



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

ПРОФЕССИОНАЛЬНО-ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ  
КАФЕДРА АВТОМОБИЛЬНОГО ТРАНСПОРТА, ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ И МЕТОДИКИ ОБУЧЕНИЯ ТЕХНИЧЕСКИМ  
ДИСЦИПЛИНАМ

**Разработка электронной тестовой оболочки по дисциплине «Основы  
алгоритмизации и программирования» как средства контроля  
учебных достижений студентов колледжа**

**Выпускная квалификационная работа по направлению  
44.03.04. «Профессиональное обучение (по отраслям)»  
Направленность программы бакалавриата  
«Информатика и вычислительная техника»  
Форма обучения очная**

Проверка на объём заимствований:

67 % авторского текста

Работа *рекомендована* к защите

« 28 » мая 2020 г.

Заведующий кафедрой АТИТиМОТД

  
В.В. Руднев

Выполнил:

студент группы ОФ-409-079-4-1

Исаев Андрей Николаевич

Научный руководитель:

старший преподаватель кафедры

АТ, ИТ и МОТД

Шварцкоп Ольга Николаевна

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	3
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ ЭЛЕКТРОННО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ ЭЛЕКТРОННОЙ ТЕСТОВОЙ ОБОЛОЧКИ .....	5
1.1 Понятие электронной тестовой оболочки .....	5
1.2 Разработка электронной тестовой оболочки как методическая проблема.....	11
1.3 Нормативно-рекомендательная база по дисциплине «Основы алгоритмизации и программирования» как основа разработки тестовых заданий .....	14
ВЫВОД ПО ГЛАВЕ 1.....	26
ГЛАВА 2. РАЗРАБОТКА ЭЛЕКТРОННОЙ ТЕСТОВОЙ ОБОЛОЧКИ ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» КАК СРЕДСТВО КОНТРОЛЯ УЧЕБНЫХ ДОСТИЖЕНИЙ СТУДЕНТОВ КОЛЛЕДЖА.....	27
2.1 Выбор среды разработки для электронной тестовой оболочки.....	27
2.2 Структура и этапы разработки электронной тестовой оболочки ....	36
2.3 Анализ результатов исследования разработки электронной тестовой оболочки.....	41
ВЫВОД ПО ГЛАВЕ 2.....	46
ЗАКЛЮЧЕНИЕ .....	47
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	50

## ВВЕДЕНИЕ

В учебных заведениях большое внимание при проведении контроля успеваемости отводится тестированию. Устные опросы и бумажное тестирование играют большую роль в обучении. Устный опрос позволяет точнее узнать уровень усвоения знаний студента, но используя такой способ, можно опросить лишь небольшое количество человек. Бумажное тестирование позволяет проверить уровень усвоения знаний всей группы, но проверка результатов теста займёт большое количество времени у преподавателя.

С увеличением компьютеризации общества, тестовый контроль был перенесён в область компьютерных технологий. Компьютерные тесты можно применять для любого вида контроля: текущего, тематического, итогового. Электронные тесты являются эффективным способом контроля знаний, поэтому вызывает интерес у педагогов.

Для педагогов учебных заведений важными становятся программы, позволяющие проводить электронные тесты – тестовые оболочки. Таких программ разработано большое количество, и многие из них выделяются простым интерфейсом, что является важным для преподавателей и студентов.

Актуальность данной работы заключается в том, чтобы, используя информационные технологии при разработке автоматизированных систем контроля знаний облегчить труд преподавателя при проведении тестирования и обработки его результатов.

В этой связи актуальной становится тема «Разработка электронной тестовой оболочки по дисциплине «Основы алгоритмизации и программирования» как средство контроля учебных достижений студентов колледжа».

*Цель исследования:* теоретико-методическое обоснование и практическая разработка электронной тестовой оболочки.

*Объект исследования:* электронное учебно-методическое обеспечение по дисциплине «Основы алгоритмизации и программирования».

*Предмет исследования:* структура и содержание разработки электронной тестовой оболочки по дисциплине «Основы алгоритмизации и программирования».

*Задачи исследования:*

1. Изучить понятие электронной тестовой оболочки.
2. Исследовать процесс разработки электронной тестовой оболочки как теоретико-методическую проблему.
3. Проанализировать нормативно-рекомендательную базу по дисциплине «Основы алгоритмизации и программирования» как основа разработки тестовых заданий.
4. Обоснование выбора среды и этапы разработки электронной тестовой оболочки.
5. Разработать и оформить структуру и содержание разработки электронной тестовой оболочки по дисциплине «Основы алгоритмизации и программирования».
6. Проанализировать результаты исследования разработки электронной тестовой оболочки.

*Методы исследования:* изучение и анализ теоретико-методической, специальной и учебной литературы, изучение и анализ учебно-программной и планирующей документации, специальные методы проектирование педагогических и программных средств, проектирование тестовой оболочки.

*База исследования:* ГБПОУ «Южно-Уральский государственный колледж».

*Структура работы* включает введение, основную часть (две главы), заключение, список использованных источников.

# ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ ЭЛЕКТРОННО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ ЭЛЕКТРОННОЙ ТЕСТОВОЙ ОБОЛОЧКИ

## 1.1 Понятие электронной тестовой оболочки

Одним из направлений совершенствования процесса обучения является разработка оперативной системы контроля знаний, умений и навыков, позволяющей объективно оценивать знания студентов, выявляя имеющиеся пробелы и определяя способы их ликвидации. Поэтому вопросы контроля знаний интересуют многих ученых, как педагогов, так и специалистов в области информационных технологий. В настоящее время существует большое количество разнообразных способов проведения контроля и оценки знаний как при традиционном, так и при компьютерном обучении.

Широко применяется метод компьютерного педагогического тестирования как один из методов контроля усвоения студентами знаний по дисциплине, обладающий рядом важных преимуществ перед традиционными методами контроля знаний. Инструментом для измерения по шкале достижений студента является правильно сконструированный тест, который соответствует не только предмету обучения, но и его задачам и служит развитию системного подхода к изучению учебной дисциплины.

Стремительный рост быстродействия компьютерных систем, появление качественных и мощных систем программирования увеличило потребность в системах, позволяющих объективно, быстро и надежно оценивать знания учащихся, предлагая интересные формы взаимодействия с ними.

Тестирование применяется для определения соответствия предмета испытания заданным спецификациям. В задачи тестирования не входит определение причин несоответствия заданным требованиям. Тестирование

– один из разделов диагностики. Качество тестирования и достоверность его результатов в значительной степени зависит от тестера.

*Педагогическое тестирование* — это форма измерения знаний учащихся, основанная на применении педагогических тестов. Включает в себя подготовку качественных тестов, собственно проведение тестирования и последующую обработку результатов, которая даёт оценку обученности тестируемых [8].

Тестирование проводится с помощью специальных тестов, состоящих из определённого набора уровня несоответствия. Тестовое задание – это чёткое и ясное задание по конкретной предметной области, требующее однозначного определяемого ответа или выполнения определённого алгоритма действий.

В.С. Аванесов определяет *педагогический тест* как систему заданий специфической формы, определённого содержания, равномерно возрастающей трудности - система, создаваемая с целью объективно оценить структуру и измерить уровень подготовленности учащихся [1].

*Тест* (англ. *test* - проба, испытание, исследование) — в психологии и педагогике, стандартизированные задания, результат выполнения которых позволяет измерить психофизиологические и личностные характеристики, а также знания, умения и навыки испытуемого [6].

*Тестовое пространство* – множество различных тестовых заданий по всем модулям модели знаний.

*Полный тест* – подмножество тестового пространства, обеспечивающего объективную оценку соответствия между личностной и экспертной моделями знаний.

*Эффективный тест* – оптимальный по объёму и времени полный тест.

Тестирование выступает как педагогическое средство обучающей системы и может быть использовано как метод (технология) контроля

образовательного процесса, позволяющий оценить уровень знаний, умений и навыков обучаемого.

Тестирование - предназначено для оценки качества знаний учащихся, а также для оценки педагогических характеристик отдельных тестовых заданий и теста в целом [11].

Педагогическое тестирование как форма контроля знаний является неотъемлемой частью учебного процесса.

Компьютерное тестирование имеет ряд преимуществ:

— быстрое получение результатов испытания и освобождение преподавателя от трудоемкой работы по обработке результатов тестирования;

— объективность в оценке;

— конфиденциальность при анонимном тестировании;

— тестирование на компьютере более интересно по сравнению с традиционными формами опроса, что создает положительную мотивацию у студентов.

Составление компьютерных тестов является довольно сложным делом. Очень важно научиться отличать профессионально сделанный добротный тест от популярно-развлекательного журнального опросника. Настоящий, действенный, валидный и эффективно работающий тест — это законченный продукт, обладающий определенными свойствами и характеристиками и отвечающий современным методическим требованиям. Тест обладает составом, целостностью и структурой. Он состоит из заданий, правил их применения, оценок за выполнение каждого задания и рекомендаций по интерпретации тестовых результатов. Целостность теста проявляется во взаимосвязи заданий, включенных в тест. Ни одно из заданий не может быть изъято из теста без ущерба для него. Структура же его проявляется в способе связи заданий между собой.

Создание теста предполагает тщательный анализ содержания учебной дисциплины, классификацию учебного материала, установление

межпредметных связей, укрупнение дидактических единиц с последующим представлением этих единиц через элементы композиции задания.

Тесты бывают двух видов [13]:

- традиционные;
- нетрадиционные.

Традиционные тесты представлены в виде системы заданий возрастающей трудности, имеющие специфическую форму, позволяющие качественно и эффективно измерить уровень и оценить структуру подготовленности студентов.

При этом в зависимости от того, по скольким учебным дисциплинам включены в тест задания, традиционные тесты разделяют на гомогенные (проверяющие знания по одному предмету) и гетерогенные (по нескольким предметам).

Нетрадиционные тесты представлены интегративными и адаптивными.

Интегративным можно назвать тест, состоящий из системы заданий, нацеленных на обобщенную итоговую диагностику подготовленности выпускника образовательного учреждения. Диагностика проводится посредством предъявления таких заданий, правильные ответы на которые требуют интегрированных (обобщенных, явно взаимосвязанных) знаний двух и большего числа учебных дисциплин.

Адаптивные тесты позволяют регулировать трудность предъявляемых заданий в зависимости от ответов тестируемого. При успешном ответе компьютер выдает следующее задание, более трудное по сравнению с предыдущим, а в случае неудачи - более легкое.

Существуют две основные формы тестовых заданий:

1. Задания закрытой формы. Среди этих заданий выделяются такие разновидности, как:

- выбор одного правильного ответа по принципу: один - правильный, все остальные (один, два, три и т.д.) – неправильные;

- выбор нескольких правильных ответов;
- задания на установление соответствия, где элементам одного множества требуется поставить в соответствие элементы другого множества;
- задания на установление правильной последовательности (вычислений, действий, шагов, операций, терминов в определениях).

## 2. Задания открытой формы.

Задания сформулированы так, что готового ответа нет; нужно сформулировать и вписать ответ самому, в отведенном для этого месте.

Для компьютерного контроля знаний, осуществляемого в виде тестов, больше всего подходят задания с выбором одного правильного ответа. Среди этих тестов наиболее распространенными в настоящее время являются тесты с возможностью выбора правильного ответа из:

- двух предложенных вариантов ответа;
- трех предложенных вариантов.

Выбор формы зависит от:

- цели тестирования;
- содержания теста;
- технических возможностей;
- уровня подготовленности преподавателя в области теории и методики тестового контроля знаний.

Каждая из форм позволяет проверить специфические виды знаний. Проверять с помощью тестов имеет смысл актуальные знания, которые студенты должны уметь применять на практике. Проверяются знания, находящиеся в оперативной памяти, то есть, не требующие обращения к справочникам, словарям, картам, таблицам и т.п.

При разработке компьютерного теста очень важно продумать уровень его трудности в целом и отдельных тестовых заданий. Традиционно вопросы располагаются в порядке возрастающей трудности. Больше всего в процентном отношении составляет вопросов средней трудности. При

подборе заданий необходимо ориентироваться на общий уровень подготовленности обучающихся.

Самым лучшим можно считать тест, в котором заложено широкое содержание, и оно охватывает более глубокие уровни знаний. Разработчики тестов должны придерживаться следующих принципов:

1. Тест должен соответствовать целям тестирования.
2. Нужно определить значимость проверяемых знаний в общей системе проверяемых знаний.
3. Должна быть обеспечена взаимосвязь содержания и формы теста.
4. Тестовые задания должны быть правильными с точки зрения содержания.
5. Должна соблюдаться репрезентативность содержания учебной дисциплины в содержании теста.
6. Тест должен соответствовать уровню современного состояния науки.
7. Содержание теста должно быть комплексным и сбалансированным.
8. Содержание теста должно быть системным, но, вместе с тем, вариативным.

В начале любого теста дается краткая инструкция по выполнению задания, например: «Выберите правильный ответ...», «Выберите наиболее правильный ответ...», «Впечатайте в свободном поле ответ...» и т.п. Если задания представлены в одной форме, инструкция пишется один раз для всего теста. Если же тест включает различные задания, то перед каждым новым заданием пишется новая инструкция. Текст задания, как правило, пишется прописными буквами или жирным шрифтом для того, чтобы зрительно сразу же отделить само задание от вариантов ответа.

Одно из важных требований при тестировании - наличие заранее разработанных правил выставления баллов. В общем случае применения тестов за правильный ответ в каждом задании дается один балл, за неправильный - ноль. Сумм всех баллов, полученных студентом, дает число

правильных ответов. Это число ассоциируется с уровнем его знаний и с понятием «тестовый балл испытуемого».

Таким образом, построение компьютерных тестов можно осуществлять в следующей последовательности:

- формализация экспертной целевой модели знаний;
- нисходящее (или снизу – вверх) проектирование тестового пространства;
- формирование и наполнение тестовых заданий;
- формирование полного компьютерного теста;
- тестовый эксперимент;
- выбор эффективного теста;
- анализ, корректировка и доводка теста до вида эксплуатации.

Тестовые оболочки – это профессиональный инструмент автоматизации процесса тестирования и обработки результатов, который предназначен для решения широкого спектра задач:

1. Тестирование и контроль знаний учащихся.
2. Определение профессионального уровня сотрудников.
3. Проведение психологического тестирования.
4. Проведение опросов.
5. Организация олимпиад и конкурсов.

## 1.2 Разработка электронной тестовой оболочки как методическая проблема

Одним из направлений совершенствования процесса обучения является разработка оперативной системы контроля знаний, умений и навыков, позволяющей объективно оценивать знания студентов, выявляя имеющиеся пробелы и определяя способы их ликвидации. Поэтому вопросы контроля знаний (КЗ) интересуют многих ученых, как педагогов, так и специалистов в области информационных технологий. В настоящее время существует большое количество разнообразных способов проведения

контроля и оценки знаний как при традиционном, так и при компьютерном обучении.

В настоящее время широко применяется метод компьютерного педагогического тестирования как один из методов контроля усвоения студентами знаний по дисциплине, обладающий рядом важных преимуществ перед традиционными методами контроля знаний. Инструментом для измерения по шкале достижений студента является правильно сконструированный тест, который соответствует не только предмету обучения, но и его задачам и служит развитию системного подхода к изучению учебной дисциплины.

Компьютерное педагогическое тестирование используется, как правило, лишь как средство объективного контроля знаний после завершения изучения каждого раздела (модуля) учебной программы по дисциплине. Контролирование, оценивание уровня знаний студентов с одной стороны — это индикатор качества обучения – определение успеваемости студента, а с другой стороны – показатель совершенства эффективности дидактической системы (методики обучения и организации учебного процесса). Включение различных форм тестовых заданий в процесс обучения реализует на деле деятельностный подход в обучении.

Составление компьютерных тестов является довольно сложным делом. Очень важно научиться отличать профессионально сделанный тест от популярно-развлекательного журнального опросника. Настоящий, действенный, валидный и эффективно работающий тест — это завершённый продукт, обладающий определенными свойствами и характеристиками и отвечающий современным методическим требованиям. Тест обладает составом, целостностью и структурой. Он состоит из заданий, правил их применения, оценок за выполнение каждого задания и рекомендаций по интерпретации тестовых результатов. Целостность теста проявляется во взаимосвязи заданий, включенных в тест. Ни одно из

заданий не может быть изъято из теста без ущерба для него. Структура же его проявляется в способе связи заданий между собой.

Основные функции программ автоматизированного контроля знаний:

- разработка компьютерных тестов,
- автоматизация предъявления теста обучаемым, регистрация ответов,
- обработка результатов тестирования,
- определение педагогической оценки (зачтено – не зачтено; неудовлетворительно – удовлетворительно – хорошо – отлично).

Дополнительными характеристиками программ компьютерного тестирования являются:

- количество поддерживаемых форм тестовых заданий (т.е. типов шаблонов для ввода формулировки задания, исходных данных, иллюстративного материала, вариантов ответа и данных для оценки степени правильности ответа);
- шкала педагогического оценивания;
- возможность использования мультимедийных файлов (графика, видео, аудио) для иллюстрации задания и в качестве вариантов ответов;
- задание условий формирования выборки вопросов при тестировании,
- задание критериев оценки,
- возможность сохранения и печати протокола тестирования (после подписания обучаемым протокол может служить документальным подтверждением факта тестирования и обоснованием педагогической оценки);
- накопление и анализ статистики ответов;
- возможность работы в сети.

Построение компьютерных тестов можно осуществить по следующим последовательным шагам:

1. Формализация экспертной целевой модели знаний.

2. Нисходящее проектирование тестового пространства.
3. Формирование и наполнение тестовых заданий.
4. Формирование полного компьютерного теста.
5. Тестовый эксперимент.
6. Выбор эффективного теста.
7. Анализ, корректировка и доводка теста до вида эксплуатации.

Требования к тестовым оболочкам:

1. Возможность создания тестов разного вида (закрытых, открытых)
2. Возможность представления вопроса не только в текстовой форме, но и в графическом виде
3. Возможность хранения большого количества вопросов
4. Наличие статистической обработки данных

В компьютерных тестах необходимо также предусматривать тестирование знаний и профессиональных способностей. Как правило, основной задачей педагогического использования компьютерных тестов является определение объема и качества знаний, уровня навыков и умений.

1.3 Нормативно-рекомендательная база по дисциплине «Основы алгоритмизации и программирования» как основа разработки тестовых заданий

Рабочая программа учебной дисциплины является частью рабочей основной образовательной программы в соответствии с ФГОС СПО 09.02.07 Информационные системы и программирование.

Учебная дисциплина «Основы алгоритмизации и программирования» принадлежит к общепрофессиональному циклу.

В результате освоения дисциплины обучающийся должен уметь:

1. Разрабатывать алгоритмы для конкретных задач.
2. Использовать программы для графического отображения алгоритмов.
3. Определять сложность работы алгоритмов.

4. Работать в среде программирования.
5. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
6. Оформлять код программы в соответствии со стандартом кодирования.
7. Выполнять проверку, отладку кода программы.

В результате освоения дисциплины обучающийся должен знать:

1. Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.
2. Эволюцию языков программирования, их классификацию, понятие системы программирования.
3. Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
4. Подпрограммы, составление библиотек подпрограмм.
5. Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

В результате освоения дисциплины обучающийся осваивает элементы компетенций:

ОК 1 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам,

ОК 2 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности,

ОК 4 Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами,

ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста,

ОК 9 Использовать информационные технологии в профессиональной деятельности,

ОК 10 Пользоваться профессиональной документацией на государственном и иностранном языках,

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4 Выполнять тестирование программных модулей.

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода.

ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ,

ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

*Объем учебной дисциплины и виды учебной работы:*

Обязательная учебная нагрузка 156 часов.

В том числе:

— теоретическое обучение 84 часа;

— практические занятия 72 часа.

Промежуточная аттестация проводится в форме экзамена.

Таблица 1.1. – Тематический план и содержание учебной дисциплины

<i>Наименование разделов и тем</i>	<i>Содержание учебного материала и формы организации деятельности обучающихся</i>		<i>Объем часов</i>	<i>Осваиваемые элементы компетенций</i>
<i>1</i>	<i>2</i>		<i>3</i>	<i>4</i>
<b>Раздел 1.</b>	<b><i>Основы алгоритмизации</i></b>		<b>24</b>	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1, ПК 1.2, ПК 2.5
<b>Тема 1.1. Введение в теорию алгоритмов</b>	<b><i>Содержание учебного материала</i></b>	<b><i>Уровень освоения</i></b>	<b>4</b>	
	1. Понятие алгоритма. Развитие представления об алгоритме. Свойства алгоритма. 2. Способы описания алгоритма. Машина Тьюринга.	<b>1</b>		
	<b><i>Тематика практических занятий</i></b>		---	
	<b><i>Самостоятельная работа обучающихся</i></b>		---	
<b>Тема 1.2. Основные алгоритмические конструкции</b>	<b><i>Содержание учебного материала</i></b>	<b><i>Уровень освоения</i></b>	<b>4</b>	
	1. Понятие блок-схемы алгоритма. Описание блоков. Алгоритмы линейной структуры. Алгоритмы разветвляющейся структуры. Алгоритмы с вложенной ветвящейся структурой. Алгоритмы циклической структуры. Цикл с предусловием, цикл с постусловием. Цикл с параметром. Алгоритмы с вложенными циклами. Алгоритмы работы с массивами. 2. Составление блок-схем алгоритмов.	<b>1</b>		

	<p><b>Тематика практических занятий</b></p> <p>Разработка алгоритма словесным способом, формульно-словесным, блок-схемным, псевдокодом, структурной диаграммой.</p> <p>Разработка алгоритмов линейной структуры.</p> <p>Разработка алгоритмов разветвляющейся структуры.</p> <p>Разработка алгоритмов циклической структуры.</p> <p>Разработка алгоритмов CASE-структуры.</p> <p>Составление блок-схем алгоритмов для обработки массивов.</p>	<b>2</b>	<b>16</b>	
	<b>Самостоятельная работа обучающихся</b>		---	
<b>Раздел 2.</b>	<b>Введение в программирование</b>		<b>8</b>	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5
<b>Тема 2.1. Языки программирования</b>	<b>Содержание учебного материала</b>	<b>Уровень освоения</b>	<b>6</b>	
	<ol style="list-style-type: none"> <li>1. Развитие языков программирования.</li> <li>2. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.</li> <li>3. Жизненный цикл программы. Программа. Программный продукт и его характеристики.</li> </ol>	<b>1</b>		
	<b>Тематика практических занятий</b>		---	
	<b>Самостоятельная работа обучающихся</b>		---	
<b>Тема 2.2. Типы данных</b>	<b>Содержание учебного материала</b>	<b>Уровень освоения</b>	<b>2</b>	
	1. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.	<b>1</b>		

	<i>Тематика практических занятий</i>	---	
	<i>Самостоятельная работа обучающихся</i>	---	

<b>Раздел 3.</b>	<b><i>Разработка программ на процедурном, структурном языках</i></b>		<b>66</b>	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5
<b>Тема 3.1. Структуризация в программировании</b>	<b><i>Содержание учебного материала</i></b>	<b><i>Уровень освоения</i></b>	<b>4</b>	
	1. Основы структурного программирования. Методы структурного программирования.	1		
	<b><i>Тематика практических занятий</i></b>		---	
	<b><i>Самостоятельная работа обучающихся</i></b>		---	
<b>Тема 3.2. Операторы языка программирования</b>	<b><i>Содержание учебного материала</i></b>	<b><i>Уровень освоения</i></b>	<b>12</b>	
	<ol style="list-style-type: none"> <li>1. Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.</li> <li>2. Условный оператор. Оператор выбора.</li> <li>3. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы.</li> <li>4. Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.</li> <li>5. Структурированный тип данных – множество. Операции над множествами.</li> <li>6. Комбинированный тип данных – запись. Файлы</li> </ol>	1		

	последовательного доступа. Файлы прямого доступа			
	<b>Тематика практических занятий</b> Знакомство со средой программирования. Составление программ линейной структуры. Составление программ разветвляющейся структуры. Составление программ циклической структуры Обработка одномерных массивов. Обработка двумерных массивов. Работа со строками. Работа с данными типа множество. Файлы последовательного доступа. Типизированные файлы. Нетипизированные файлы.	2	22	
	<b>Самостоятельная работа обучающихся</b>		---	
<b>Тема 3.3. Процедуры и функции</b>	<b>Содержание учебного материала</b>	<b>Уровень освоения</b>	<b>4</b>	
	1. Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций.	1		
	2. Рекурсия. Программирование рекурсивных алгоритмов.			
	<b>Тематика практических занятий</b> Организация процедур. Организация функций. Применение рекурсивных функций.	2	6	
	<b>Самостоятельная работа обучающихся</b>		---	

<b>Тема 3.4. Модульное программирование</b>	<i>Содержание учебного материала</i>	<i>Уровень освоения</i>	<b>4</b>	
	1. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы. 2. Стандартные модули	1		
	<i>Тематика практических занятий</i> Программирование модуля. Создание библиотеки подпрограмм.	2	<b>4</b>	
	<i>Самостоятельная работа обучающихся</i>		---	
<b>Тема 3.5. Указатели.</b>	<i>Содержание учебного материала</i>	<i>Уровень освоения</i>	<b>6</b>	
	1. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных. 2. Структуры данных на основе указателей. 3. Стек и очередь.	<b>1</b>		
	<i>Тематика практических занятий</i> Использование указателей для организации связанных списков.	2	<b>4</b>	
	<i>Самостоятельная работа обучающихся</i>		---	
<b>Раздел 4.</b>	<i>Разработка программ в объектно-ориентированной среде</i>		<b>58</b>	ОК 1 ОК 2
<b>Тема 4.1. Основные</b>	<i>Содержание учебного материала</i>	<i>Уровень освоения</i>	<b>12</b>	ОК 4 ОК 5

<b>принципы объектно-ориентированного программирования (ООП)</b>	<p>1. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.</p> <p>2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.</p> <p>3. Классы объектов. Компоненты и их свойства.</p> <p>4. Событийно-управляемая модель программирования. Компонентно-ориентированный подход.</p>	<b>1</b>		<p>ОК 9 ОК 10 ПК 1.1- ПК 1.6 ПК 2.4, 2.5</p>
	<i>Тематика практических занятий</i>		---	
	<i>Самостоятельная работа обучающихся</i>		---	
<b>Тема 4.2. Интегрированная среда разработки</b>	<i>Содержание учебного материала</i>	<i>Уровень освоения</i>	<b>10</b>	
	<p>1. Требования к аппаратным и программным средствам интегрированной среды разработки.</p> <p>2. Интерфейс среды разработки: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.</p> <p>3. Панель компонентов и их свойства. Окно кода проекта. Настройка среды и параметров проекта.</p> <p>4. Состав и характеристика проекта. Выполнение проекта.</p> <p>5. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта.</p>	<b>1</b>		
	<i>Тематика практических занятий</i>	<b>2</b>	<b>2</b>	
	Изучение интегрированной среды разработки.			

	<i>Самостоятельная работа обучающихся</i>		---	
<b>Тема 4.3. Визуальное событийно- управляемое программирование</b>	<i>Содержание учебного материала</i>	<i>Уровень освоения</i>	<b>6</b>	
	1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.  2. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.  3. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.	<b>1</b>		
	<i>Тематика практических занятий</i>  Создание проекта с использованием компонентов для работы с текстом. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий. Создание проекта с использованием кнопочных компонентов. Создание проекта с использованием компонентов стандартных диалогов и системы меню.	<b>2</b>	<b>8</b>	
	<i>Самостоятельная работа обучающихся</i>		---	
<b>Тема 4.4 Разработка</b>	<i>Содержание учебного материала</i>	<i>Уровень освоения</i>	<b>4</b>	

<b>оконного приложения</b>	1. Разработка функционального интерфейса приложения. Создание интерфейса приложения. 2. Разработка функциональной схемы работы приложения. 3. Разработка игрового приложения.	<b>1</b>		
	<b>Тематика практических занятий</b> Разработка функциональной схемы работы приложения. Разработка оконного приложения с несколькими формами.	<b>2</b>	<b>4</b>	
	<b>Самостоятельная работа обучающихся</b>		---	
<b>Тема 4.5 Этапы разработки объектно-ориентированных приложений</b>	<b>Содержание учебного материала</b>	<b>Уровень освоения</b>	<b>4</b>	
	1. Проектирование объектно-ориентированного приложения. 2. Создание интерфейса пользователя. 3. Тестирование, отладка приложения.	<b>1</b>		
	<b>Тематика практических занятий</b> Создание процедур обработки событий. Компиляция и запуск приложения. Разработка интерфейса приложения. Тестирование, отладка приложения.	<b>2</b>	<b>4</b>	
	<b>Самостоятельная работа обучающихся</b>		---	
<b>Тема 4.6 Иерархия классов</b>	<b>Содержание учебного материала</b>	<b>Уровень освоения</b>	<b>2</b>	
	1. Классы в ООП: виды, назначение, свойства, методы, события.	<b>1</b>		

	2. Перегрузка методов.			
	<i>Тематика практических занятий</i>	<b>2</b>	<b>2</b>	
	Объявления класса. Создание наследованного класса. Программирование приложений.			
	<i>Самостоятельная работа обучающихся</i>		---	
<b>Всего:</b>			<b>156</b>	

## ВЫВОД ПО ГЛАВЕ 1

Педагогическое тестирование как форма контроля знаний является неотъемлемой частью учебного процесса. Оно имеет ряд преимуществ, позволяющее ему стать одним из основных способов контроля знаний студентов.

При написании главы, нами были рассмотрены различные понятия связанные с методом тестирования. Мы выделили виды тестирования, дали их описание, рассмотрели преимущества каждого. Рассмотрели на какие типы, в свою очередь, они разделяются. Описали основные формы тестовых заданий. Дали характеристику, выбора формы теста для различных задач тестирования. Выделили принципы разработки тестов, дали характеристику составления инструкций по выполнению тестов.

Обозначили основные функции программ автоматизированного контроля знаний, дали их описание. Определили последовательные шаги для построения компьютерных тестов, и требования к тестовым оболочкам.

Рассмотрели существующую документацию по дисциплине «Основы алгоритмизации и программирования».

## **ГЛАВА 2. РАЗРАБОТКА ЭЛЕКТРОННОЙ ТЕСТОВОЙ ОБОЛОЧКИ ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» КАК СРЕДСТВО КОНТРОЛЯ УЧЕБНЫХ ДОСТИЖЕНИЙ СТУДЕНТОВ КОЛЛЕДЖА**

### **2.1 Выбор среды разработки для электронной тестовой оболочки**

Средства, входящие в состав Visual Studio для Windows, можно использовать для создания привлекательных инновационных приложений Windows 7, Windows 8, Windows 10.

Эти средства включают полнофункциональный редактор кода, мощный отладчик, специальный профилировщик и широкие возможности языковой поддержки, которые позволяют выполнять сборку приложений, написанных на языках HTML5/JavaScript, C++, C# и Visual Basic.

В качестве языка программирования был выбран C#.

C# — язык программирования, сочетающий объектно-ориентированные и контекстно-ориентированные концепции. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET.

В последнее время в общем объеме вновь создаваемого программного обеспечения различного назначения увеличивается доля .NET-приложений – программ, ориентированных на платформу Microsoft .NET. Это объясняется, прежде всего, новыми возможностями, которые предоставляет платформа прикладным программам, а также тем, что технология .NET поддерживается новейшими операционными системами.

Microsoft .NET – это технология, в основе которой лежит идея универсального программного кода, который может быть выполнен любым компьютером, в не зависимости от используемой операционной системы. Универсальность программного кода обеспечивается за счет предварительной (выполняемой на этапе разработки) компиляции исходной

программы в универсальный промежуточный код (CIL –код, Common Intermediate Language), который во время загрузки транслируется в выполняемый. Преобразование промежуточного кода в выполняемый осуществляет JIT-компилятор (от Just In Time – в тот же момент, «на лету»), являющийся элементом виртуальной выполняющей системы (Virtual Execution System, VES). Работу .NET-приложений в операционной системе Windows обеспечивает Microsoft .NET Framework.

Среда разработки Microsoft Visual C# является инструментом разработки .NET приложений для Windows. В ней интегрированы удобный дизайнер форм, специализированный редактор кода, отладчик, мастер публикации и другие инструменты, необходимые программисту.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML. Переняв многое от своих предшественников — языков C++, Delphi, Modula и Smalltalk — C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++) или вывода типов.

На рисунках 2.1. и 2.2 изображён интерфейс Microsoft Visual Studio при использовании языка C#.

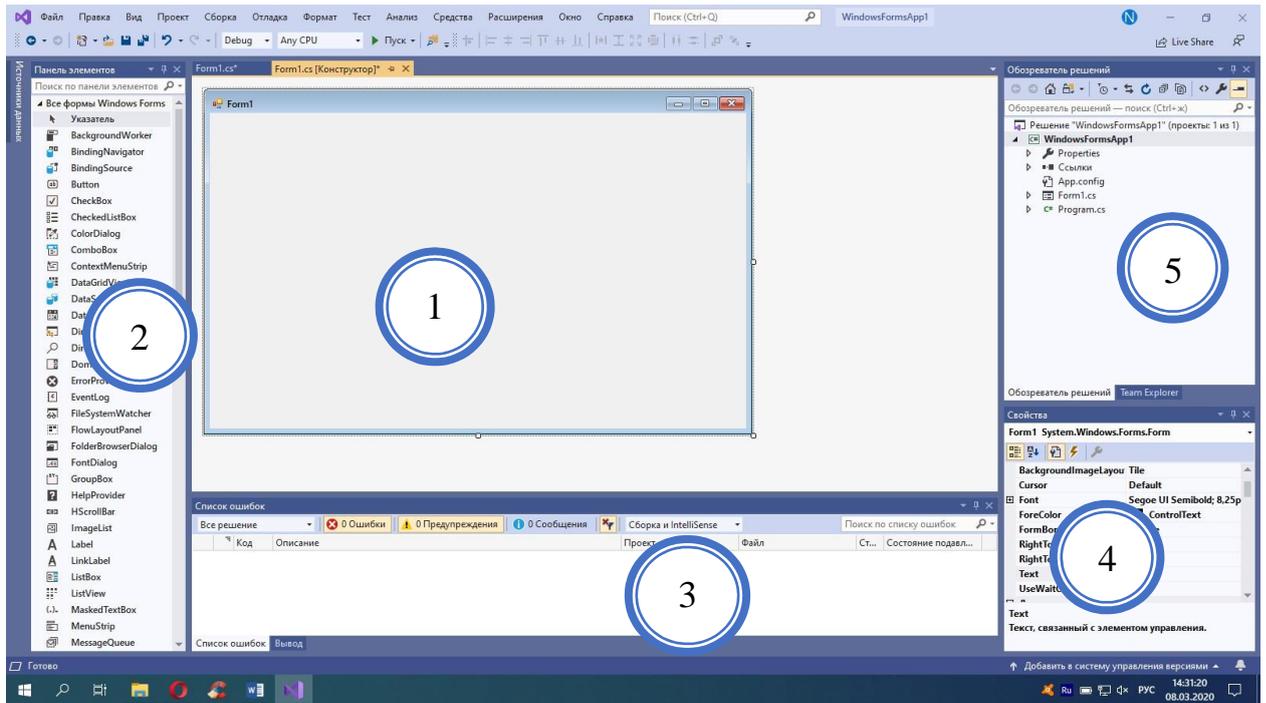


Рисунок 2.1 – Интерфейс Microsoft Visual Studio при использовании языка C#

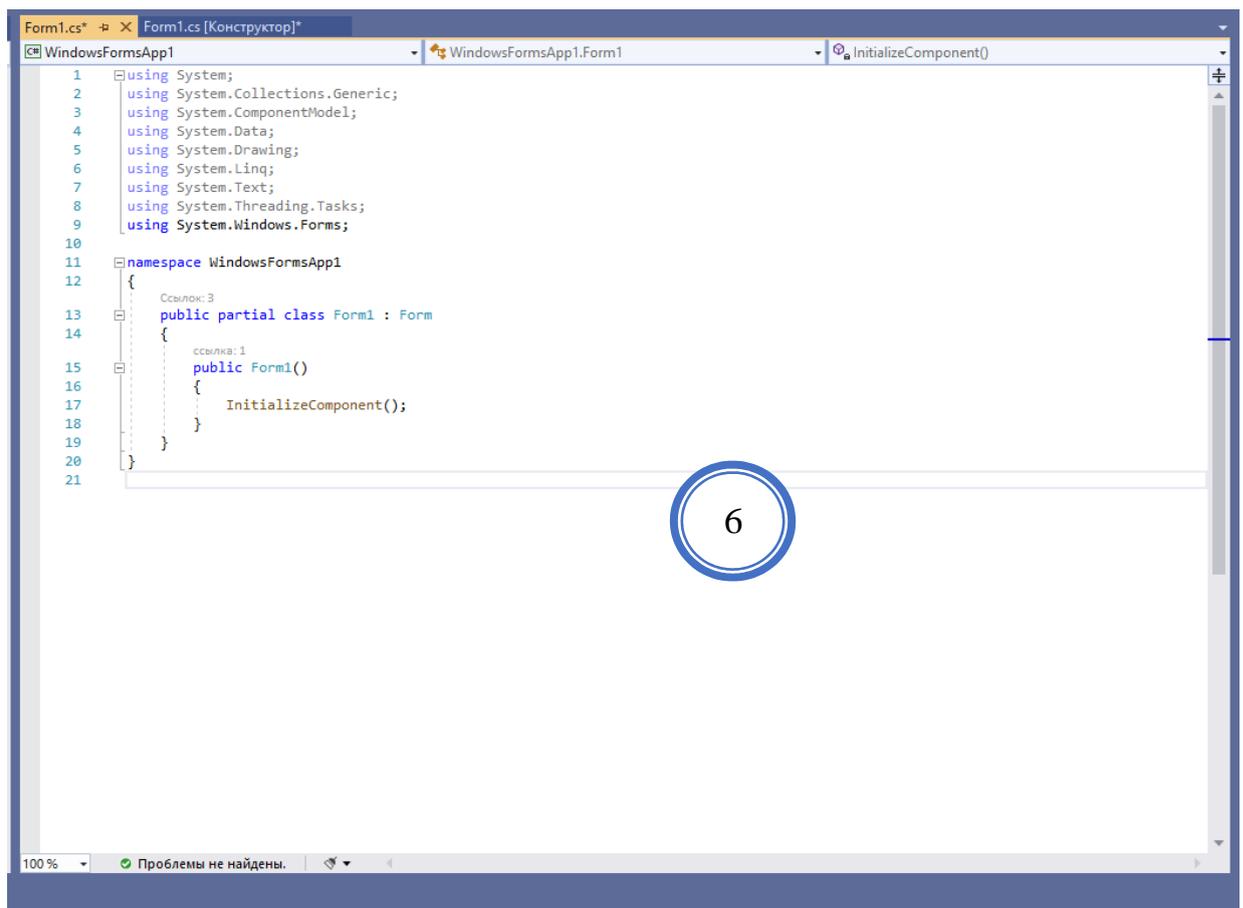


Рисунок 2.2 – Интерфейс Microsoft Visual Studio при использовании языка C#

На рисунках представлены основные части интерфейса:

1. Главная форма, открываемая при создании нового проекта.
2. Панель элементов, которые программист может использовать на главной форме.
3. Панель вывода сообщений и ошибок.
4. Окно свойств.
5. Окно обозревателя решений.
6. Окно программного кода.

Синтаксис языка C#:

1. Комментарии.
2. Переменные.
3. Управляющие операторы.
4. Операторные скобки.

Комментарии бывают двух видов: однострочные и многострочные.

Однострочные комментарии отображаются как двойной слеш (//).

Многострочные комментарии имеют вид слеш звёздочка - звёздочка слеш (/\* ... \*/).

Переменная — это идентификатор, содержащий какое-либо значение. Переменная определяется типом и именем, а также может инициализироваться при объявлении. Переменные бывают целочисленные, вещественные, символьные, строковые и логические.

Управляющими операторами являются операторы циклы и выбора.

Операторы цикла: while ... do, do ... while, for.

Операторы выбора: if, if ... else ..., switch.

Операторные скобки ({...}) используются для обозначения начала и конца блоков кода, а также обозначения области видимости. Идентификатор, объявленный внутри фигурных скобок, видим только внутри них и во вложенных областях видимости.

Данная среда программирования была выбрана, потому что она имеет понятный интерфейс, широкий набор инструментов для разработки

приложений который расширяется. Возможна интеграция со многими языками программирования и программами, быстро компилирует программный код и выявляет ошибки.

Для выбора тестовой оболочки принимались во внимание следующие критерии:

1. Возможность работы через сеть Интернет.
2. Возможность добавления текста, формул, графиков и рисунков.
3. Задание вопросов и вариантов ответов случайным образом.
4. Разнообразные типы вопросов.
5. Ограничение по времени (для режима «контроль»).
6. Вывод результата (в процентном соотношении правильных и не правильных ответов).
7. Неограниченное количество вопросов.

#### *Let's test.*

Система тестирования Let's test позволяет проводить онлайн тестирования знаний через интернет. Она является не просто конструктором тестов, а обладает широким набором функциональных возможностей, благодаря которым можно построить целую инфраструктуру для вашей организации.

Let's test позволяет построить систему проверки знаний с помощью тестов, затратив при этом минимум усилий.

Для управления системой тестирования не нужно привлекать ни системных администраторов, ни программистов. Заниматься обслуживанием может рядовой сотрудник с навыками пользователя ПК.

#### *Google формы.*

Google формы – один из типов документов, доступных на Google Docs. Бесплатно можно создавать неограниченное количество опросов, анкет, тестов и приглашать неограниченное количество респондентов. Для создания опросов пользователю обязательно необходим аккаунт Google.

Позволяет создавать форму с различными элементами или типами вопросов. Любой вопрос можно сделать обязательным для ответа. В процессе создания формы можно легко изменять порядок вопросов. Для каждой созданной формы можно выбрать дизайн для её оформления. Ссылка на форму генерируется автоматически после её создания.

Приглашение к участию в опросе:

- электронным письмом списку адресатов, включив в него ссылку на опрос или саму форму;
- разместив форму (ссылку на форму) на сайте или блоге.

Большим минусом является то, что один и тот же респондент, работая с анкетой или опросом, может давать ответы неограниченное количество раз и отследить это невозможно.

Для каждого опроса автоматически создается таблица результатов в формате Excel в Google Docs. Все полученные ответы тут же отображаются в ней. Таблицу можно экспортировать в различные форматы: pdf, xls, txt, но в таблице ответов нельзя фильтровать результаты. Пользователю, создавшему опрос или анкету, в любой момент доступна сводка опроса с диаграммами по каждому вопросу.

### *Simpoll*

Simpoll — сервис для проведения тестирований, голосований, создания анкет и других форм. Генерирует код, который можно добавить на свой сайт. Также платформа позволяет публиковать опросы в соцсетях и давать прямые ссылки на них.

В Simpoll можно ограничивать сроки проведения опросов по дате или количеству респондентов. Защита от любителей повторного прохождения тоже есть — по IP-адресу и по cookies.

Формы здесь создаются при помощи перетаскивания различных типов полей в окно редактора. Полей всего 11: выбор одного или нескольких вариантов, ввод ответа в одну строку или в несколько, открытый и

выпадающий список, шкала оценки, имя и e-mail респондента, дата, загрузка файла и адрес.

Внешний вид форм можно изменять. Цвета, шрифты, отступы для любого типа элементов — все меняется. Возможность предпросмотра имеется.

Кроме того, в опросы можно добавлять соавторов. Ими могут быть зарегистрированные в Simpoll пользователи, которые имеют право только на просмотр результатов или на редактирование вопросов. После добавления соавтор увидит вашу форму в своем личном кабинете и сможет в ней работать.

#### *Online Test Pad.*

Online Test Pad - многофункциональный веб-сервис, разработанный для создания опросников, кроссвордов, логических игр и комплексных заданий. Программа работает в вебе, поэтому доступна со всех современных устройств. Софт также предлагает создать мини-сайт - площадку, на которой можно организовать процесс тестирования респондентов.

Платформу можно использовать преподавателям для тестирования учеников и студентов, проведения экспресс-проверок уровня знаний, зачётов и контрольных работ. Система также пригодится для сбора информации с сайта компании, независимо от её масштаба и сферы деятельности. Опции создания онлайн-кроссворда пригодятся редакторам и копирайтерам в СМИ, а также могут быть использованы в развлекательных и личных целях.

В конструкторе тестов предусмотрено большое количество различных настроек тестов. Функции онлайн-конструктора позволяют реализовать в онлайн тест любого уровня сложности, провести опросы, протестировать сотрудников и собрать статистику.

Конструктор тестов предусматривает варианты 17 типов вопросов, в том числе: одиночный выбор, множественный выбор, ввод числа, ввод текста, ответ в свободной форме, установление последовательности,

установление соответствий, заполнение пропусков, интерактивный диктант.

Доступ к тесту может быть создан по основной ссылке или в виде виджета на сайте организации, а также в качестве общедоступной публикации на сайте Online Test Pad.

Система предоставляет возможность не только точных ответов, но и заполнения в свободной форме (в этом случае ответы направляются администратору для личной проверки).

Доступен просмотр каждого результата, статистики ответов и набранных баллов по каждому вопросу, статистики по каждому результату. В табличном виде представлены все результаты, регистрационные параметры, ответы на все вопросы, которые можно сохранить в Excel.

Сравнение онлайн конструкторов тестов приведено в таблице ниже.

Таблица 2.1 – Сравнение онлайн конструкторов тестов

Критерии	Let's test	Simpoll	Google формы	Online Test Pad
1. Возможность работы через сеть Интернет	+	+	+	+
2. Возможность добавления рисунков	+	+	+	+
3. Задание вопросов и вариантов ответов случайным образом	-	+	-	+
4. Типы вопросов	— выбор одного правильного ответа; — выбор нескольких правильных ответов; — ввод текстового ответа; — установка последовательности	— выбор одного варианта из списка; — выбор нескольких вариантов из списка; — ввод строк и подробных комментариев	— текст (строка) — текст (абзац) — один из списка — несколько из списка — раскрывающийся список	— одиночный выбор; — множественный выбор; — ввод числа; — ввод текста; — ответ в свободной форме; — установление последовательности; — установление соответствий; заполнение пропусков
5. Ограничение по времени (для режима «контроль»)	-	-	-	+
6. Вывод результата (в процентном соотношении правильных и не правильных ответов)	+	-	-	+
7. Неограниченное количество вопросов	+	+	+	+

Выбор был сделан в пользу онлайн конструктора тестов Online Test Pad.

## 2.2 Структура и этапы разработки электронной тестовой оболочки

Данный мультимедийный образовательный ресурс помогает управлять умственной работой студентов в процессе усвоения учебного материала и решения задач разного вида – типовых, комплексных, нестандартных.

Запустив программу нашему вниманию представлена название электронной тестовой оболочки и его содержание (рисунок 2.3).

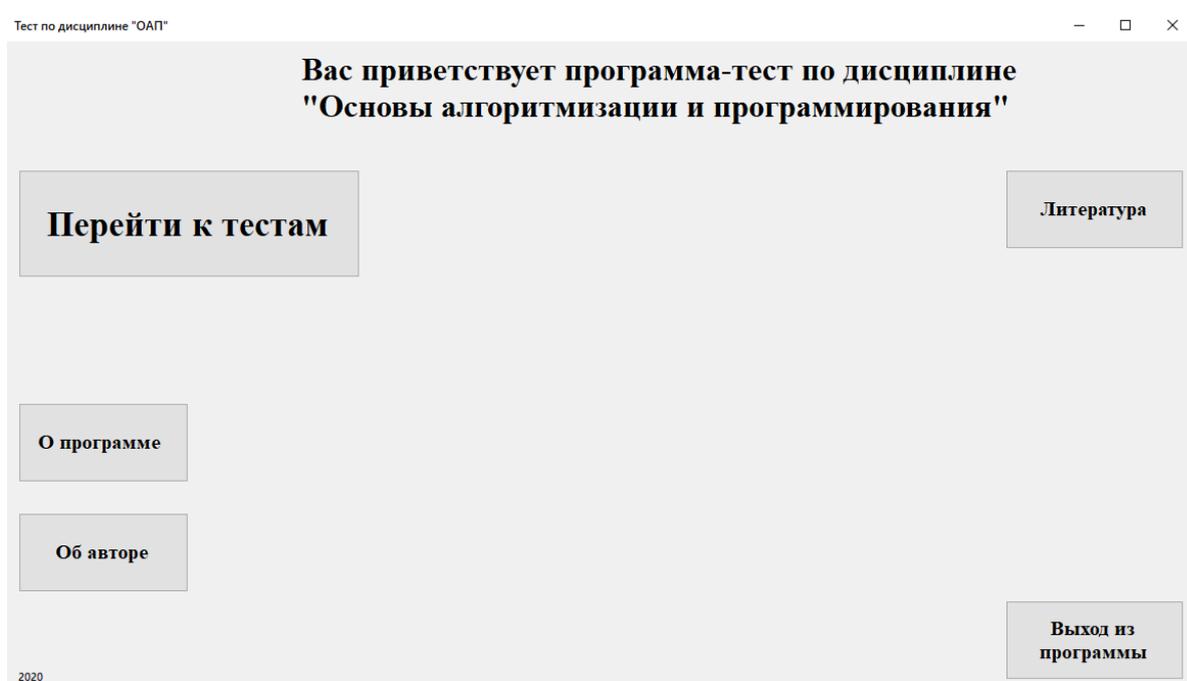


Рисунок 2.3 – Содержание

Нажимая на кнопку «Об авторе», откроется новое окно с информацией о создателе программного продукта (рисунок 2.4).

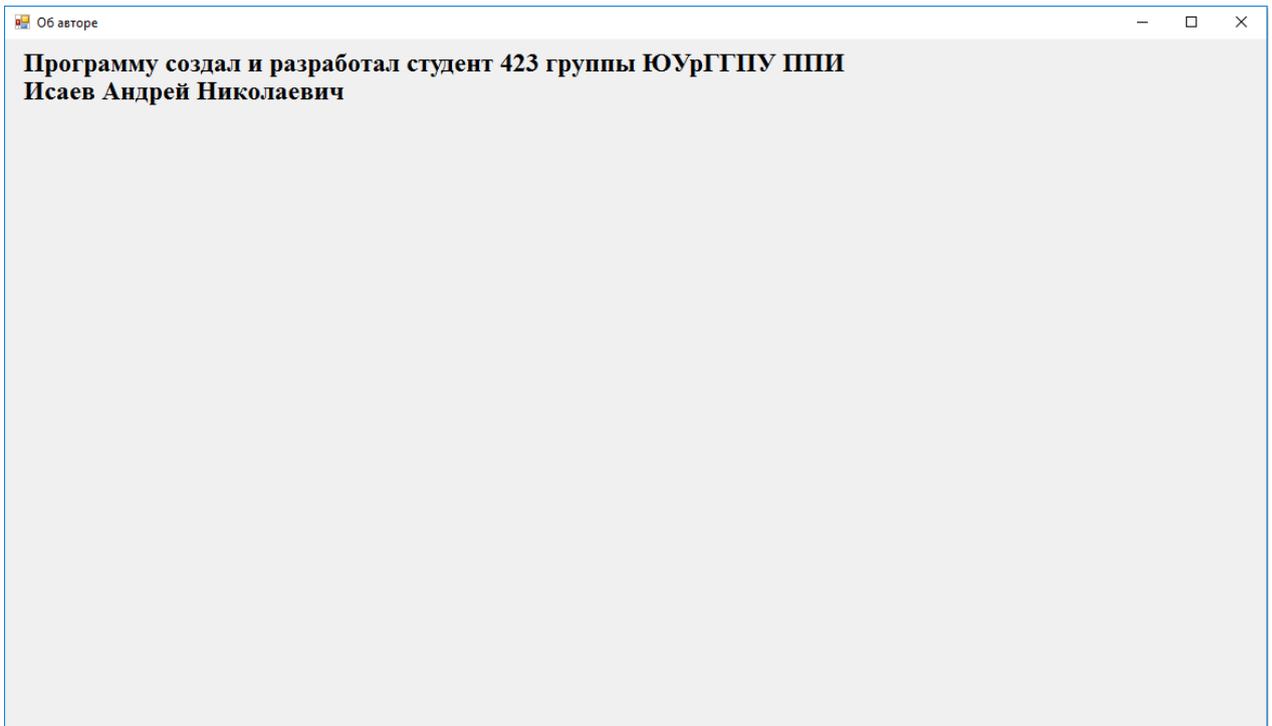


Рисунок 2.4 – Окно «Об авторе»

Кнопка «О программе» (рисунок 2.5) содержит сведения о программе.

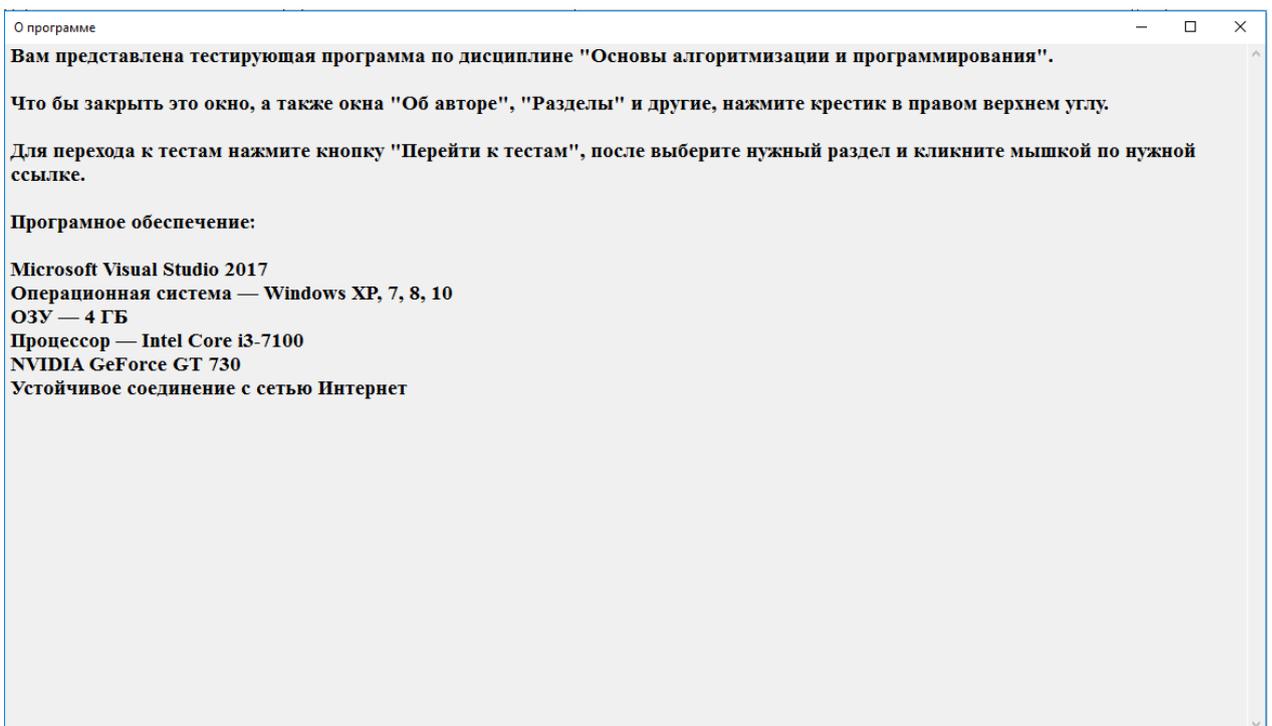


Рисунок 2.5 – Кнопка «О программе»

Кнопка «Литература» (рисунок 2.6) позволяет перейти к списку использованной литературы.

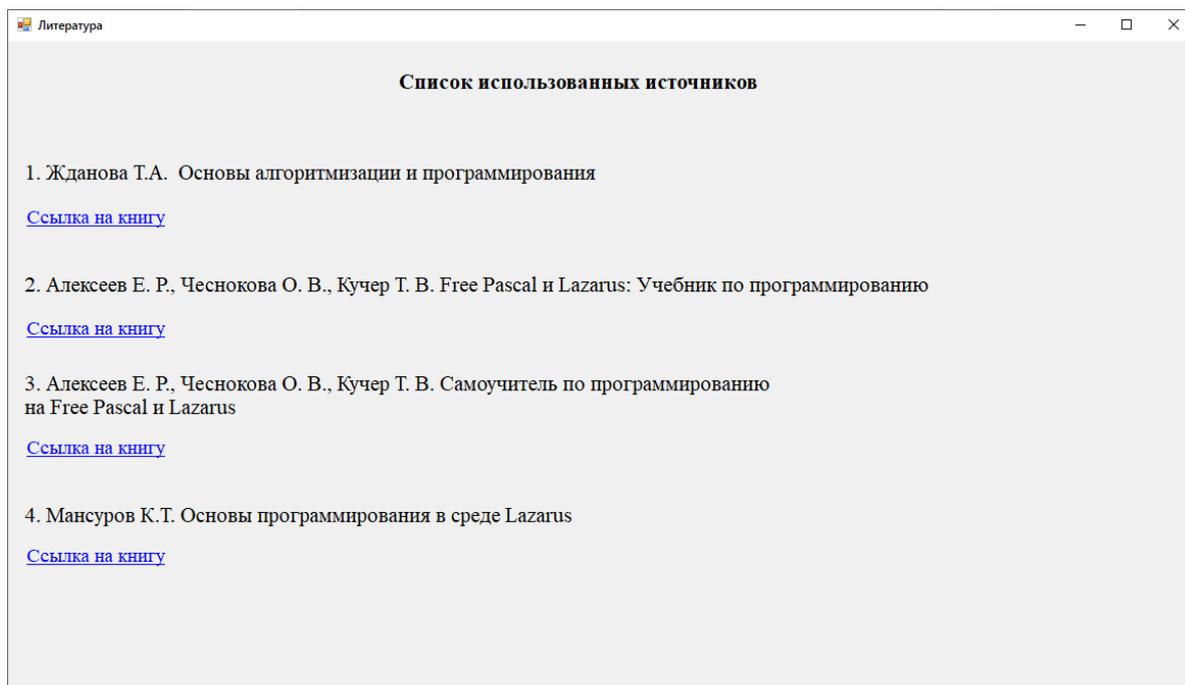


Рисунок 2.6 – Кнопка «Литература»

Все переходы реализованы в виде гиперссылок, при нажатии на выделенный текст, в браузере откроется интернет страница с книгой.

Кнопка «Перейти к тестам» (рисунок 2.7) позволяет выбрать раздел с тестами.

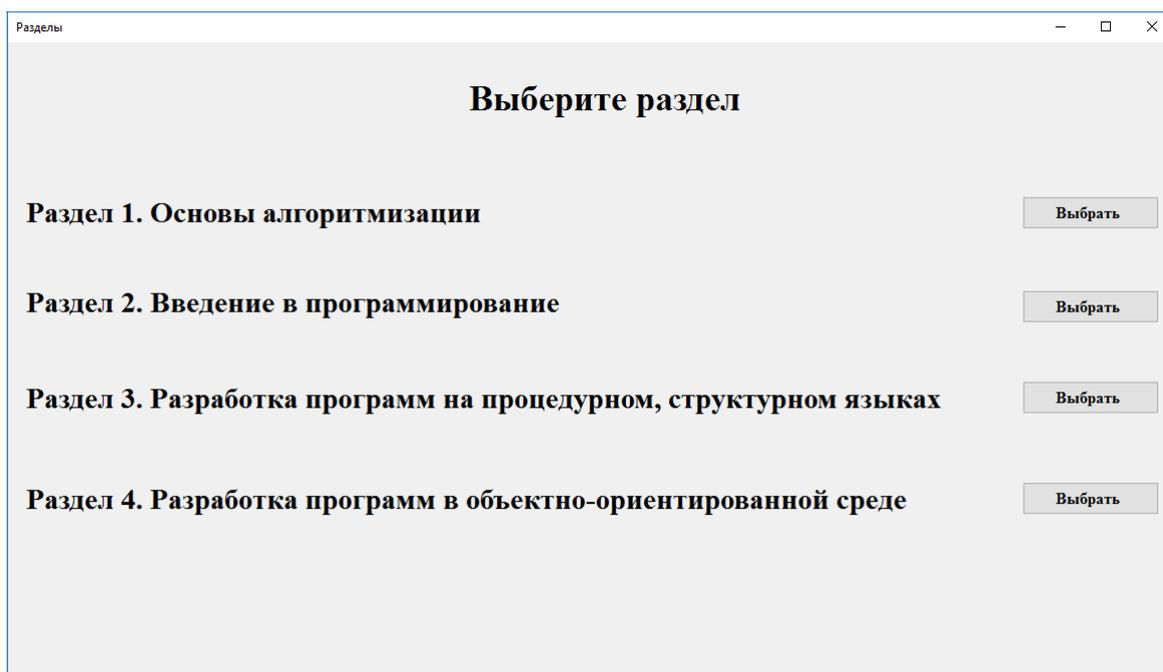


Рисунок 2.7 – Кнопка «Перейти к тестам»

Нажимая кнопку «Выбрать» напротив Раздела 1, мы перейдём к выбору темы (рисунок 2.8).

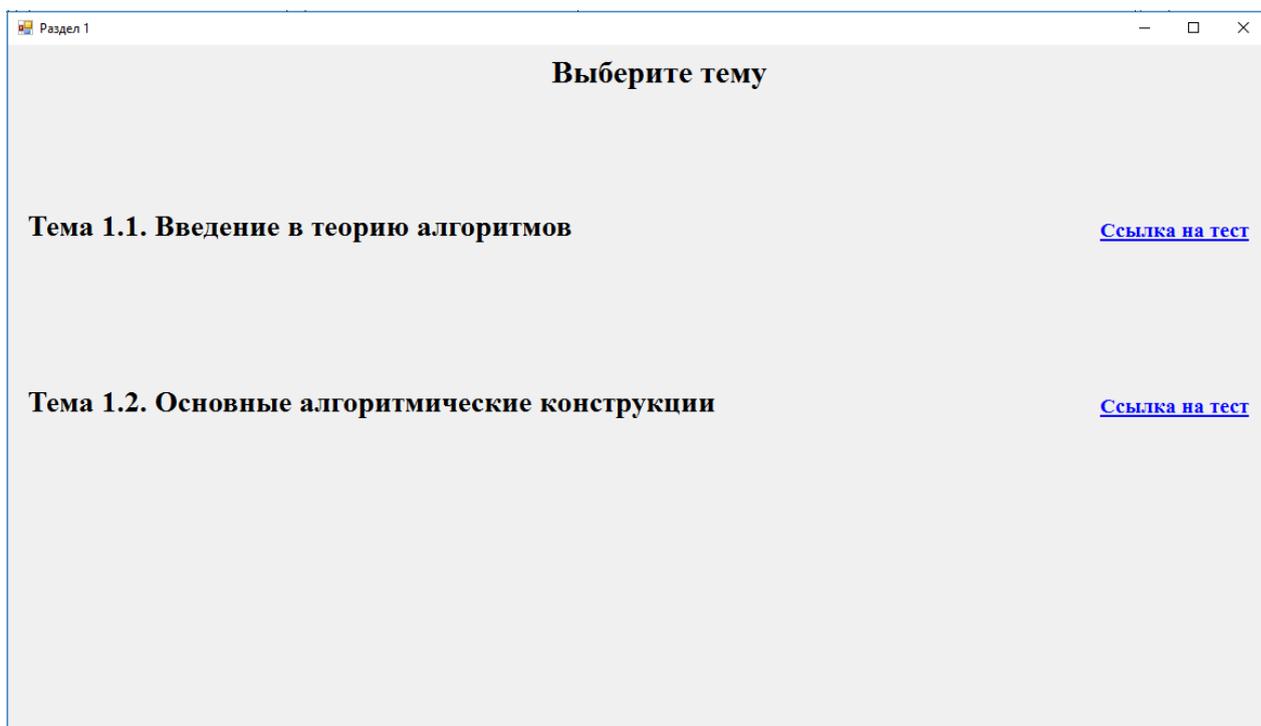


Рисунок 2.8 – Выбор темы

Нажимая на текст «Ссылка на тест» напротив Темы 1.1., мы перейдём к странице теста (рисунок 2.9). Все переходы реализованы в виде гиперссылок, при нажатии на выделенный текст, в браузере откроется интернет страница с тестом.

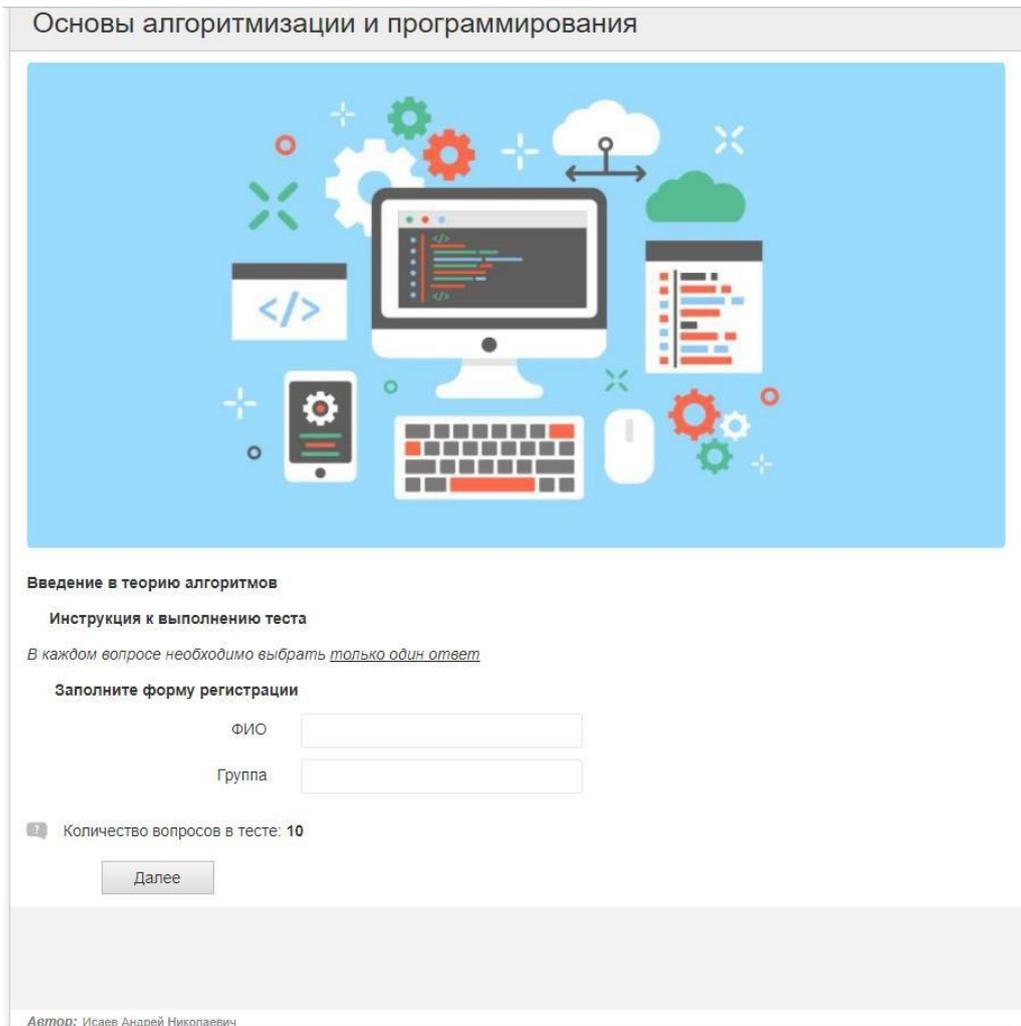


Рисунок 2.9 – Страница теста

Заполнив форму регистрации, мы переходим к самому тесту (рисунок 2.10).

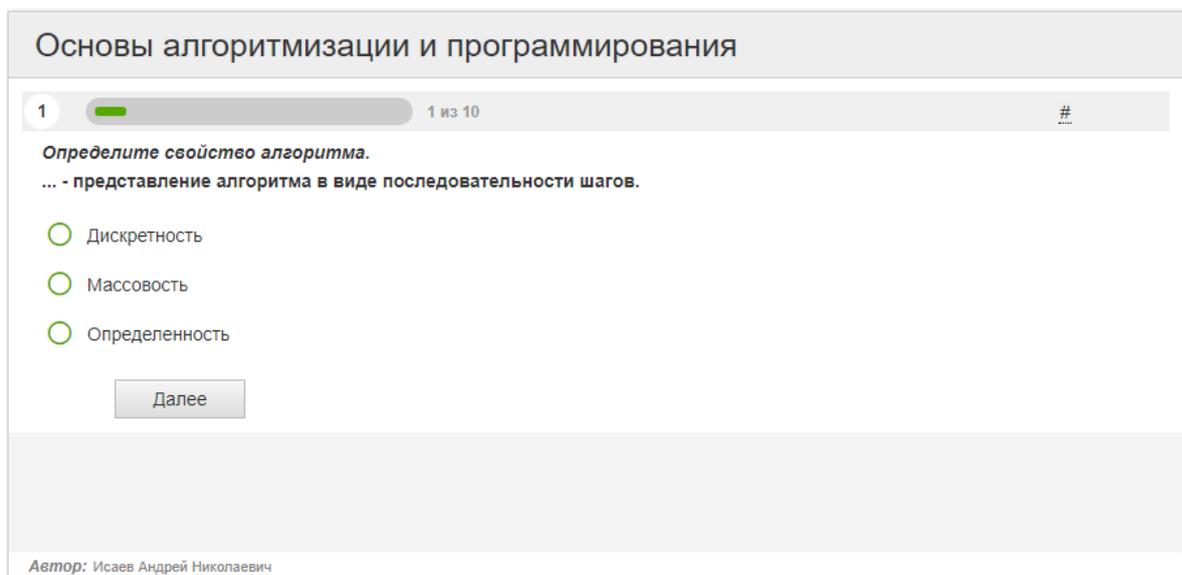


Рисунок 2.10 – Тест

После прохождения теста мы переходим на страницу результата, который отображается в процентном и балльном отношении (рисунок 2.11).

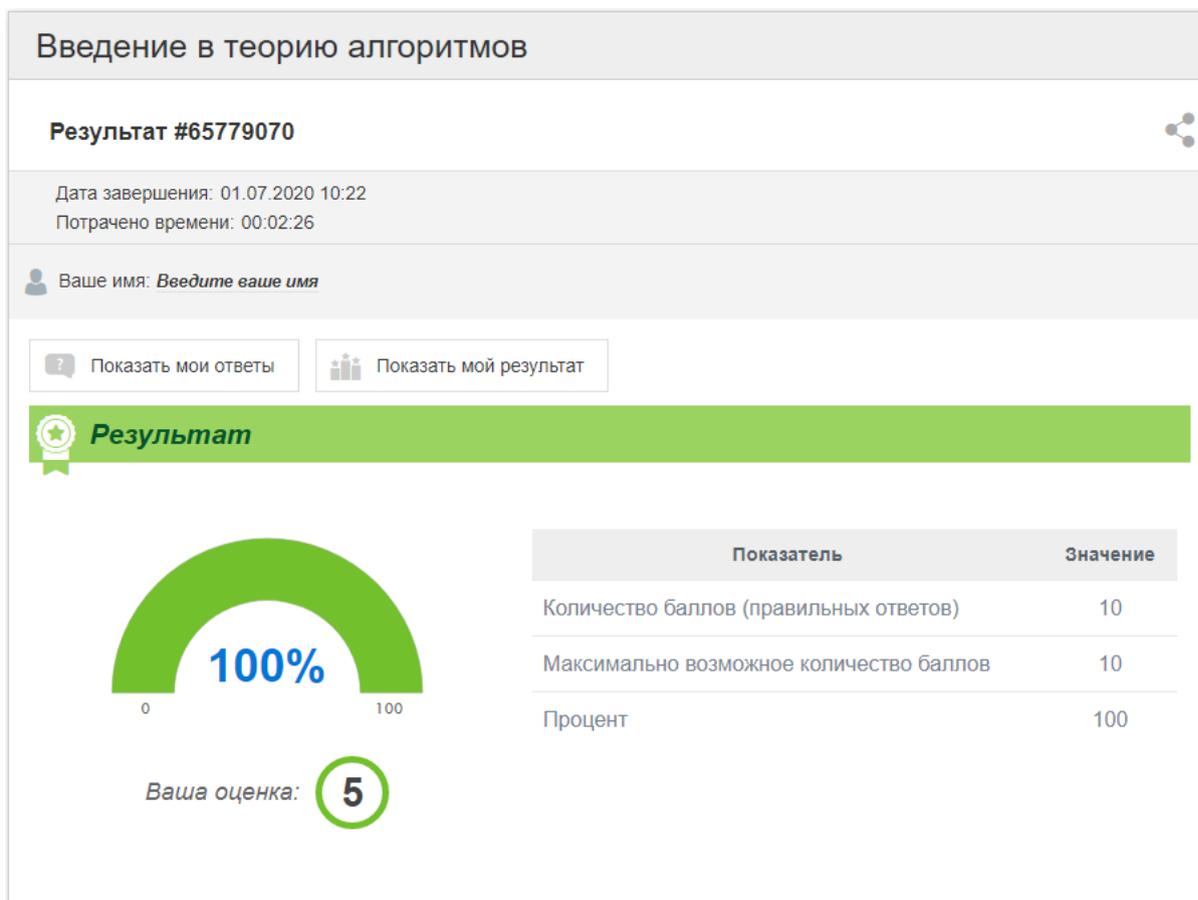


Рисунок 2.11 – Результат

Разработанная электронная тестовая оболочка является одним из вариантов решения проблемы контроля успеваемости студентов по дисциплине «Основы алгоритмизации и программирования» по специальности 09.02.07 Информационные системы и программирование студентами учреждений среднего профессионального образования.

### 2.3 Анализ результатов исследования разработки электронной тестовой оболочки

Проверка результатов эффективности форм контроля по дисциплине профессионального цикла проходила на базе ГБПОУ «Южно-Уральский государственный колледж» в группе ИС 248Д, обучающейся по специальности: «Информационные системы и программирование».

Цель эксперимента: определить эффективность применения тестового контроля по дисциплине: «Основы алгоритмизации и программирования».

Задачи обучающего эксперимента:

- 1) разработать тестовые задания по дисциплине «Основы алгоритмизации и программирования»;
- 2) применить тестовый контроль знаний после изучения дисциплины «Основы алгоритмизации и программирования»;
- 3) оценить эффективность применения тестового контроля знаний.

Путем анализа проверки теоретических и практических знаний и умений в области дисциплины: «Основы алгоритмизации и программирования» были определены: одна подгруппа экспериментальная, в которой проверка итоговых знаний предполагалась в инновационной тестовой форме и контрольная, проходившая оценку итоговых знаний в традиционной устной форме (по 13 студентов в каждой группе).

Организация эксперимента:

Первый этап. Преподаватель разрабатывает систему тестовых заданий по дисциплине «Основы алгоритмизации и программирования».

Второй этап. В экспериментальной подгруппе из 13 человек студентов проводился контроль знаний по тестовым заданиям по вышеуказанной дисциплине в компьютерной форме. Преподаватель использовал специальную программу для тестового контроля знаний.

Третий этап. Анализ результатов.

Для любой образовательной системы, в том числе и для учреждения среднего профессионального образования, оценка знаний студентов является одной из основных и сложных задач. Методом исследования уровня знаний, умений, навыков, способностей и других качеств личности является такая форма контроля как тест. В последние годы тестирование, как метод оценки знаний, приобретает всё большую популярность. Вместе с традиционными формами тестирования широкое применение получило

компьютерное, что соответствует общей концепции модернизации и компьютеризации российской системы образования.

Наглядность, которую дают дисплейные формы представления контрольно-оценочных мероприятий, позволяет каждому студенту понять степень усвоения знаний по данному разделу дисциплины.

Эксперимент проходил в два этапа:

1 этап – начальный, на котором проводилось исследование уровня теоретических знаний по изучаемой дисциплине.

2 этап – итоговый.

Со студентами, участвовавшими в экспериментальной работе, проводились планомерные срезы знаний по теории (в виде опроса) и в области оценки познавательной деятельности.

Принципиально важным является выбор критериев и показателей усвоения знаний. На основе анализа состояния проблемы исследования в педагогической теории и практики, разработки методики оценки знаний студентов по технологии был выделен критерий общетеоретической подготовки студентов как усвоение технических знаний. Оценка усвоения знаний проводилась в тестовой форме.

В ходе изучения психолого-педагогической литературы, наблюдения за студентами, анкетирования, анализа ответов и выполненных практических заданий была выбрана порядковая шкала и выделено условно 3 уровня каждого критерия общетеоретической подготовки: высокий, средний, низкий:

1. Высокий уровень общетеоретической подготовки, которому соответствует активное владение знаниями и их постоянное применение;
2. Средний уровень – недостаточное владение технологическими знаниями и ситуативное их применение;
3. Низкий уровень – отсутствие у студента технических знаний.

Для количественного анализа результатов каждому уровню присваивался определенный балл: высокому – 3 балла, среднему – 2 балла,

низкому – 1 балл. На протяжении эксперимента проводились контрольные срезы с целью определения правильности хода эксперимента и уровня общетеоретической подготовки у студентов экспериментальной и контрольной группы.

Планировалось два контрольных среза по выявлению уровня усвоения общетеоретической подготовки студентов (начальный и итоговый).

Состояние проблемы исследования на практике и определение уровня общетеоретической подготовки на момент начала эксперимента отражает начальный срез. Значения средних баллов и стандартных отклонений очень близки по подгруппам, что позволяет делать вывод о неразличимости подгрупп на начальном срезе (Таблица 2.2).

После изучения дисциплины «Основы алгоритмизации и программирования» контроль полученных студентами знаний проводился следующим образом: в экспериментальной подгруппе - по разработанным тестам с использованием компьютерной программы, а в контрольной подгруппе – в виде устного опроса.

Таблица 2.2 - Уровень усвоения знаний ОАиП студентами на начальном этапе эксперимента

Подгруппа	Уровни усвоения знаний ОАиП					
	Высокий		Средний		Низкий	
	Кол-во чел.	%	Кол-во чел.	%	Кол-во чел.	%
КГ	3	25,0	4	33,3	5	41,7
ЭГ	3	23,1	4	30,8	6	46,2

Эксперимент по разработке и применению тестового контроля знаний в процессе преподавания дисциплины профессионального цикла показал, что в экспериментальной подгруппе большинство студентов смогли ответить на поставленные в тестах вопросы. Многие студенты, из числа опрашиваемых, благодаря четко поставленным вопросам в тестовых

заданиях смогли показать хорошие знания по изучаемой дисциплине

(Таблица 2.3)

Таблица 2.3 - Уровень усвоения знаний ОАиП студентами на итоговом этапе эксперимента

Подгруппа	Уровни усвоения знаний ОАиП					
	Высокий		Средний		Низкий	
	Кол-во чел.	%	Кол-во чел.	%	Кол-во чел.	%
КГ	2	16,7	6	50,0	4	33,3
ЭГ	4	33,3	7	58,3	1	8,4

В целом результаты исследования показали, что применение метода тестирования в качестве итоговой оценки значительно повышает уровень познавательной активности студентов, что напрямую отражается на уровне общетеоретической подготовки.

По результатам экспериментальной проверки можно судить об эффективности применения электронной тестовой оболочки на занятиях как средства контроля учебных достижений студентов колледжа.

## ВЫВОД ПО ГЛАВЕ 2

Все больше и больше в нашем информационном обществе развивается тестирование. Его используют при сдаче экзаменов при окончании школы, в средних и высших учебных заведениях, в психологических тестах, при приемах на работу и т.д., поэтому автоматизация этого процесса просто необходима. Это решает массу проблем и отмечается ряд преимуществ использования тестирования, например, отсутствие влияния человеческого фактора, значительная экономия времени тестирующего, сокращение времени контроля и т.д.

К преимуществам программы можно отнести следующие пункты:

1. Абстрагирование преподавателя от процесса непосредственной проверки работ студентов.
2. Увеличение эффективности процесса оценки знаний студента.
3. Возможность задействования к процессу сдачи любого количества студентов.
4. Преподаватель не ограничивается количеством внесенных вопросов в тестовые задания.
5. Значительное уменьшение времени преподавателя на проверку результата.
6. Уменьшение времени студента на контрольную работу.

В данном проекте была предоставлена реализация организации тестирования знаний студентов с возможностью оценки их знаний по количеству верных ответов.

## ЗАКЛЮЧЕНИЕ

Все больше и больше в нашем информационном обществе развивается тестирование. Его используют при сдаче экзаменов при окончании школы, в средних и высших учебных заведениях, в психологических тестах, при приемах на работу и т.д., поэтому автоматизация этого процесса просто необходима. Это решает массу проблем и отмечается ряд преимуществ использования тестирования, например, отсутствие влияния человеческого фактора, значительная экономия времени тестирующего, сокращение времени контроля и т.д.

В ходе выполнения выпускной квалификационной работы, мы дали определение понятию электронная тестовая оболочка. Для этого мы рассмотрели такие понятия как, педагогическое тестирование, тест, тестовое пространство, полный тест, эффективный тест, выявили преимущества компьютерного тестирования. Создание теста предполагает тщательный анализ содержания учебной дисциплины, классификацию учебного материала, установление межпредметных связей, укрупнение дидактических единиц с последующим представлением этих единиц через элементы композиции задания. Также мы выявили состав и структуру теста, установили виды тестовых заданий, нашли особенности каждого вида теста. Определили две основные формы тестовых заданий, это задания с закрытой формы и задания открытой формы, каждая из форм позволяет проверить специфические виды знаний. Дали определение понятию тестовые оболочки, выявили задачи, которые решаются при их применении.

Обозначили основные функции программ автоматизированного контроля знаний, дали их описание. Определили последовательные шаги для построения компьютерных тестов, и требования к тестовым оболочкам.

Во время написания дипломной работы, нами была проанализирована рабочая программа учебной дисциплины, специальности 09.02.07 «Информационные системы и программирования». Были выделены общие

и профессиональные компетенции, а также что должен уметь и знать студент после успешного усвоения программы дисциплины «Основы алгоритмизации и программирования». Указали количество часов, выделяемых на освоение дисциплины. Рассмотрели тематический план и содержание учебной дисциплины.

При создании электронной тестовой оболочки, нами был проведён анализ различных сред программирования и языков программирования. Были выявлены преимущества и недостатки каждой из сред программирования. В конечном итоге выбор был сделан в пользу среды разработки Microsoft Visual C#. Она обладает современными средствами разработки программных продуктов и имеет ряд других, необходимых при создании приложения, преимуществ. Описали основные части интерфейса Visual Studio и синтаксис языка C#. Обозначили критерии выбора тестовой оболочки. Были рассмотрены следующие онлайн-конструкторы тестов:

1. Let's test.
2. Google формы.
3. Simpoll.
4. Online Test Pad.

Дали описание каждого конструктора, рассмотрели их преимущества и недостатки. В итоге нами была представлена сравнительная таблица онлайн конструкторов тестов. Выбор был сделан в пользу онлайн конструктора тестов Online Test Pad.

В ходе выполнения работы, мы разработали структуру и содержание электронной тестовой оболочки по дисциплине «Основы алгоритмизации и программирования». Сделали описание каждого элемента программы. Описали минимальные системные требования к программному продукту.

Следующим этапом написания выпускной квалификационной работы, был проведён анализ результатов применения электронной тестовой оболочки на базе ГБПОУ «Южно-Уральский государственный колледж» в группе ИС 248Д, обучающейся по специальности: «Информационные

системы и программирование». Для этого мы сформулировали цель эксперимента и поставили задачи. Эксперимент проходил в два этапа, в результате которых мы пришли к следующему выводу, что по результатам экспериментальной проверки можно судить об эффективности применения электронной тестовой оболочки на занятиях как средства контроля учебных достижений студентов колледжа.

Таким образом, во время выполнения работы, все задачи были решены, а цель достигнута.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аванесов В.С. Теория и методика педагогических измерений: URL: <http://testolog.narod.ru/Theory21.html> (дата обращения: 25.03.2020).
2. Алексеев, Е.Р. Free Pascal и Lazarus: учебник по программированию / Е.Р. Алексеев, О.В. Чеснокова, Т.В. Кучер. — 2-е изд. — Саратов: Профобразование, 2019.
3. Аузяк. А.Г. Программирование и основы алгоритмизации: Для инженерных специальностей технических университетов и вузов / А.Г. Аузяк, Ю.А. Богомолов, А.И. Маликов, Б.А. Старостин. Казань: Изд-во Казанского национального исследовательского технического ун-та - КАИ, 2015, 153 с.
4. Ахмедханлы, Д.М. Основы алгоритмизации и программирования: электрон. учеб.-метод. пособие / Д.М. Ахмедханлы, Н.В. Ушмаева. – Тольятти: Изд-во ТГУ, 2016.
5. Ачкасов, В.Ю. Программирование на Lazarus / В.Ю. Ачкасов. — 2-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.
6. Большая Советская Энциклопедия: URL: <http://bse.scilib.com/article110297.html> (дата обращения: 25.03.2020).
7. Боровкова Т.И., Морев И.А. Мониторинг развития системы образования. Часть 1. Теоретические аспекты: Учебное пособие. — Владивосток: Изд-во Дальневосточного университета, 2015.
8. Википедия URL: [https://ru.wikipedia.org/wiki/Педагогическое\\_тестирование](https://ru.wikipedia.org/wiki/Педагогическое_тестирование) (дата обращения: 25.03.2020).
9. Волобуева, Т.В. Информатика. Основы программирования на языке Pascal: учебное пособие / Т.В. Волобуева. — Воронеж: Воронежский государственный архитектурно-строительный университет, ЭБС АСВ, 2019.

10. Горелов, С.В. Современные технологии программирования: разработка Windows-приложений на языке С#. В 2 томах.: учебник / С.В. Горелов; под редакцией П. Б. Лукьянова. — Москва: Прометей, 2019.

11. Гузаева М.Ю., Компьютеризация контроля знаний как основа эффективного функционирования системы образования URL: <http://pedsovet.su/publ/44-1-0-1060> (дата обращения: 22.04.2020).

12. Жилко, Е.П. Информатика и программирование. Часть 1: учебное пособие / Е.П. Жилко, Л.Н. Титова, Э.И. Дямина. — Москва: Ай Пи Ар Медиа, 2020.

13. Использование тестов в учебном процессе URL: <http://testobr.narod.ru/3.htm> (дата обращения: 25.03.2020).

14. Коврижных, А.Ю. Основы алгоритмизации и программирования. Часть 1. Задачи и упражнения. Практикум: учебно-методическое пособие / А.Ю. Коврижных, Е.А. Конончук, Г.Е. Лузина. — Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2016.

15. Коврижных, А.Ю. Основы алгоритмизации и программирования. Часть 2. Расчетные работы. Практикум: учебно-методическое пособие / А.Ю. Коврижных, Е.А. Конончук, Г.Е. Лузина. — Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2016.

16. Лебедева, Т.Н. Технология программирования: учебное пособие / Т.Н. Лебедева, С. С. Юнусова. — 2-е изд. — Челябинск, Саратов: Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019.

17. Лебедева, Т.Н. Информатика. Информационные технологии: учебно-методическое пособие для СПО / Т.Н. Лебедева, Л.С. Носова, П.В. Волков. — Саратов: Профобразование, 2019.

18. Лебедева, Т.Н. Технология программирования: учебное пособие для СПО / Т.Н. Лебедева, С.С. Юнусова. — Саратов: Профобразование, 2019.

19. Лебедева, Т.Н. Теория и практика объектно-ориентированного программирования: учебное пособие для СПО / Т.Н. Лебедева. — Саратов: Профобразование, 2019.
20. Лубашева, Т.В. Основы алгоритмизации и программирования: учебное пособие / Т.В. Лубашева, Б.А. Железко. — Минск: Республиканский институт профессионального образования (РИПО), 2016.
21. Маматова, О.Г. Формы контроля знаний студентов педагогических вузов / О.Г. Маматова. — Текст: непосредственный // Молодой ученый. — 2012. — № 8 (43).
22. Мухаметзянов, Р.Р. Основы программирования в Delphi: учебно-методическое пособие / Р.Р. Мухаметзянов. — Набережные Челны: Набережночелнинский государственный педагогический университет, 2017.
23. Николаев, Е.И. Объектно-ориентированное программирование: учебное пособие / Е.И. Николаев. — Ставрополь: Северо-Кавказский федеральный университет, 2015.
24. Обучение программированию: язык Pascal / Н.А. Тарануха, Л.С. Гринкруг, А.Д. Бурменский, С.В. Ильина. — Москва: СОЛОН-ПРЕСС, 2016.
25. Основы алгоритмизации и программирования: лабораторный практикум / составители Е.И. Николаев. — Ставрополь: Северо-Кавказский федеральный университет, 2015.
26. Разработка приложений на C# в среде Visual Studio: учебное пособие / А.М. Нужный, Н.И. Гребенникова, В.Ф. Барабанов, О.Б. Кремер. — Воронеж: Воронежский государственный архитектурно-строительный университет, ЭБС АСВ, 2019.
27. Санников, Е.В. Курс практического программирования в Delphi. Объектно-ориентированное программирование / Е.В. Санников. — Москва: СОЛОН-ПРЕСС, 2016.

28. Саукова, Н.М. Использование систем автоматизированного контроля знаний в профессиональной деятельности педагога: учебно-методическое пособие / Н.М. Саукова, Г.Ю. Соколова, С.А. Моркин; ред. Н.М. Саукова. – Москва: Прометей, 2013.

29. Семакин И.Г. Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования / И.Г. Семакин, А.П. Шестаков. — 3-е изд., стер. — М.: Издательский центр «Академия», 2016. — 304 с.

30. Сорокин А.А. Объектно-ориентированное программирование. LAZARUS (Free Pascal): лабораторный практикум. – Ставрополь: Изд-во СКФУ, 2015.

31. Троелсен, Эндрю, Джепикс, Филипп. Язык программирования C# 6.0 и платформа .NET 4.6, 7-е изд. : Пер. с англ. М.: ООО «И.Д. Вильямс» , 2016 . - 1440 с. ил. - Парал. тит. англ.

32. Трофимов, В.В. Основы алгоритмизации и программирования: учебник для СПО / В.В. Трофимов, Т.А. Павловская; под ред. В.В. Трофимова. — М.: Издательство Юрайт, 2018. — 137 с. — (Серия: Профессиональное образование).

33. Тюльпинова, Н.В. Технология алгоритмизации и программирования на языке Pascal: учебное пособие / Н.В. Тюльпинова. — Саратов: Вузовское образование, 2019.

34. Уйманова, Н.А. Основы объектно-ориентированного программирования: практикум / Н.А. Уйманова, М.Г. Таспаева. — Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2017.

35. Уйманова, Н.А. Основы объектно-ориентированного программирования: практикум для СПО / Н.А. Уйманова, М.Г. Таспаева. — Саратов: Профобразование, 2019.

36. Хейлсберг А., Торгерсен М., Вилтамут С., Голд П. Язык программирования C#. Классика Computers Science. 4-е изд. — СПб.: Питер, 2015. — 784 с.: ил.

37. Черепахин А.А. Контроль знаний студентов: монография / А.А. Черепахин. – Москва: РУСАЙНС, 2019.