке решения задачи (таблица 2.6).

Таблица 2.6 – Описание атрибутов таблицы «Try»

Название атрибута	Описание	Тип
exercise	Связь с заданием	ForeignKey
inputdata	Входные данные	TextField
outputdata	Выходные данные	TextField

Таблица «Result» содержит информацию о результате проверки задания (таблица 2.7).

Таблица 2.7 – Описание атрибутов таблицы «Result»

Название атрибута	Описание	Тип
user	Связь с моделью пользователя	ForeignKey
exercise	Связь с заданием	ForeignKey
time_create	Время создания модели	DateTimeField
result_time	Время затраченное на работу	FloatField
status	Статус выполнения задания	BooleanField
error	Сообщение об ошибке	TextField
stdinput	Входные данные	TextField
error_test	Провальный тест	TextField
pass_command	Флаг выполнения	BooleanField

Таблица «Tests» содержит информацию о модели тестов (таблица 2.8).

Таблица 2.8 – Описание атрибутов таблицы «Tests»

Название атрибута	Описание	Тип
course	Связь с курсом	ForeignKey
text	Имя теста	TextField

Таблица «Tests» содержит информацию о модели тестов (таблица 2.9).

Таблица 2.9 – Описание атрибутов таблицы «TestBody»

Название атрибута	Описание	Тип
test	Связь с тестами	ForeignKey
text	Вопрос	TextField
answer	Правильный ответ	CharField

Окончание таблицы 2.9

point	Баллы за правильный ответ	IntegerField
number	Порядок вопроса	SmallIntegerField
type_answer	Тип вопроса	CharField

Таблица «Options» содержит информацию о модели вариантов ответа (таблица 2.10).

Таблица 2.10 – Описание атрибутов таблицы «Options»

Название атрибута	Описание	Тип
body	Связь с телом теста	ForeignKey
text	Вариант ответа	TextField

Таблица «TryPassTest» содержит информацию о модели прохождения теста (таблица 2.11).

Таблица 2.11 – Описание атрибутов таблицы «TryPassTest»

Название атрибута	Описание	Тип
test	Связь с тестом	ForeignKey
user	Связь с пользователем	ForeignKey
answer	Ответ	CharField

Таблица «TestResult» содержит информацию о модели прохождения теста (таблица 2.12).

Таблица 2.12 – Описание атрибутов таблицы «TestResult»

Название атрибута	Описание	Тип
test	Связь с тестом	ForeignKey
user	Связь с пользователем	ForeignKey
passtest	Кол-во правильных ответов	IntegerField
alltest	Кол-во вопросов	IntegerField

Данная база данных отлично подходит для разрабатываемой информационной системы и хранения информации, которая в нее поступает.

2.3 Руководство пользователя

Разработка веб-системы «Steach.space» производилась с помощью таких языков программирования и технологий как Django, HTML, CSS, JavaScript и PostgreSQL. Перечисленные средства дают возможность для реализации проекта, а также являются бесплатными. Интерфейс очень прост в использовании, следовательно, у людей с низким уровнем взаимодействия с компьютером не должно возникнуть проблем.

При начале работы пользователя встречает главная страница сайта. (см. рис. 2.3).

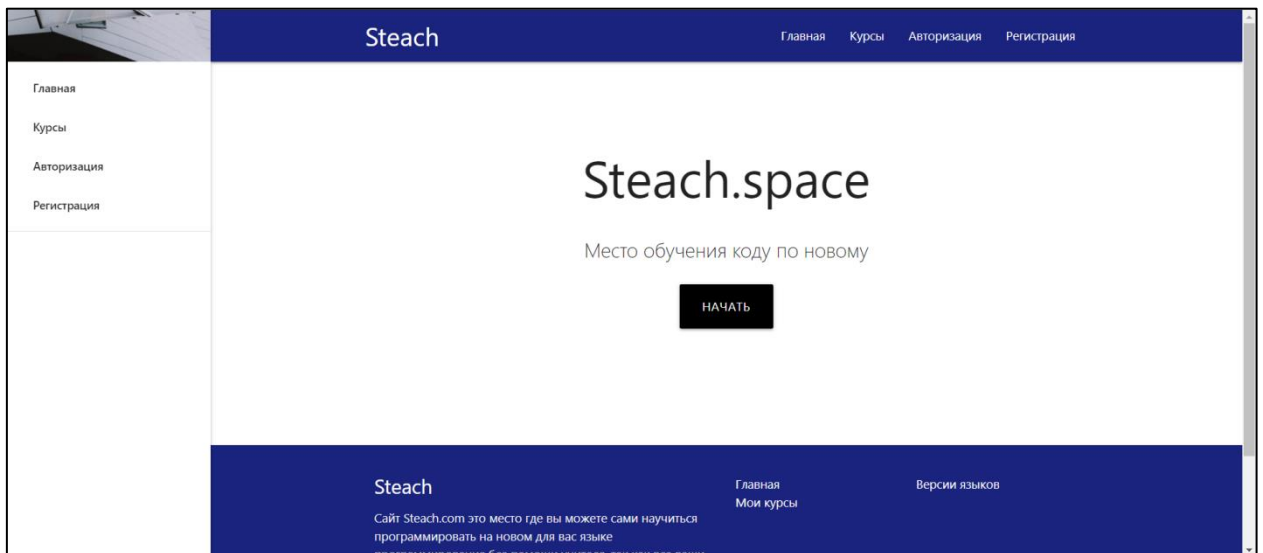


Рис. 2.3. Главная страница

При дальнейшем просмотре страницы видны меню авторизации, регистрации, курсов и версий языков. Подробнее рассмотрим данные функции сайта. Стоит заметить, что они доступны только авторизированному пользователю.

Начнем с того, что администратор создает преподавателей и они уже создают студентов, другими словами, обычному пользователю не получится просто зарегистрироваться в системе. Авторизация происходит по «приглашению». Авторизовавшись за преподавателя, главная страница выглядит как на рисунке 2.4.

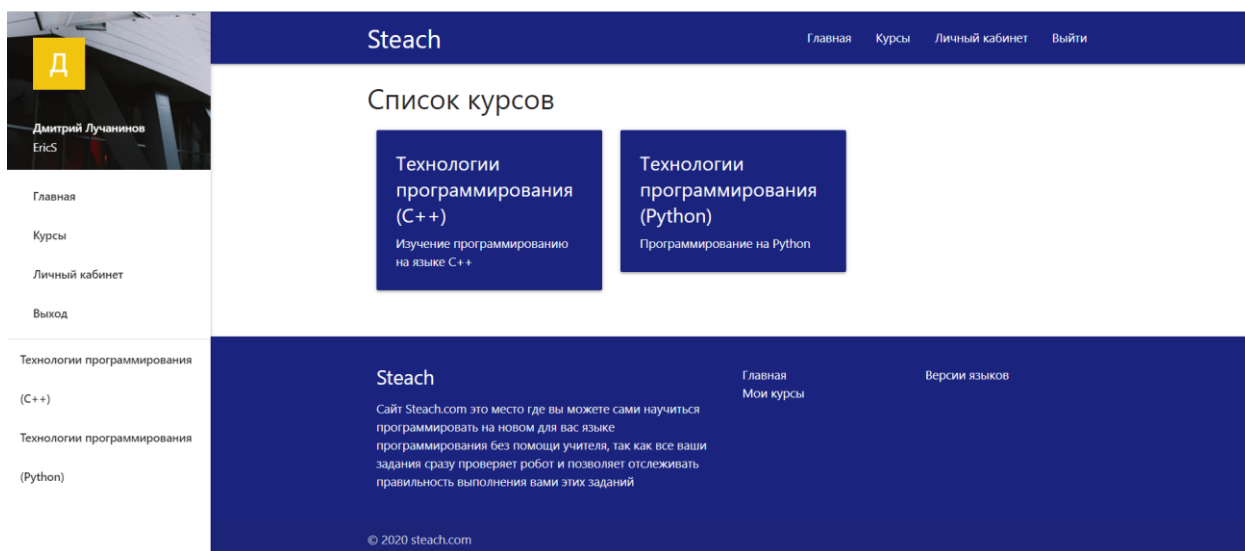


Рис. 2.4. Главная страница авторизованного пользователя

Первое что видит пользователь это список курсов, они также отображаются в левом боковом меню. Преподаватель сам может создавать и наполнять такие курсы. Откроем первый курс. Его вид изображен на рисунке 2.5.

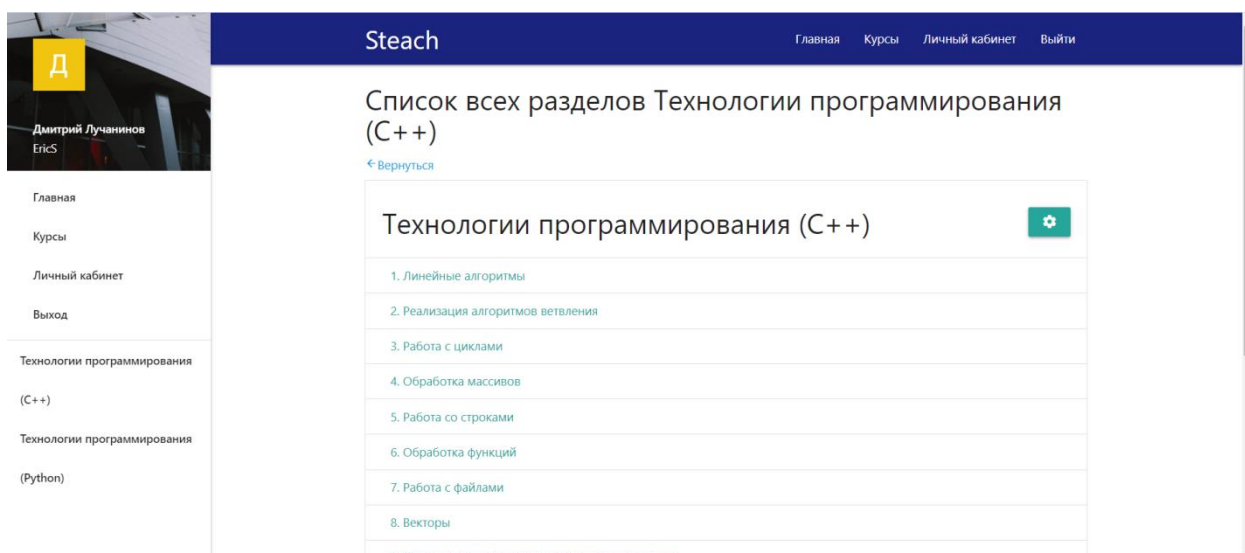


Рис. 2.5. Страница курса

В меню справа есть меню для работы с курсом. Если его раскрыть, будет как на рисунке 2.6.

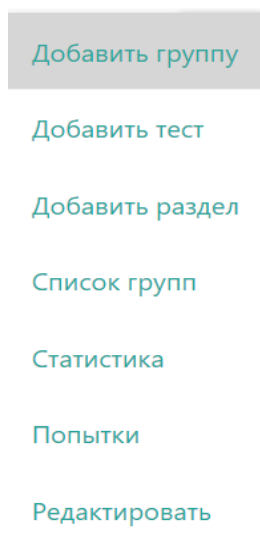


Рис. 2.6. Меню курса

Преподаватель может добавить на курс уже существующую группу и соответственно студентов, которые входят в эту группу. Может добавить тест, раздел, отобразить список групп, которые есть на курсе, посмотреть статистику, попытки решить задания и отредактировать курс.

При редактировании открывается форма с заполненными полями как на рисунке 2.7.

A form for editing a course. At the top left is a blue link with a left arrow labeled "Вернуться". Below it is a label "Название курса:" followed by a text input field containing "Технологии программирования (C++)". Below that is a label "Описание курса:" followed by a larger text area containing "Изучение программированию на языке C++". At the bottom right is a teal button labeled "СОХРАНИТЬ".

Рис. 2.7. Страница редактирования курса

Если щелкнуть на один из разделов, то откроется большая страница с теорией и заданиями как на рисунках 2.8 и 2.9.

1. Линейные алгоритмы

[← Вернуться](#)

[РЕДАКТИРОВАТЬ](#)

Язык программирования C++ на данный момент является одним из универсальных языков. Многие другие языки, такие как Java, PHP и т.д. используют сходные принципы программирования, кроме того, в отличие от Pascal, данный язык имеет полностью рабочую и устойчивую систему работы в объектно-ориентированной парадигме и средствами работы с базами данных.

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

```

1  /* первая программа */
_____

2  #include <iostream>
_____

   #include <cmath>
   #include <windows.h>

```

Рис. 2.8. Верхняя часть раздела

8. Каким образом на языке C++ вводятся константы?

9. Что означает команда return 0?

Задания

[Добавить задание](#)

1.0	Сдано
1.1.1	Не сдано
1.1.2	Не сдано
1.1.3	Не сдано
1.1.4	Не сдано
1.1.5	Не сдано
1.1.6	Не сдано
1.1.7	Не сдано
1.1.8	Не сдано

Рис.2.9. Нижняя часть раздела

Есть возможность редактирования раздела при помощи верхней кнопки

справа. Форма редактирования представлена на рисунке 2.10.

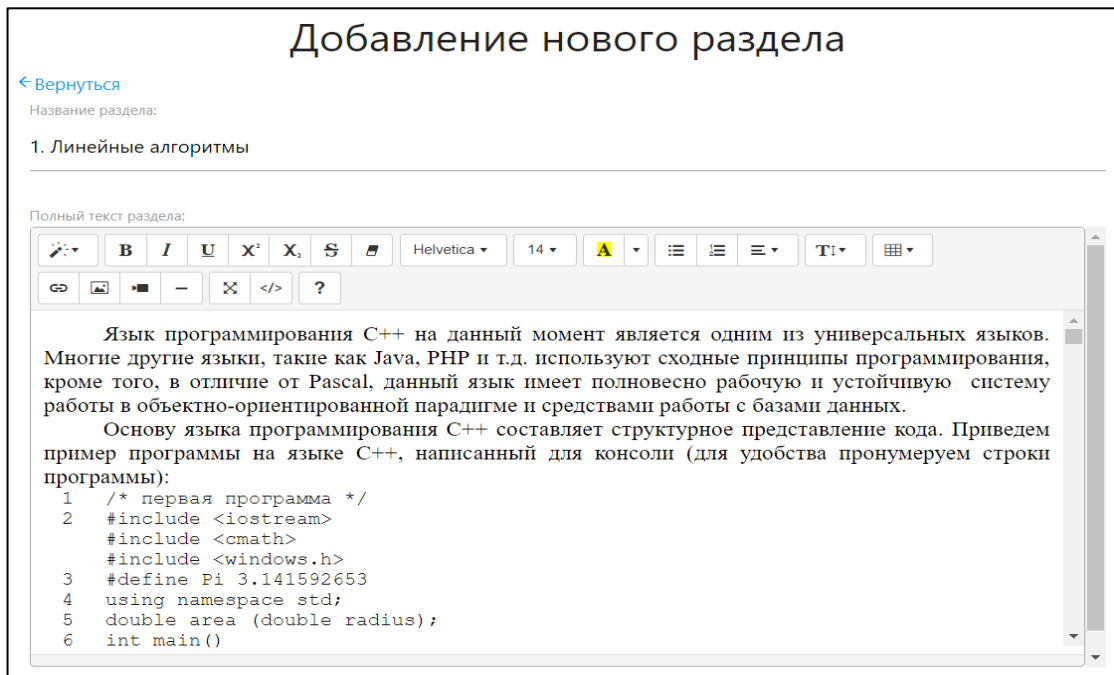


Рис. 2.10. Форма редактирования раздела

Страница задания выглядит как на рисунке 2.11 и 2.12.

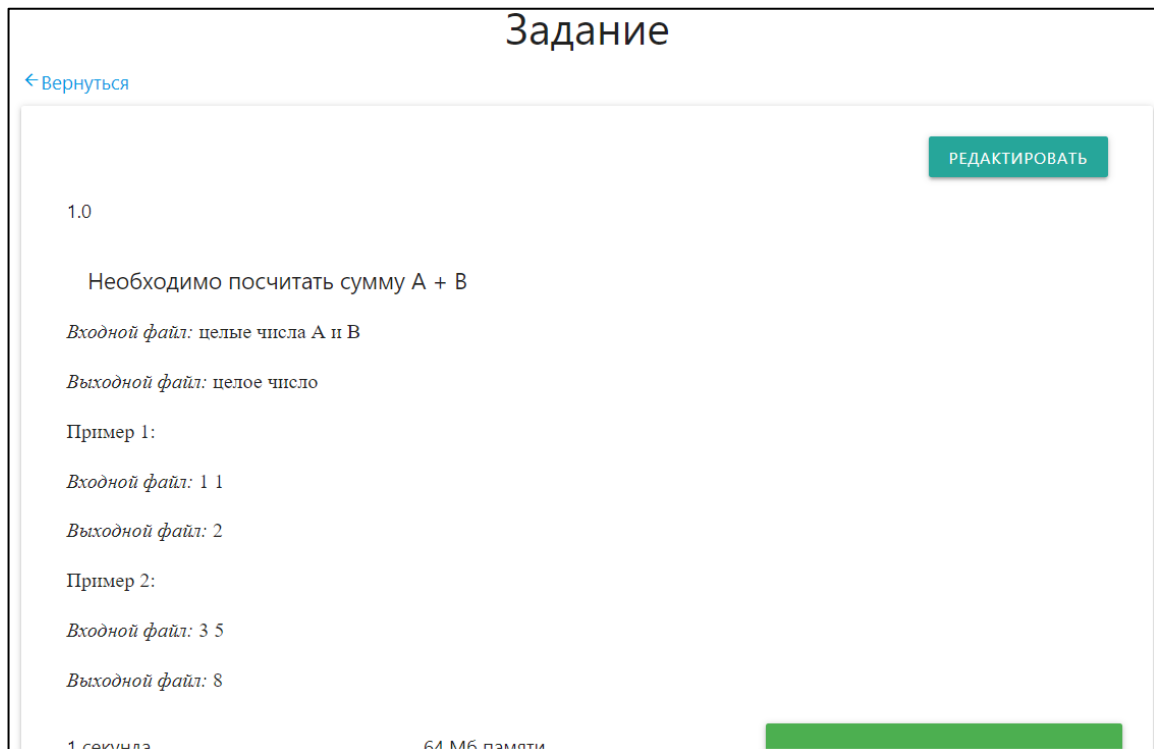


Рис. 2.11. Страница задания (верх)

Пример 2:

Входной файл: 3 5

Выходной файл: 8

1 секунда 64 Мб памяти

Сдано

Сдать нужно на языке C++

Введите код

ФАЙЛ

ОТПРАВИТЬ >

Время отправки	Статус	Скорость выполнения	Сообщение	Код
----------------	--------	---------------------	-----------	-----

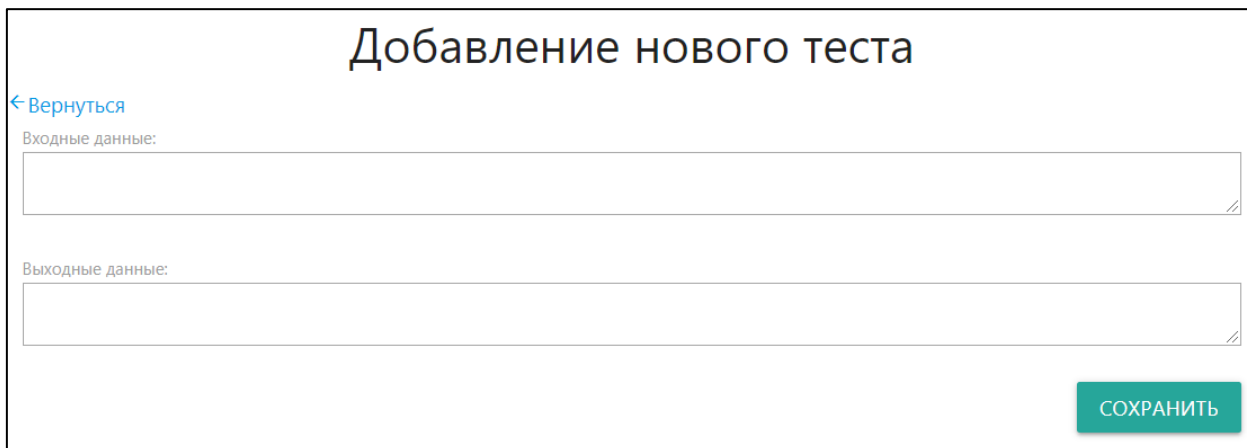
Рис. 2.12. Страница задания (низ)

Все попытки пользователя появляются внизу страницы и содержат важную информацию про статус решения задачи. Пример можно увидеть на рисунке 2.13.

Время отправки	Статус	Скорость выполнения	Сообщение	Код
30 сентября 2019 г. 8:21	Сдано	0,0 секунд		<code>#include <iostream> using namespace std; int main(){ int a,b; cin>>a>>b; cout<<a+b; return 0; }</code>
<h2>Тесты</h2>				
Добавить тест				
1 1		2		
2 3		5		
0 0		0		

Рис. 2.13. Вариант попытки сдачи задания

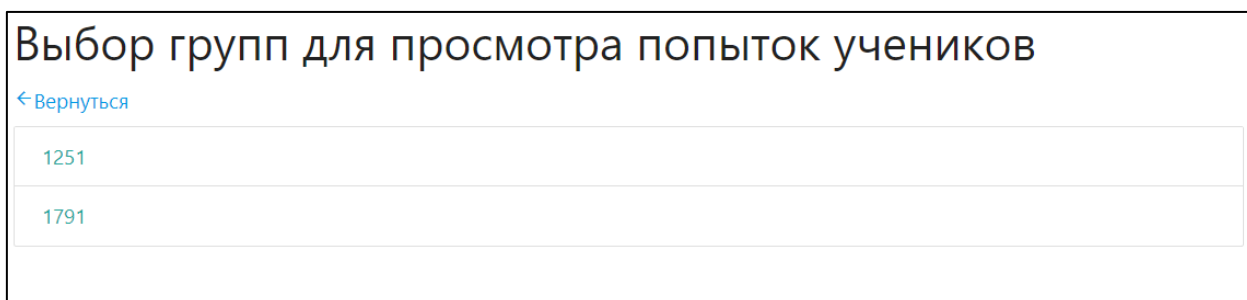
Преподаватели могут видеть какие тесты будут проходить программы пользователей, их видно на рисунке 2.13. Так же они могут и добавить новые тесты к существующим, форма представлена на рисунке 2.14.



The screenshot shows a web form titled "Добавление нового теста" (Adding a new test). At the top left, there is a blue link with a left-pointing arrow labeled "← Вернуться" (Return). Below the title, there are two input fields. The first is labeled "Входные данные:" (Input data) and the second is labeled "Выходные данные:" (Output data). Both fields are empty and have a small diagonal icon in the bottom right corner. At the bottom right of the form, there is a teal button with the text "СОХРАНИТЬ" (Save).

Рис. 2.14. Форма добавления нового теста

Есть возможность просмотреть все попытки студентов, отсортированных по времени, пример можно видеть на рисунках 2.15 и 2.16.



The screenshot shows a web form titled "Выбор групп для просмотра попыток учеников" (Select groups for viewing student attempts). At the top left, there is a blue link with a left-pointing arrow labeled "← Вернуться" (Return). Below the title, there is a list of two items, each in a separate row. The first item is the number "1251" and the second is "1791".

Рис. 2.15. Список групп для просмотра их попыток

ogoiuqdmkc	28 января 2020 г. 9:05	Сдано	0,0 секунд	
ogoiuqdmkc	28 января 2020 г. 9:03	Не сдано	0,0 секунд	b"main.cpp:1:21: warning: extra tokens at end of #include directive\n#include <iostream> #include <vector> using namespace std; int n; k=0; cin>>n; vector <int> avек(n); for (int i=1; i <= n; i++) { if (n % i == 0) { bool simple=true; for(int j=2; j<=i/2; j++) if(i%j == 0) { simple=false; break; } if (simple) { avек[k]=i; k++; } } } for (int i=0; i<k; i++) cout<<авек[i]<<' '; return 0; }\n ^\n/usr/lib/gcc/x86_64-linux-gnu/6/../../../../x86_64-linux-gnifunction `_start'\n(text+0x20): undefined reference to `main'\ncollect2: error: ld returned 1 exit status\n"
ogoiuqdmkc	28 января 2020 г. 9:02	Не сдано	0,0 секунд	b"main.cpp:1:21: warning: extra tokens at end of #include directive\n#include <iostream> using namespace std; int main () { int n; int k=0; <int> авек(n); for (int i=1; i <= n; i++) { if (n % i == 0) { bool simple=true; for(int j=2; j<=i/2; j++) if(i%j == 0) { simple=false; break; } if (simple) { avек[k]=i; k++; } } } for (int i=0; i<k; i++) cout<<авек[i]<<' '; return 0; }\n ^~~~~~\n/usr/lib/gcc/x86_64-linux-gnu/6/../../../../x86_64-linux-gnifunction `_start'\n(text+0x20): undefined reference to `main'\ncollect2: error: ld returned 1 exit status\n"
ogoiuqdmkc	28 января 2020 г. 9:01	Не сдано	0,0 секунд	
	28 января	Не		b"main.cpp: In function `int main()':main.cpp:12:3: error: expected `;' before

Рис. 2.16. Попытки студентов

Также преподаватель может видеть статистику по каждому заданию и студенту с автоматическим просчетом оценки. Пример такой статистики представлен на рисунке 2.17.

Статистика по всем пользователям группы 1791												
ФИО	Контрольная работа	9. Объектно-ориентированное программирование	8. Векторы	7. Работа с файлами	6. Обработка функций	5. Работа со строками	4. Обработка массивов	3. Работа с циклами	2. Реализация алгоритмов ветвления	1. Линейные алгоритмы	Оценка	Результаты
Бабин Юрий Олегович	12 из 40	0 из 0	6 из 6	6 из 6	6 из 6	8 из 8	12 из 12	8 из 8	6 из 6	8 из 8	Удовлетворительно	72 %
Бобров Андрей Сергеевич	24 из 40	0 из 0	6 из 6	6 из 6	6 из 6	8 из 8	12 из 12	8 из 8	6 из 6	8 из 8	Хорошо	84 %
Быков Алексей Игоревич	4 из 40	0 из 0	6 из 6	6 из 6	6 из 6	8 из 8	4 из 12	4 из 8	5 из 6	8 из 8	Удовлетворительно	51 %
Вебер Наталья Дмитриевна	24 из 40	0 из 0	6 из 6	6 из 6	6 из 6	8 из 8	12 из 12	8 из 8	6 из 6	8 из 8	Хорошо	84 %
Иванов Владислав Николаевич	0 из 40	0 из 0	0 из 6	0 из 6	0 из 6	0 из 8	0 из 12	0 из 8	5 из 6	8 из 8	Неудовлетворительно	13 %

Рис. 2.17. Статистика группы

Преподаватель может удалять и добавлять группы на курс, это можно видеть на рисунке 2.18.

Список групп добавленных на курс

[← Вернуться](#)

Название	Действие
1251	УДАЛИТЬ ГРУППУ
1791	УДАЛИТЬ ГРУППУ

Рис. 2.18. Список групп на курсе

Так как преподаватель может создавать группы, то он может и редактировать их, добавлять по 1 студенту, по несколько студентов, добавить существующего студента (если перевелся в другую группу), сделать выборку с логинами и паролями для последующей печати и выдачи студентам для входа. Все это можно наблюдать на рисунках с 2.19 по 2.23.

Группа "1791"

[← Вернуться](#)

[Добавить 1 пользователя](#)

[Добавить сразу несколько пользователей](#)

[Добавить существующего пользователя](#)

[Сделать выборку для выдачи](#)

Фамилия	Имя	Отчество	Редактирование	Удаление
[blurred]	[blurred]	[blurred]	РЕДАКТИРОВАТЬ	УДАЛЕНИЕ
[blurred]	[blurred]	[blurred]	РЕДАКТИРОВАТЬ	УДАЛЕНИЕ
[blurred]	[blurred]	[blurred]	РЕДАКТИРОВАТЬ	УДАЛЕНИЕ
[blurred]	[blurred]	[blurred]	РЕДАКТИРОВАТЬ	УДАЛЕНИЕ
[blurred]	[blurred]	[blurred]	РЕДАКТИРОВАТЬ	УДАЛЕНИЕ
[blurred]	[blurred]	[blurred]	РЕДАКТИРОВАТЬ	УДАЛЕНИЕ

Рис. 2.19. Страница редактирования группы

Добавить пользователя в группу "1791"

[← Вернуться](#)

Фамилия: _____ Имя: _____ Отчество: _____

ДОБАВИТЬ

Рис. 2.20. Форма добавления нового студента

Добавление 10 пользователей в группу "1791"

[← Вернуться](#)

Пользователь № 1

Фамилия: _____ Имя: _____ Отчество: _____

Пользователь № 2

Фамилия: _____ Имя: _____ Отчество: _____

Пользователь № 3

Фамилия: _____ Имя: _____ Отчество: _____

Пользователь № 4

Фамилия: _____ Имя: _____ Отчество: _____

Рис. 2.21. Форма добавления сразу 10 новых студентов

Найти пользователя и добавить его в группу

[← Вернуться](#)

Search: _____

НАЙТИ

Рис. 2.22. Форма создания и добавления нового студента в группу

Выборка паролей на группу

[← Вернуться](#)

Фамилия	Имя	Отчество	Логин	Пароль
[blurred]	[blurred]	[blurred]	[blurred]	[blurred]
[blurred]	[blurred]	[blurred]	[blurred]	[blurred]
[blurred]	[blurred]	[blurred]	[blurred]	[blurred]
[blurred]	[blurred]	[blurred]	[blurred]	[blurred]
[blurred]	[blurred]	[blurred]	[blurred]	[blurred]
[blurred]	[blurred]	[blurred]	[blurred]	[blurred]
[blurred]	[blurred]	[blurred]	[blurred]	[blurred]

Рис. 2.23. Страница с логинами и паролями студентов конкретной группы

На рисунке 2.24 представлена форма добавления нового раздела. Форма поддерживает многочисленное форматирование текста для лучшего восприятия текста.

Добавление нового раздела

[← Вернуться](#)

Название раздела:

Полный текст раздела:

🔍 **B** *I* U X^2 X_2 $\$$ Helvetica 14 **Tt**

[СОХРАНИТЬ](#)

Рис. 2.24. Форма добавления раздела

Преподаватель может добавить новую группу сразу со всеми студентами как на рисунке 2.25.

Добавление группы на курс

[← Вернуться](#)

Название	Действие
1251	ДОБАВИТЬ

Рис. 2.25. Страница добавления групп на курс

У каждого преподавателя есть личный кабинет, там отображаются список его курсов и групп, которые создал сам преподаватель. Пример можно видеть на рисунке 2.26.



Мои курсы 	Группы 
Технологии программирования (Python)	1251
Технологии программирования (C++)	1791

Рис. 2.26. Страница личного кабинета

Помимо разделов преподаватель может создавать тесты, первая форма для создания теста представлена на рисунке 2.27.

Добавление нового теста

[← Вернуться](#)

Имя теста:

Рис. 2.27. Форма добавления нового теста

В тест можно добавлять неограниченное количество вопросов. Пример показан на рисунке 2.28.

Работа с тестом "Итоговый тест"

[← Вернуться](#)

Итоговый тест

Рис. 2.28. Страница теста

Форма добавления контекста вопроса представлена на рисунке 2.29.

Добавление нового вопроса

[← Вернуться](#)

Вопрос:

Тип ответа: Баллы за правильный ответ:

Один ответ

Несколько ответов

[СОХРАНИТЬ](#)

Рис. 2.29. Форма заполнения вопроса на тест

После добавления вопроса, нужно добавить варианты ответа. Это проиллюстрировано на рисунках 2.30 и 2.31.

Работа с тестом "Итоговый тест"

[← Вернуться](#)

Итоговый тест

[⚙️](#)

[Нет ответа](#)

Сколько будет 2 в 5 степени?

Рис. 2.30. Страница теста

Вопрос "Сколько будет 2 в 5 степени?"

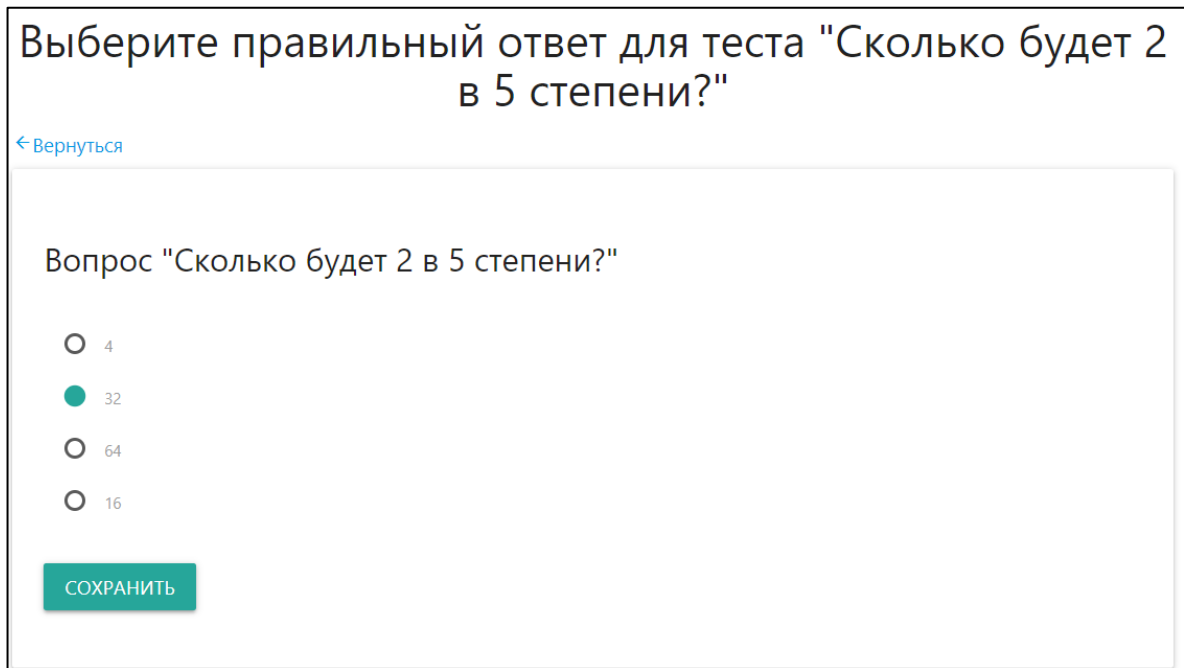
[← Вернуться](#)

Сколько будет 2 в 5 степени?

4	Добавить ответ
32	Изменить порядок вопросов
64	Выбрать правильный ответ
16	

Рис. 2.31. Страница вопроса на тест

Ну и последний шаг отметить верный вариант ответа. Пример изображен на рисунке 2.32.



Выберите правильный ответ для теста "Сколько будет 2 в 5 степени?"

[← Вернуться](#)

Вопрос "Сколько будет 2 в 5 степени?"

4

32

64

16

Рис. 2.32. Выбор правильного ответа на тест

2.4 Структура курса по изучению языка программирования C++ в рамках дистанционного обучения

Вводный урок

На данный момент существует множество систем дистанционного обучения. Они направлены на автоматизацию рутинных задач и позволяют сократить время на понимание принципов осваиваемой темы. Сегодня я расскажу вам про систему автоматического обучения языкам программирования в дистанционном формате. Для работы с ней потребуется любой современный браузер и выход в интернет. Вам не потребуется устанавливать себе какие-либо программные среды, вам достаточно отправить свой код через форму на этом сайте и подождать результата. Как это все выглядит мы сейчас с вами рассмотрим. Для начала, в системе не предусмотрена самостоятельная регистрация пользователей, вас должен зарегистрировать преподаватель, который ведет у вас материал.

Преподаватель заранее создает группу с вашим номером или номером класса и заполнить формы для учащихся в виде фамилии имени и отчества. После этого преподавателю будут доступны логин и пароль на каждого студента которые он раздаст. Также преподаватель заблаговременно создает лекционный курс и заполнить его материалом, заданиями и итоговым тестом. После этого он может добавить конкретную группу в курс, тем самым открыв доступ для студентов. Также, преподаватель может в любой момент редактировать и изменять свой курс. После авторизации студенту на главной странице будут доступны курсы, в которые его добавили. В каждом курсе есть несколько разделом, в зависимости от того сколько их добавит преподаватель. Раздел представляет собой теоретический материал, а в конце него задания для проверки. С заданиями уже становится интереснее, так как каждое сданное задание помечается флажком, сигнализирующем о его статусе (сдано, не сдано). При создании задания преподаватель может установить на каком языке программирования необходимо его выполнить, сколько будет начислено баллов за выполненное, и ограничения по памяти и времени выполнения. Например «Задание выполнить на языке программирования Javascript, с ограничениями по памяти в 32 мб и по времени за 1 секунду». Студент имеет возможность отправить исходный код решения в виде обычного текста в форму, либо в виде файла. После проверки, внизу задания будут отображены попытки студента, удачные и, возможно, не очень. Каждая попытка сообщает о статусе выполнения задания, времени сдачи, сообщения ошибки если такое произошло. После всех разделов курса есть контрольный тест для проверки усвоения тем всего курса. Он представляет собой классический тест в одним или множественным выбором варианта ответа. Так протекает работа студента, со стороны преподавателя есть статистика по результатам всех студентов. Преподаватель может отследить посмотреть какие задания выполнил конкретный студент. Система автоматически считает количество возможных

баллов и количество полученных баллов. На основании этих результатов высчитывается процент выполненных заданий и оценка.

Разберем работу системы на примере выполнения простого задания сложения двух чисел. Переходим на курс «Технологии программирования C++» и видим список разделов как на рисунке 2.33.

Список всех разделов Технологии программирования (C++)

[← Вернуться](#)


Технологии программирования (C++) 
1. Линейные алгоритмы
2. Реализация алгоритмов ветвления
3. Работа с циклами
4. Обработка массивов
5. Работа со строками
6. Обработка функций
7. Работа с файлами
8. Векторы
9. Объектно-ориентированное программирование
Контрольная работа

Рис. 2.33. Список разделов курса

Выбираем раздел «Линейные алгоритмы» изучив теоретический материал спускаемся вниз страницы и выбираем задание «1.0». Оно выглядит как показано на рисунке 2.34.

[← Вернуться](#)

1.0

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

1 секунда 64 Мб памяти **Не сдано**

Сдать нужно на языке C++

Введите код

ФАЙЛ

ОТПРАВИТЬ ▶

Рис. 2.34. Страница задания

Исходя из рисунка 2.34 мы видим, описание задания, требуемый язык программирования, входные и выходные данные, ограничения по времени выполнения в 1 секунду и по памяти в 64 мб. Также есть статус, который гласит, что мы ещё не выполнили задание.

Предположим, мы написали код, и он выглядит как показано ниже.

```
#include <iostream>
```

```
using namespace std;
```

```
int a,b;
```

```

int main(){

    cin >> a >> b;

    cout << a+b;

    return 0;

}

```

Теперь нужно вставить его в форму и нажать кнопку отправить.

На рисунке 2.35 можно увидеть, как статус «Не сдано» поменялся на «Сдано», что сигнализирует и успешной сдаче задания, также внизу появляются сообщения о попытках, на которых также можно увидеть, что задание принято и другую информацию.

Время отправки	Статус	Скорость выполнения	Сообщение	Код
17 июня 2020 г. 20:52	Сдано	0,0 секунд		#include <iostream> using namespace std; int a,b; int main(){ cin >> a >> b; cout << a+b; return 0; }

Рис. 2.35. Описание выполненного задания

Таблица 2.13 – Тематическое планирование

№	Тема занятия	Содержание занятий
1	Линейные алгоритмы	Описание базовых конструкций языка программирования C++ содержащих прямое и линейное решение.
2	Реализация алгоритмов ветвления	Описание логических выражений, конструкций if else и switch.
3	Работа с циклами	Описание необходимости повторяющихся операций и применение конструкций while и for.
4	Обработка массивов	Описание конструкций создания массивов. Описание типов массивов.
5	Работа со строками	Описание символов, символьных констант и строк.
6	Обработка функций	Описание функций, типов функций, возвращаемых значений.
7	Работа с файлами	Описание взаимодействия программ с текстовыми и байтовыми файлами.
8	Векторы	Описание специальных типов данных, векторы.
9	Объектно-ориентированное программирование	Описание принципов ООП, таких как наследование, переопределение и других.
10	Контрольная работа	Контрольный тест по всем темам курса.

В конце каждого курса для студентов подготовлен итоговый тест. По результатам которого можно сделать вывод, усвоил студент материал или нет. Пройдя все темы одного курса, студент выполняет итоговый тест. Вопросы итогового теста имеют специфику сложности, которую выставил преподаватель (за сложные вопросы начисляется больше баллов, за легкие – меньше).

2.5 Педагогический эксперимент по использованию автоматизированной системы обучению программирования

В апробации участвовали студенты 1 курса, группы 1791, Приамурского Государственного университета имени Шолом-Алейхема в количестве 5 человек. После апробации студентам было предложено пройти опрос, по результатам которого можно установить сильные и слабые стороны разработанного продукта.

Вопросы в опросе были следующие:

1. Как вы считаете, насколько актуальна данная система для вас?
2. Оцените сложность данной системы.
3. Хотели ли бы вы что бы данная система была частью вашего урока по изучению программирования?
4. Хотели ли бы вы использовать подобные ресурсы и по другим предметам?

Результаты были обработаны и построены диаграммы. Первые два вопроса оценивались по 10 бальной шкале, 2 последних вопроса оценивались ответом. Да или Нет

На первый вопрос ответили следующим образом (см рис. 2.36):

- 40% опрошенных – 10 баллов;
- 20% опрошенных – 9 баллов;
- 20% опрошенных – 8 баллов;
- 20% опрошенных – 1 балл.

Как вы считаете, насколько актуальна данная система для вас?

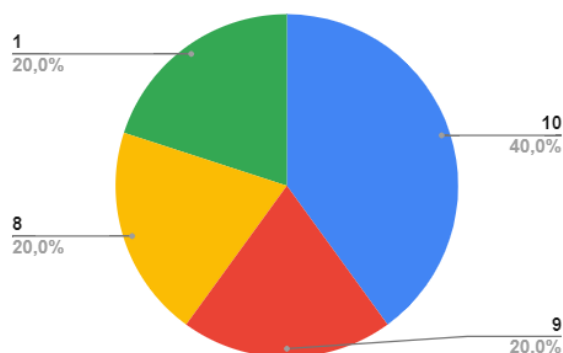


Рис. 2.36. Диаграмма 1 вопроса

На второй вопрос ответили следующим образом (см рис. 2.37):

- 60% опрошенных – 6 баллов;
- 20% опрошенных – 5 баллов;
- 20% опрошенных – 10 баллов.

Оцените сложность данной системы.

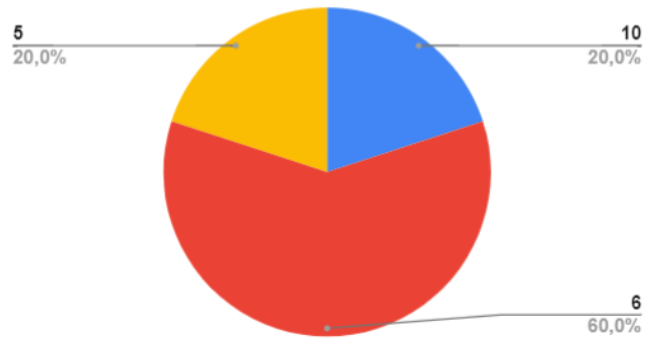


Рис. 2.37. Диаграмма 2 вопроса

На третий вопрос ответили следующим образом (см рис. 2.38):

- 80% ответили «Да»;
- 20% ответили «Нет».

Хотели ли бы вы что бы данная система была частью
вашего урока по изучению программирования?

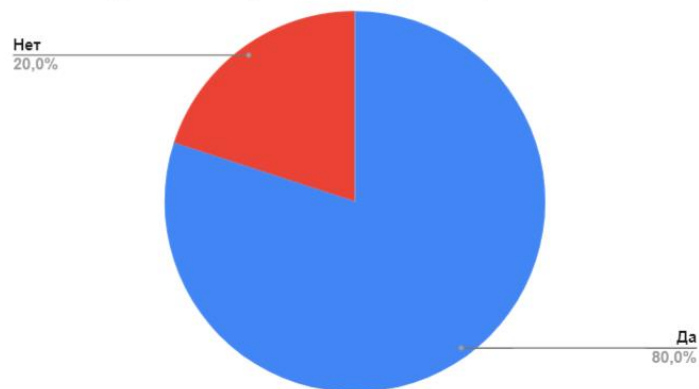


Рис. 2.38. Диаграмма 3 вопроса

На четвертый вопрос ответили следующим образом (см рис. 2.39):

- 80% ответили «Да»;
- 20% ответили «Нет».

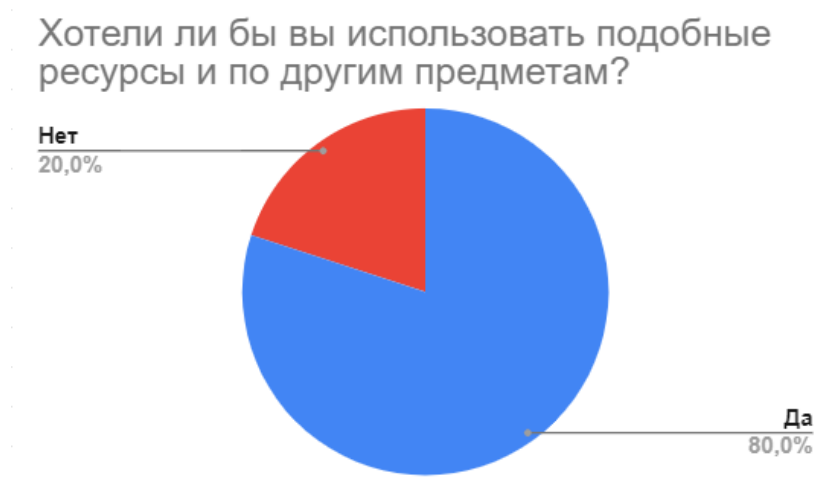


Рис. 2.39. Диаграмма 4 вопроса

Анализируя данные анкет, можно сделать вывод, что данная система очень актуальна на сегодняшний день и использовании подобных систем на других предметах хорошо принялось, но все может упереться в сложность системы.

Также была проведена экспертная оценка данной системы. В роли экспертов выступали пять преподавателей Приамурского государственного университета имени Шолом-Алейхема.

На сколько актуально использование данной системы на уроках информатики?

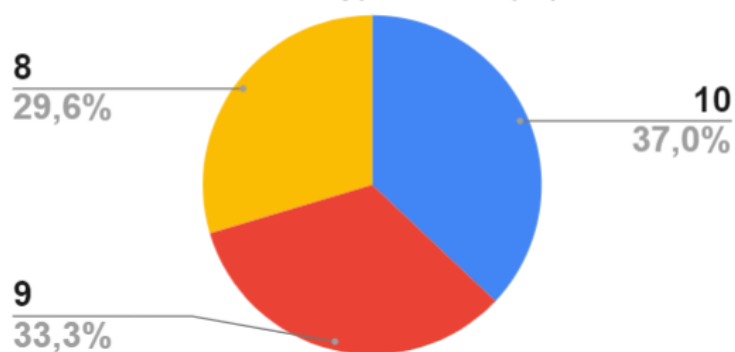


Рис. 2.40. Диаграмма 1 вопроса

На первый вопрос ответили следующим образом (см рис. 2.40):

- 37% опрошенных – 10 баллов;
- 33,3% опрошенных – 9 баллов;

– 29,6% опрошенных – 8 баллов.

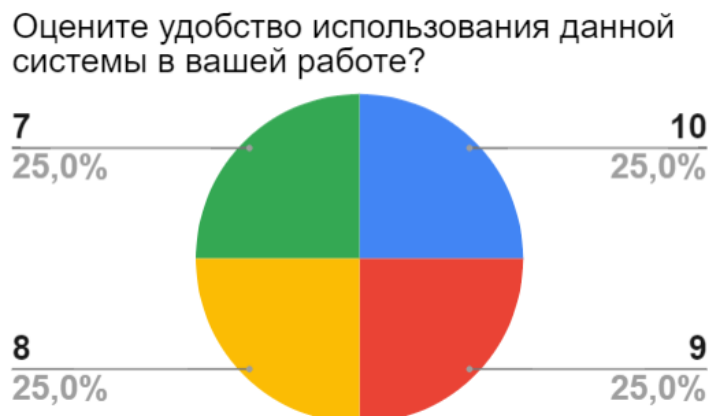


Рис. 2.41. Диаграмма 2 вопроса

На второй вопрос ответили следующим образом (см рис. 2.41):

- 25% опрошенных – 10 баллов;
- 25% опрошенных – 9 баллов;
- 25% опрошенных – 8 баллов;
- 25% опрошенных – 7 баллов.

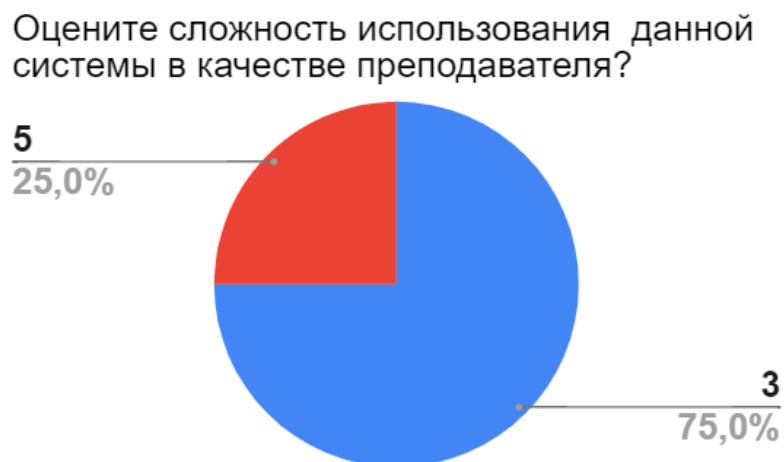


Рис. 2.42. Диаграмма 3 вопроса

На третий вопрос ответили следующим образом (см рис. 2.42):

- 75% опрошенных – 3 баллов;
- 25% опрошенных – 5 баллов.

Оцените возможность использования данной системы на самостоятельной работе

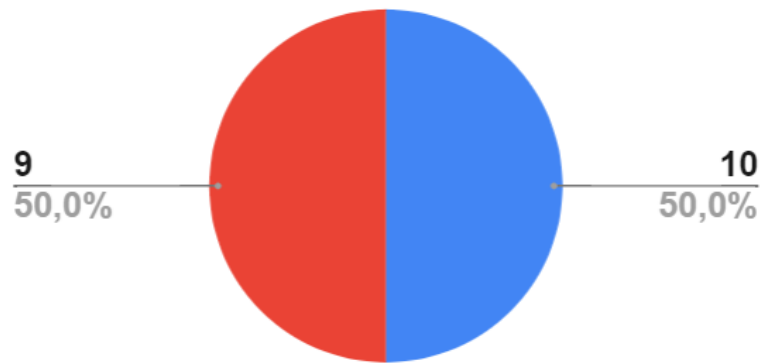


Рис. 2.43. Диаграмма 4 вопроса

На четвёртый вопрос ответили следующим образом (см рис. 2.43):

- 50% опрошенных – 10 баллов;
- 50% опрошенных – 9 баллов.

Оцените важность использования подобных ресурсов на свои занятия?

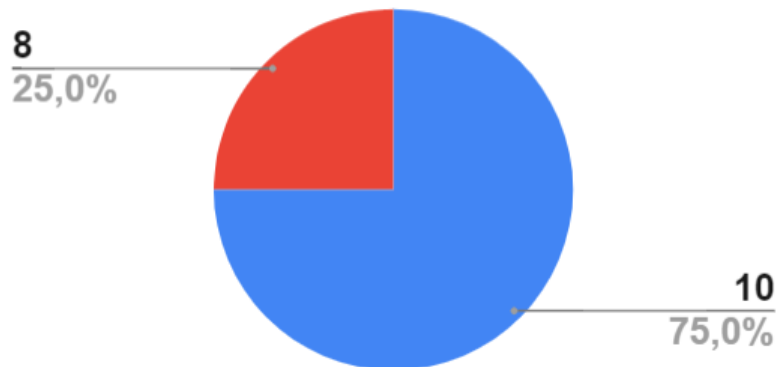


Рис. 2.44. Диаграмма 5 вопроса

На пятый вопрос ответили следующим образом (см рис. 2.44):

- 75% опрошенных – 10 баллов;
- 25% опрошенных – 8 баллов.

Оцените сложность использования
данного ресурса на своих уроках?



Рис. 2.45. Диаграмма 6 вопроса

На шестой вопрос ответили следующим образом (см рис. 2.45):

- 50% опрошенных – 4 баллов;
- 50% опрошенных – 5 баллов.

Анализируя данные анкет, можно сделать вывод, что экспертами была дана высокая оценка актуальности использования данной системы и они хотели бы её использовать в своих занятиях, но оценки удобства говорят о том, что принцип интерфейса требует доработки, так как с пониманием работы у экспертов возникли трудности.

Заключение

В результате выполнения данной работы была разработана автоматизированная система «Steach.space». Для достижения цели работы были решены следующие задачи:

1. Изучить методы традиционного обучения программированию.
2. Анализировать статьи, учебники и существующие среды автоматической проверки заданий по программированию.
3. Разработать информационную систему автоматизированного обучения программированию.
4. Провести педагогический эксперимент по апробации разработанной системы.

Методы и средства решения поставленной цели:

- изучение литературы по теме исследования;
- проектирование разработка программных систем;
- моделирование;
- педагогический эксперимент.

Разработанная система позволила достичь следующих результатов:

- уменьшение времени, затрачиваемого студентам и преподавателями университета на проверку задач по программированию;
- удобство в гибкой деятельности студента и преподавателя;
- автоматический расчет оценки для студента.

В ходе выполнения работы были решены все поставленные задачи и достигнута её основная цель. Разработанная система не вызывает трудностей при работе с ней. Интерфейс очень прост и удобен для любого пользователя.

Основными практическими результатами являются интерактивная методичка по обучению языкам программирования с автоматизированной проверкой заданий и возможностью создавать курсы с теоретической частью и практической. Такую систему можно применять на уроках по обучению программированию.

Библиографический список

1. Гладков В.П. Методы обучения программированию // Вестник Пермского государственного технического университета. Электротехника, информационные технологии, системы управления. 2009. № 3. С. 94-101. URL: <https://elibrary.ru/item.asp?id=12990680> (Дата обращения 26.03.2020)
2. Информатика и ИКТ. Учебник для 9 класса. В 2 ч. Босова Л.Л., Босова А.Ю. (2012; 244с., 79с.)
3. Информатика. 10 класс. Базовый уровень. Угринович Н.Д. (2017, 288с.)
4. Информатика. 9 класс. В 2 кн. Горячев А.В., Островский С.Л. и др. (2015; 288с., 192с.)
5. Конова Е.А., Поллак Г.А. Обучение программированию с использованием метода кейсов // Совет ректоров. 2014. № 2. С. 57-63. URL: <https://elibrary.ru/item.asp?id=21921085> (Дата обращения 26.03.2020)
6. Орлова М.С. Модели смешанного обучения и их применение при обучении программированию // Вестник Московского городского педагогического университета. Серия: Информатика и информатизация образования. 2008. № 11. С. 122-124. URL: <https://elibrary.ru/item.asp?id=13414339> (Дата обращения 26.03.2020)
7. Codecademy – Учись кодировать – бесплатно. Электронный ресурс URL: <https://www.codecademy.com/> (Дата обращения 26.03.2020)
8. Coursera – Навыки разработки с онлайн-курсами от ведущих учебных заведений. Электронный ресурс URL: <https://www.coursera.org/> (Дата обращения 26.03.2020)
9. GeekBrains – образовательный портал. Электронный ресурс URL: <https://geekbrains.ru/> (Дата обращения 26.03.2020)
10. Hexlet – Обучение программированию: онлайн-курсы для новичков и опытных. Электронный ресурс URL: <https://ru.hexlet.io/> (Дата обращения 26.03.2020)
11. Smotriuchis Мастер классы Сайт с мастер классами – Лучшие

онлайн курсы на сайте Smotriuchis Все образовательные программы и мастер-классы только у нас! Развивайся вместе с Smotriuchis! Электронный ресурс URL: <https://smotriuchis.ru/> (Дата обращения 26.03.2020)

12. Stepik – образовательная платформа и конструктор онлайн-курсов. Электронный ресурс URL: <https://stepik.org/> (Дата обращения 26.03.2020)

13. Udemу Онлайн-курсы – узнать что-нибудь новое, придерживаясь своего расписания | Udemу. Электронный ресурс URL: <https://www.udemy.com/> (Дата обращения 26.03.2020)