



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

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

**Электронная рабочая тетрадь по дисциплине «Основы алгоритмизации  
и программирования» как средство организации самостоятельной  
работы студентов профессиональной образовательной организации**

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

Проверка на объем заимствований:

95,17 % авторского текста

Работа рекомендована к защите


«18» февраля 2026 г.

Зав. кафедрой АТИТ и МОТД

 Руднев В.В.

Выполнила:

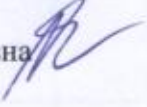
Студенка группы ЗФ-409-079-3-1

Бычок Кристина Александровна 

Научный руководитель:

старший преподаватель кафедры

АТ, ИТ и МОТД

Шварцкоп Ольга Николаевна 

Челябинск

2026

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b> .....	3
<b>ГЛАВА 1 ТЕОРЕТИКО-МЕТОДИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ ЭЛЕКТРОННОЙ РАБОЧЕЙ ТЕТРАДИ ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»</b> .....	6
1.1 Понятие, структура, виды и функции электронной рабочей тетради по дисциплине общепрофессионального цикла .....	6
1.2 Методические аспекты разработки электронной рабочей тетради по дисциплине «Основы алгоритмизации и программирования» как средство организации самостоятельной работы студентов профессиональной образовательной организации .....	10
1.3 Дидактические условия преподавания дисциплины ОП.04. «Основы алгоритмизации и программирования» в ГБПОУ «Златоустовский индустриальный колледж имени П.П. Аносова» .....	16
Выводы по главе 1 .....	18
<b>ГЛАВА 2 ПРОЕКТИРОВАНИЕ ЭЛЕКТРОННОЙ РАБОЧЕЙ ТЕТРАДИ КАК СРЕДСТВА ОБЕСПЕЧЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»</b> .....	20
2.1 Выбор среды и этапы разработки электронной рабочей тетради .....	20
2.2 Описание структуры и содержания электронной рабочей тетради по дисциплине «Основы алгоритмизации и программирования» как средство организации самостоятельной работы студентов профессиональной образовательной организации на примере раздела «Введение в программирование» .....	22
2.3 Анализ результатов применения электронной рабочей тетради по дисциплине «Основы алгоритмизации и программирования» .....	33
Выводы по Главе 2 .....	37
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b> .....	39

## ВВЕДЕНИЕ

Современные образовательные процессы с каждым годом требуют все больше постоянного совершенствования методов обучения, направленных на развитие самостоятельности студентов и повышение качества усвоения ими материала. Одной из перспективных форм организации учебного процесса является использование электронных рабочих тетрадей, которые позволяют интегрировать теоретический материал с практическими заданиями, обеспечивая интерактивное взаимодействие студента с учебным контентом как совместно с преподавателем, так и самостоятельно. Электронная рабочая тетрадь становится эффективным инструментом формирования профессиональных компетенций будущих специалистов, особенно в области информатики и программирования, где умение алгоритмически мыслить и грамотно проектировать программы имеет ключевое значение. В данной работе проведено исследование возможностей электронной рабочей тетради как средства развития основ алгоритмизации и программирования у студентов профессиональных образовательных организаций, направленного на формирование устойчивых навыков самостоятельного освоения дисциплины и подготовки к профессиональной деятельности.

*Актуальность.* Современное образование диктует необходимость усиления роли самостоятельной работы студентов в процессе обучения. Электронные рабочие тетради выступают эффективным средством реализации индивидуального подхода, обеспечивают доступность учебных материалов вне зависимости от места нахождения студента и способствуют развитию навыков самообразования и самоконтроля. Использование таких средств особенно актуально в условиях цифровой трансформации образовательных учреждений и перехода на дистанционное обучение.

Несмотря на значительное количество разработок в области электронного обучения, большинство созданных учебников и учебных пособий слабо интегрированы в практику самостоятельной работы

студентов. Между возможностями, предоставляемыми цифровыми технологиями, и реальной ситуацией, характеризующейся недостаточной мотивацией студентов к освоению материала самостоятельно, возникает *противоречие*.

*Проблема* заключается в отсутствии эффективных, но простых моделей и подходов к созданию электронных учебных материалов, направленных именно на организацию качественной самостоятельной работы учащихся.

*Цель исследования:* разработать эффективную понятную модель электронной рабочей тетради по дисциплине «Основы алгоритмизации и программирования», Раздел 1: «Введение в программирование», направленной на поддержку самостоятельной работы студентов профессиональной образовательной организации.

*Объект исследования:* процесс организации самостоятельной работы студентов профессиональных образовательных организаций по дисциплине «Основы алгоритмизации и программирования».

*Предмет исследования:* структура и содержание, как средство поддержки самостоятельной работы студентов, электронной рабочей тетради по дисциплине «Основы алгоритмизации и программирования».

*Задачи исследования:*

- изучить теорию и практику разработки электронных учебных пособий;
- определить принципы и этапы проектирования электронной рабочей тетради.
- создать структуру и наполнение содержательной части электронной рабочей тетради.
- провести экспериментальную проверку эффективности предлагаемой электронной рабочей тетради.
- сформулировать рекомендации по дальнейшему развитию и распространению опыта.

*Теоретико-методологическую основу исследования:* работа основана на концепциях личностно-ориентированного обучения, компетентностного подхода, концепции дистанционного обучения и опыте авторов, разрабатывающих учебно-методические комплексы нового поколения. Применяются также педагогические теории, направленные на активизацию познавательной активности студентов и создание условий для самостоятельного приобретения знаний.

*Методы исследования:* изучение и анализ методической, педагогической, специальной и учебной литературы; анализ федерального государственного образовательного стандарта СПО по направлению подготовки 09.02.04 «Информационные системы (по отраслям)»; изучение нормативных и методических документов; изучение и анализ учебно-программной и планирующей документации по дисциплине; изучение Интернет-ресурсов по проблеме исследования.

*База исследования:* исследования проводились на базе ГБПОУ «Златоустовский индустриальный колледж имени П.П. Аносова», г. Златоуст.

*Структура выпускной квалификационной работы.* Выпускная квалификационная работа включает оглавление, введение, основную часть (две главы), выводы по главам, заключение, список использованных источников.

# ГЛАВА 1 ТЕОРЕТИКО-МЕТОДИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ ЭЛЕКТРОННОЙ РАБОЧЕЙ ТЕТРАДИ ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

## 1.1 Понятие, структура, виды и функции электронной рабочей тетради по дисциплине общепрофессионального цикла

В современной педагогической теории и практике обучения большое внимание уделяется вопросам использования в системе среднего профессионального образования проблемных методов обучения и современных образовательных технологий. Однако действующим преподавателям известно, что в образовательных учреждениях, реализовывающих этот уровень образования (за редким исключением), учебный процесс осуществляется с использованием традиционных методов и форм обучения.

Необходимость повышения эффективности лекционных, практических и лабораторных занятий по общепрофессиональным дисциплинам требует от педагога применения новых форм и методов обучения на основе использования современных информационных технологий и методического сопровождения. В качестве такого методического сопровождения могут быть использованы рабочие тетради. Кроме того, именно через рабочие тетради педагог опосредованно управляет самостоятельной работой студентов.

В межгосударственном стандарте по информации, библиотечному и издательскому делу ГОСТ 7.60-2003 понятие «рабочая тетрадь» определяется следующим образом: «Рабочая тетрадь – это учебное пособие, имеющее особый дидактический аппарат, способствующий самостоятельной работе учащегося над освоением учебного процесса» [6]. Однако педагоги неоднозначно понимают назначение рабочих тетрадей в образовательном процессе, что влечет за собой существование различных трактовок этого понятия.

Например, по Р.В. Фесичу, «рабочая тетрадь – учебное издание (средство обучения), на основании работы с которым у обучающихся происходит осознание нового учебного материала и формирование практических умений, которое также служит целям организации самостоятельной работы обучающихся и контроля их учебных достижений» [7, с. 13].

В работе Л.П. Алексеевой читаем: «Рабочая тетрадь представляет собой средство обучения, являющееся одновременно как средством преподавания, так и средством учения. Они быстрее других откликаются на потребности образовательного процесса и одновременно формируют эти потребности. Введение рабочих тетрадей в арсенал дидактических средств предоставляет преподавателю реальную возможность оптимизировать учебный процесс» [8].

Рабочая тетрадь – учебно-практическое издание, предназначенное для работы обучающихся, как в аудитории, так и для самостоятельной подготовки, в котором соединяется изложение основных положений курса с выработкой общих и профессиональных компетенций у обучающегося, формирования практических умений и навыков. Изложение материала в рабочей тетради чередуется с пробелами, которые заполняет обучающийся по ходу ведения лекции, практического (лабораторного) занятия, выполнения курсовой работы (проектирования).

Цель рабочей тетради – обеспечить пооперационное формирование мыслительных процессов, способствовать повышению эффективности обучения студентов и уровня их творческого развития. Электронная рабочая тетрадь – это специализированный цифровой ресурс (средство обучения), направленный на поддержку образовательного процесса, который включает теорию, тесты, практические задания, интерактивные упражнения, и т.п.

Структуру электронной рабочей тетради можно представить таким образом:

- теоретический раздел темы (основные понятия, определения изучаемой дисциплины), представленный понятным для студентов образом;
- практический раздел (задания, упражнения, лабораторные работы, которые позволяют закрепить полученные знания);
- контрольный раздел (упражнения, тестовые задания для самоконтроля студентов);
- дополнительный раздел (дополнительная литература в виде списка или ссылок, методические рекомендации, мультимедийный материал или ссылки на него).

В зависимости от уровня интеграции с образовательным процессом, различают следующие виды электронных рабочих тетрадей:

- простые (тексты, таблицы, схемы, рисунки);
- интерактивные (задания с возможностью проверки, гиперссылки, анимация и т.д.);
- адаптивные (под уровень знаний и потребность конкретного студента);
- смешанные (объединяет все выше перечисленные виды).

Основными функциями электронной рабочей тетради являются:

- обеспечение доступности учебных материалов в любое удобное время и месте, что особенно удобно при дистанционном образовательном процессе;
- повышение мотивации учащихся путем внедрения определенных элементов (например, задания, оценивающиеся по бальной шкале на определенную отметку, или бонус/поощрение при его выполнении);
- формирование у студентов навыков критического мышления и самообразования.
- организация индивидуального процесса обучения каждого студента (под определенный уровень знаний, или потребность);
- автоматизация контроля знаний и оценки результатов (например, автоматически подсчет баллов при прохождении тестов и т.п., что существенно ускоряет процесс обработки контроля знаний педагогом);

- создание условий для активного участия преподавателя в процессе обучения (вариант прямой связи с преподавателем видео/аудио/чат и т.п.).

В таблице приведено соответствие уровня самостоятельной работы студента и возможностей ЭРТ в рамках качественного освоения рассматриваемого вида работ. Задания 4-го уровня «Исследовательские работы» не включены в ЭРТ. Эти задания, целью которых является формирование навыков исследовательской, творческой деятельности, предполагают выполнение курсовых и научно-исследовательских работ. Такие задания предназначены только для тех студентов, которые успешно справились с заданиями первых трех уровней, поэтапное выполнение которых подготовило их к работе исследовательского характера.

Таблица 1 - Соответствие уровней и целей самостоятельной работы студентов типам заданий ЭРТ

Уровень СРС	Цель СРС	Тип заданий ЭРТ	Деятельность студента
1 уровень. Воспроизводящие самостоятельные работы по образцу	Закрепление знаний, формирование умений, навыков	Задачи для решения на аудиторном занятии, для которых указана последовательность действий	Обучается самостоятельно работать с теоретическим материалом, находить ответ на поставленный вопрос, решать примеры и задачи
2 уровень. Реконструктивно-вариативные работы	Формирование умений и навыков преобразования, реконструирования, обобщения ранее приобретенных знаний и умений, установления внутрипредметных связей	Вариативные домашние задания, предназначенные для самостоятельного решения, для которых указана схема действий	Воспроизводит не только отдельные части знаний, но и их целостную структуру. Обучается переходить от идеи и общего метода к способу решения конкретной задачи
3 уровень. Эвристические работы	Формирование умений и навыков осуществления поисковой деятельности и установления межпредметных связей	Индивидуальное задание (кейс), содержащее проблемную ситуацию, для разрешения которой необходимо составить план действий и собрать статистическую информацию	Разрешает проблемную ситуацию, приобретает опыт поисковой деятельности, овладевает элементами творчества

На сегодняшний день внедрение рабочей тетради в образовательный процесс помогает решить важные задачи: развития мышления у студентов, прочное усвоение теоретических знаний, приобретение практических умений и навыков решения не только типовых, но и развивающих, творческих заданий, контроль за ходом обучения студентов конкретной учебной дисциплине (профессиональному модулю), формирование у студентов умений и навыков самоконтроля.

Таким образом, электронная рабочая тетрадь выступает важным элементом современного образования, обеспечивающим комплексный подход к обучению и повышающим эффективность учебно-воспитательного процесса.

1.2 Методические аспекты разработки электронной рабочей тетради по дисциплине «Основы алгоритмизации и программирования» как средство организации самостоятельной работы студентов профессиональной образовательной организации

С реализацией ФГОС среднего профессионального образования значительно увеличилась доля самостоятельной работы студентов, возросла необходимость рассмотрения самостоятельной работы как важнейшей составляющей образовательного процесса: раскрытие закономерностей самостоятельной работы студента как системы; разработка новых дидактических средств, методов обучения и образовательных технологий, форм взаимодействия преподавателя и студентов. Самостоятельная форма работы позволяет преобразовать студента из пассивного потребителя знаний в активного их творца, который способен сформулировать проблему, разработать и проанализировать пути ее решения, выбрать оптимальный путь и, получив результат, доказать его правильность.

О важности самостоятельной работы обучающихся писали многие педагоги прошлого (А. Дистервег, Я.А. Коменский, К.Д. Ушинский). По

мнению К.Д. Ушинского, именно самостоятельная работа обучающихся является важнейшим условием овладения знаниями и является «единственно прочным основанием всякого плодovitого учения» [3, с. 226].

Определению сущности и организации самостоятельной работы обучающихся посвящены работы современных педагогов, психологов, методистов. Видные отечественные ученые (Ю.К. Бабанский, Л.В. Бордонская, В.И. Загвязинский, И.А. Зимняя, П.И. Пидкасистый, А.П. Тряпицына и др.) рассматривают самостоятельную работу как:

- учебную деятельность;
- средство обучения;
- средство формирования познавательных способностей;
- способ организации учебного процесса, который направлен на формирование принципов самодеятельности, саморазвития и самоорганизации;
- форму организации самостоятельной познавательной деятельности;
- условие формирования самоорганизации и самодисциплины.

С целью организации самостоятельной работы студентов используют как традиционные дидактические средства (учебники, учебные пособия, справочники, методички, задачки и практикумы), так и современные (портфолио педагога, технологические карты, рабочие тетради).

Результативность использования разных видов рабочих тетрадей зависит от уровня сформированности навыков самостоятельной работы студентов и степени их самоорганизации, следовательно, отличается у студентов разных курсов.

Создание эффективной электронной рабочей тетради (ЭРТ) по дисциплине «Основы алгоритмизации и программирования» требует учета множества факторов, среди которых особое внимание уделяется дидактическим принципам, педагогической целесообразности и эргономичности интерфейса.

Методологические подходы к разработке ЭРТ предполагают поэтапное внедрение инновационных технологий, ориентированных на усиление самостоятельности и активности обучающихся. Основной задачей электронного ресурса является создание комфортной среды для восприятия и закрепления базовых концепций алгоритмизации посредством интерактивных практикумов, тестов и визуализаций.

Ключевые аспекты разработки ЭРТ включают проектирование структуры и содержания пособия, подбор соответствующего программного инструментария, определение критериев оценивания достижений студентов. Важно учитывать особенности возрастной группы, специфику самого предметного направления. Грамотно составленная методика позволит повысить мотивацию студентов, способствовать глубокому освоению материала и формированию необходимых профессиональных компетенций.

Особенное внимание уделено вопросам адаптивности ЭРТ: включение уровней сложности заданий, вариативность практических ситуаций, возможность индивидуального выбора путей решения задач способствуют лучшему восприятию материала и стимулируют познавательную активность обучающихся. Таким образом, грамотная разработка ЭРТ способствует повышению эффективности процесса обучения, улучшает качество усвоения студентами основных принципов алгоритмизации и развивает навыки самостоятельного овладения профессиональными дисциплинами.

Конструирование электронной рабочей тетради состоит из нескольких этапов. К ним можно отнести подготовительный, основной и завершающий этапы (таблица 2).

Таблица 2 –Этапы конструирования электронной рабочей тетради

Название этапа	Содержание этапа
Подготовительный	<ul style="list-style-type: none"> <li>- Определение курса и возрастной группы обучающихся;</li> <li>- анализ учебно-тематического планирования по выбранному курсу;</li> <li>- определение принципов структурирования ЭРТ;</li> <li>- создание структуры ЭРТ в соответствии с выбранным курсом.</li> </ul>
Основной	<ul style="list-style-type: none"> <li>- Определение типа, вида и цели урока;</li> <li>-определение цели использования ЭРТ на уроке;</li> <li>-определение УУД, формируемых на уроке при работе с ЭРТ;</li> <li>- определение этапов урока, на которых будет использована ЭРТ;</li> <li>-отбор содержания по указанной теме курса;</li> <li>- подбор учебных заданий в соответствии с УУД, формируемыми на определённом этапе урока;</li> </ul>
Завершающий	<ul style="list-style-type: none"> <li>- Занесение дидактического материала в технологическую карту урока</li> </ul>

Первое, что необходимо сделать при создании электронной рабочей тетради - это выбрать курс дисциплины «Основы алгоритмизации и программирования», темы и возрастную группу обучающихся, на которых будет рассчитано содержание рабочей тетради.

В нашем случае это темы по разделу 1 «Введение в программирование», преподаваемые в колледже на 1 курсе. Следовательно, возрастная группа обучающихся соответствует 15-16 годам.

Затем проводим анализ учебно-тематического планирования заявленного курса.

На данном этапе необходимо определить темы, которые будут отражены в оглавлении электронной рабочей тетради, и количество часов, отводимых на их изучение (таблица 2).

Таблица 3 – Планирование использования электронной рабочей тетради

Глава I «Основы алгоритмизации»			
Тема урока	Количество часов	Тип задания	Формат электронной доски
Тема 1.1. Языки программирования	6	Задания в виде тестов и вопросов для самоконтроля	Рабочая область с теорией, ссылками, медиафайлами
Тема 1.2. Типы данных	4		

В соответствии с данными темами пройдёт отбор типовых заданий, дополнительной учебной литературы и источников, которые в дальнейшем будут размещены на рабочей области, а также структура ЭРТ.

Структура электронной рабочей тетради выглядит следующим образом:

1. Введение;
2. Тема 1.1 Языки программирования:  
Развитие языков программирования.

Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.

Жизненный цикл программы. Программа. Программный продукт и его характеристики.

Основные этапы решения задач на компьютере.

3. Тема 2.2 Типы данных:

- Типы данных. Простые типы данных. Производные типы данных.

Структурированные типы данных.

Процесс отбора содержания, а именно основной и дополнительной информации, расположенной в рабочей тетради, опирается на важные факторы. Для их определения необходимо приступить к очередному шагу

методики: определение принципов конструирования электронной рабочей тетради.

Определение принципов конструирования электронной рабочей тетради играет главную роль в отборе основного и дополнительного содержания.

Ключевые этапы:

1. Определение целей и задач. Перед началом разработки важно чётко сформулировать цели и задачи ЭРТ. Они определяют структуру, содержание и методы подачи материала. Например, целью может стать углубление понимания студентом ключевых аспектов алгоритмизации и программирования, а также приобретение навыков самостоятельной работы.

2. Принцип научности и систематичности. Принцип научности предполагает предоставление проверенной и актуальной информации, соответствующей современным научным представлениям и требованиям профессионального стандарта. Систематичность подразумевает структурированную подачу материала, начиная с простейших концепций и постепенно переходя к более сложным.

3. Принцип наглядности и доступности. Наглядность обеспечивается использованием графических схем, таблиц, диаграмм и примеров, облегчающих восприятие сложных идей. Доступность достигается ясностью изложения и отсутствием избыточной терминологии.

4. Принцип дифференциации и индивидуализации. Дифференциация позволяет предложить студентам разные уровни сложности заданий, соответствующие их уровню подготовки. Индивидуализация направлена на удовлетворение потребностей отдельных студентов путём предоставления выборочных вариантов упражнений и консультационных ресурсов.

5. Принцип обратной связи и самоконтроля. Обратная связь необходима для своевременного выявления ошибок и недопониманий. Самоконтроль помогает студентам самостоятельно оценивать своё продвижение и выявлять слабые места.

6. Практическое применение полученных знаний. Практикоориентированность обеспечивает тесную взаимосвязь теории и практики, позволяя студентам применять полученные знания непосредственно в реальных ситуациях.

Эти принципы формируют основу успешного конструирования электронной рабочей тетради.

1.3 Дидактические условия преподавания дисциплины ОП.04. «Основы алгоритмизации и программирования» в ГБПОУ «Златоустовский индустриальный колледж имени П.П. Аносова»

Место дисциплины в структуре основной профессиональной образовательной программы.

Учебная дисциплина «Основы алгоритмизации и программирования» является обязательной частью Общепрофессионального учебного цикла основной образовательной программы в соответствии с ФГОС СПО и ПООП по специальности 09.02.07 Информационные системы и программирование. Особое значение дисциплина имеет при формировании и развитии ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-ПК 1.5, ПК 2.4, 2.5.

Цель и планируемые результаты освоения дисциплины.

В результате освоения дисциплины «Основы алгоритмизации и программирования» обучающийся должен уметь: разрабатывать алгоритмы для конкретных задач; использовать программы для графического отображения алгоритмов; определять сложность работы алгоритмов; работать в среде программирования; реализовывать построенные алгоритмы в виде программ на конкретном языке программирования; оформлять код программы в соответствии со стандартом кодирования; выполнять проверку, отладку кода программы.

В результате освоения дисциплины «Основы алгоритмизации и программирования» обучающийся должен знать: понятие алгоритмизации,

свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; эволюцию языков программирования, их классификацию, понятие системы программирования, основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти, подпрограммы, составление библиотек под программ 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

Ниже, представлена реализуемая часть в ЭРТ Тематического плана и содержание учебной дисциплины «ОП.04 Основы алгоритмизации и программирования».

Таблица 4 – Тематический план и содержание учебной дисциплины «ОП.04 Основы алгоритмизации и программирования» Раздел 1

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающегося, курсовая работа (проект)	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
<b>Раздел 1.</b>	<b>Введение в программирование</b>	<b>10</b>	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5 ЛР 13,14,15,16
<b>Тема 1.1. Языки программирования</b>	<b>Содержание учебного материала</b>	<b>6</b>	
	1. Развитие языков программирования.		
	2. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.		
	3. Жизненный цикл программы. Программа. Программный продукт и его характеристики.		
	4. Основные этапы решения задач на компьютере.		
	<b>В том числе практических занятий и лабораторных работ</b>	<b>2</b>	
	Знакомство со средой программирования. Составление программ линейной структуры		

Продолжение таблицы 4

Тема 1.2. Типы данных	Содержание учебного материала	4
	1. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.	
	<b>В том числе практических занятий и лабораторных работ</b> Составление программ разветвляющейся структуры. Составление программ циклической структуры	2

Выводы по главе 1

В первой главе были выявлены основные теоретико-методические аспекты разработки электронной рабочей тетради по дисциплине «Основы алгоритмизации и программирования», раздел 1: «Введение в программирование». Рассмотрены ключевые моменты, касающиеся назначения, функций и видов электронной рабочей тетради, их влияния на образовательный процесс.

Вкратце подытожим основные положения.

1. Современные требования к образованию характеризуется постоянным совершенствованием методов обучения, направленными на повышение самостоятельности студентов и улучшение качества усвоения материала. Электронные рабочие тетради играют важную роль в формировании учебного процесса.

2. Функциональные возможности электронной рабочей тетради. Важнейшими характеристиками электронной рабочей тетради являются её способности поддерживать учебный процесс, обеспечивать доступ к материалам независимо от местоположения, повышать мотивацию студентов и формировать навыки критического мышления и самообучения. Это инструмент, позволяющий организовать индивидуальный подход к каждому учащемуся.

3. Структура и содержание электронной рабочей тетради. Оптимальная структура включает четыре раздела: теоретический, практический, контрольный и дополнительный. Каждый раздел выполняет

свою уникальную функцию, помогающую эффективно усваивать знания и развивать необходимые компетенции.

4. Виды электронных рабочих тетрадей. Выделяют простые, интерактивные, адаптивные и смешанные формы, каждая из которых обладает своими преимуществами и особенностями. Выбор подходящего вида зависит от конкретных нужд учебного заведения и цели применения педагогом.

5. Концепция разработки электронной рабочей тетради. Процесс разработки начинается с анализа требований Федерального государственного образовательного стандарта, изучения особенностей выбранной дисциплины и определения возрастных характеристик студентов. Затем осуществляется тщательная подготовка содержательных компонентов, включая отбор наиболее значимых и полезных материалов.

6. Методы и критерии оценивания. Эффективность разработанной модели определяется качеством выполненных заданий, уровнем самостоятельности студентов и степенью сформированности профессиональных компетенций. Оценочные показатели помогают отслеживать прогресс и вносить изменения в методику обучения.

Разработанная модель электронной рабочей тетради представляет собой инструмент, помогающий образовательному процессу. Свойства электронной рабочей тетради делают её незаменимым помощником в современном учебном пространстве, открывая новые возможности эффективного взаимодействия преподавателей и студентов.

## ГЛАВА 2 ПРОЕКТИРОВАНИЕ ЭЛЕКТРОННОЙ РАБОЧЕЙ ТЕТРАДИ КАК СРЕДСТВА ОБЕСПЕЧЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

### 2.1 Выбор среды и этапы разработки электронной рабочей тетради

Для разработки электронной рабочей тетради была выбрана CoreApp.AI — инновационная среда разработки программного обеспечения, специально созданная для учебных целей. Эта облачная IDE позволяет создавать, редактировать и запускать код прямо в браузере без установки дополнительного ПО. Ее преимущества включают интуитивно понятный интерфейс, встроенную систему автоматического тестирования, поддержку популярных языков программирования и совместимость с различными устройствами.

*При выборе среды разработки были учтены следующие факторы:*

1. Среда разработана специально для новичков, позволяя легко освоить процесс написания кода даже людям без опыта программирования. Интерфейс прост и интуитивен, что помогает избежать путаницы и быстро начать работу.

#### 2. Онлайн-доступ

Пользователи могут сразу приступить к созданию и тестированию приложений, не устанавливая дополнительное программное обеспечение. Это особенно удобно для образовательных учреждений, поскольку снимает необходимость обновления ПО на всех компьютерах.

#### 3. Автоматическое тестирование

Интеграция автоматической системы тестирования значительно упрощает контроль качества кода студентами. Они могут мгновенно проверить правильность своего решения, что ускоряет обучение и повышает мотивацию.

#### 4. Поддержка множества языков программирования

CoreApp.AI поддерживает широкий спектр языков программирования, включая Python, JavaScript, Ruby и многие другие. Это позволяет выбирать подходящий язык для конкретных задач курса и облегчает переход между языками при обучении.

#### 5. Интерактивные инструменты

Среда оснащена мощными средствами визуализации процессов исполнения программы, такими как пошаговая трассировка, что существенно улучшает восприятие сложных концепций программирования.

#### 6. Совместимость устройств

CoreApp.AI доступна практически на любых устройствах с доступом в Интернет, будь то компьютер, планшет или смартфон. Это создает гибкость в процессе обучения и снижает зависимость от конкретного оборудования.

#### *Особенности среды разработки CoreApp.AI*

CoreApp.AI — инновационная среда разработки программного обеспечения, специально созданная для учебных целей. Эта облачная IDE позволяет создавать, редактировать и запускать код прямо в браузере без установки дополнительного ПО. Ее преимущества включают интуитивно понятный интерфейс, встроенную систему автоматического тестирования, поддержку популярных языков программирования и совместимость с различными устройствами.

Однако использование CoreApp.AI имеет и некоторые ограничения:

Преимущества:

Удобна для начинающих разработчиков благодаря простоте интерфейса.

Не требует специальной подготовки оборудования.

Возможность мгновенной проверки правильности написания кода.

Наличие интегрированной системы подсказок и исправлений ошибок.

Недостатки:

Ограниченная функциональность по сравнению с профессиональными средами разработки.

Отсутствие возможности локального сохранения проектов.

Возможны проблемы с производительностью при работе с большими проектами.

Таким образом, электронная рабочая тетрадь обеспечивает эффективную организацию учебного процесса и способствует формированию необходимых компетенций у обучающихся.

2.2 Описание структуры и содержания электронной рабочей тетради по дисциплине «Основы алгоритмизации и программирования» как средство организации самостоятельной работы студентов профессиональной образовательной организации на примере раздела «Введение в программирование»

Рабочая тетрадь состоит из одного раздела и двух тем, которые имеют свои подтемы для структурированного изучения материала и самостоятельной работы студентов. В тетради представлено последовательное освоение тем, выполнение задания и проверка знаний.

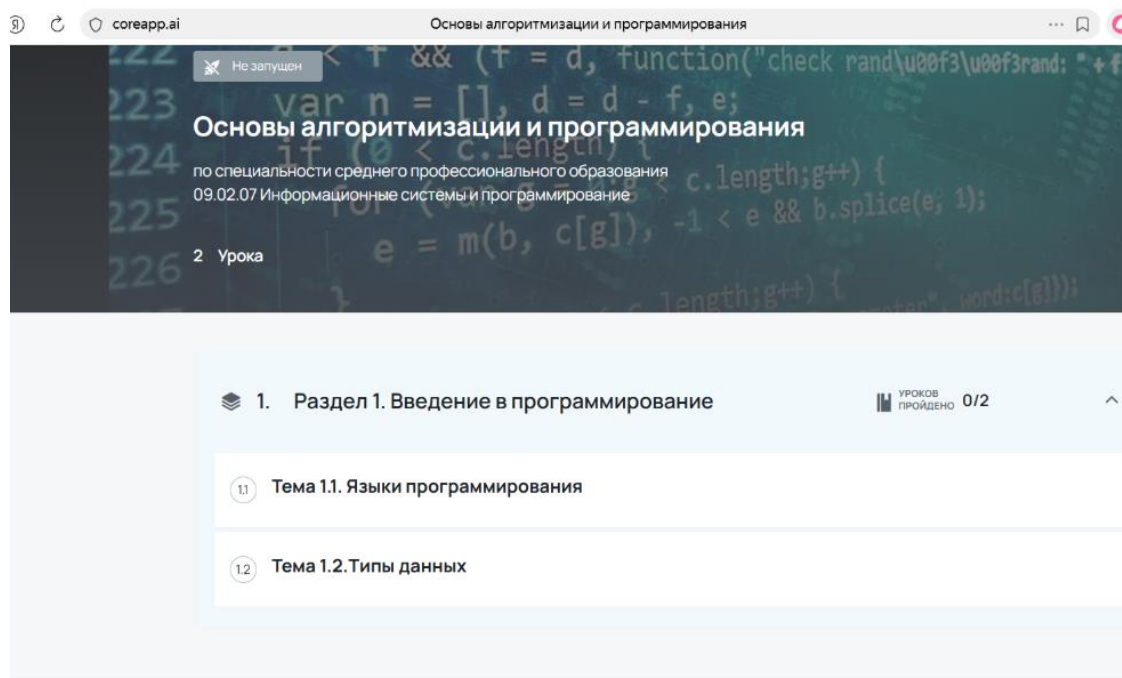


Рисунок 1 - Главная страница ЭРТ

Раздел 1: «Введение в программирование» охватывает базовые понятия дисциплины и включает теоретический материал, сопровождающийся примерами, заданиями и упражнениями для закрепления изученного материала. Тетрадь состоит из последовательно расположенных модулей, охватывающих основные темы раздела:

1. Тема 1.1 «Языки программирования»

Каждый модуль сопровождается подробными пояснениями, наглядными иллюстрациями.

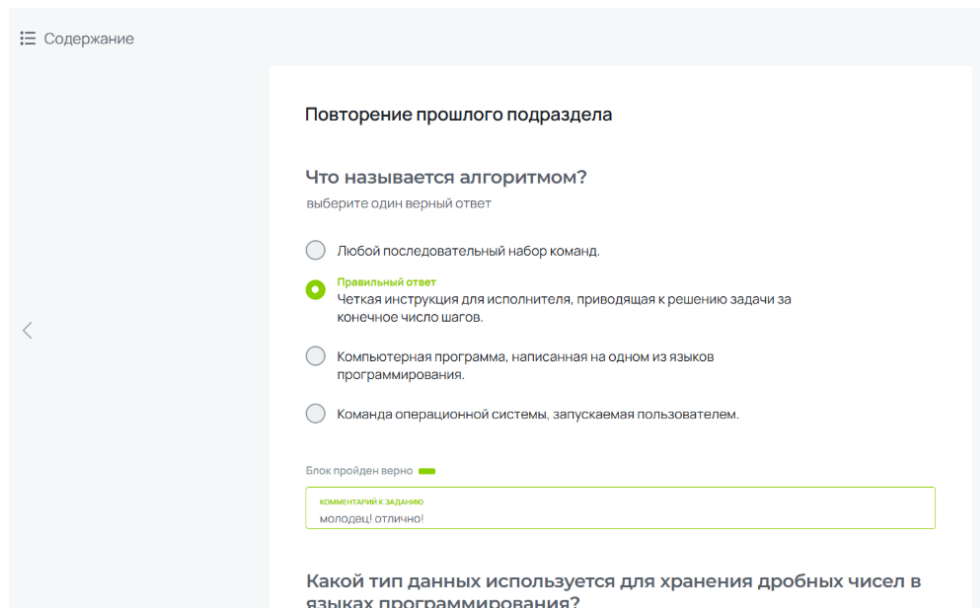


Рисунок 2 - Повторение прошлой темы

На рисунке 2 представлен один из тестовых вопросов на повторение пройденного материала.

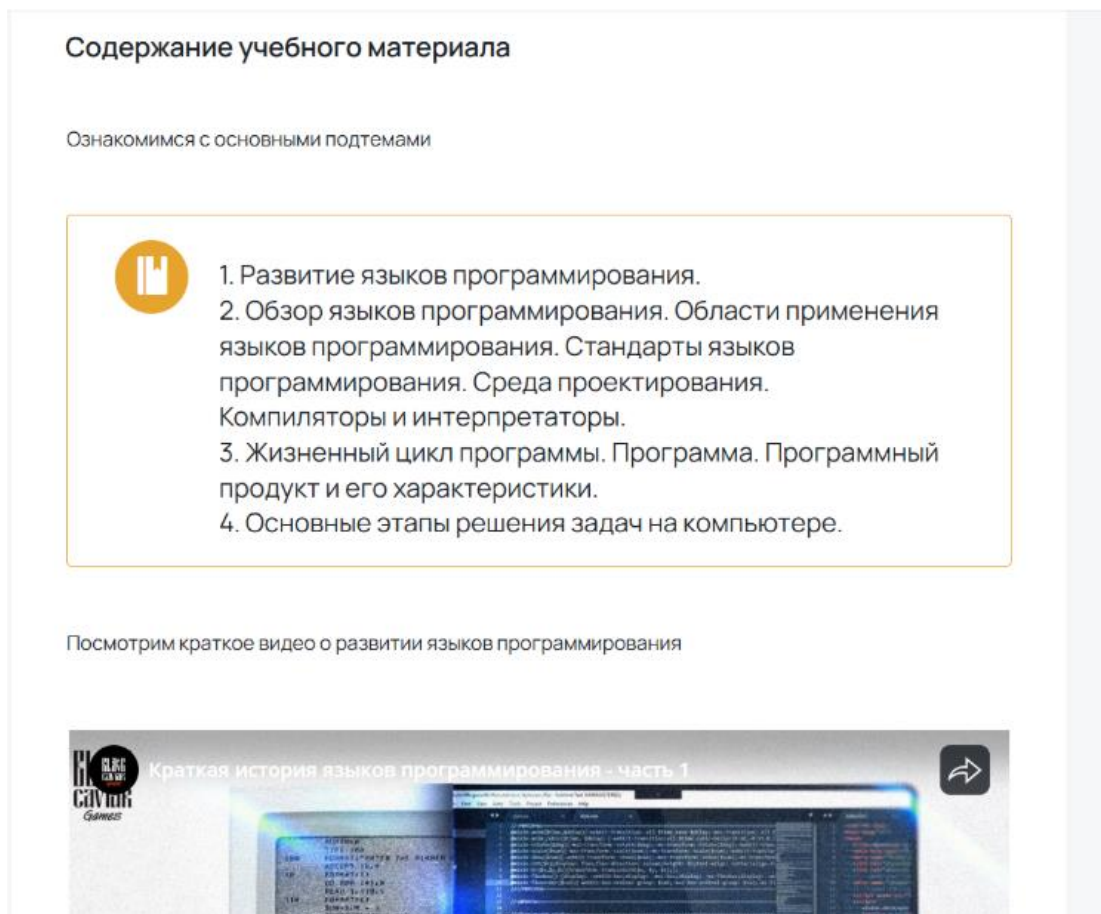


Рисунок 3-Содержание учебного материала, видеоурок

На рисунке 3 представлены содержание учебного материала, краткий видеоурок.

Далее расположены страницы с лекциями по содержанию учебного материала на указанную тему.

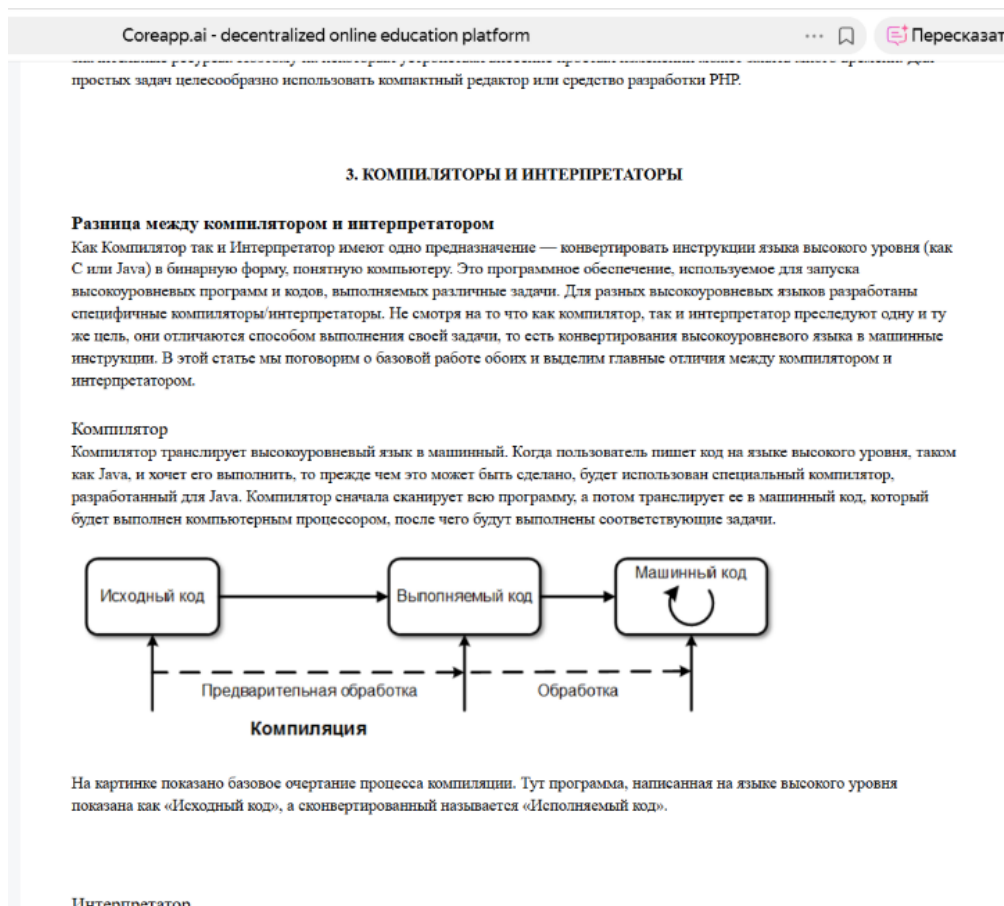


Рисунок 4 - Страница с теоретическим материалом

Переход на страницы и непосредственно на содержание урока осуществляется путем нажатия на кнопки: «Предыдущая страница», «Содержание», «Следующая страница».

Coreapp.ai - decentralized online education platform

Очертание процесса интерпретации на картинке выше показывает, что сначала исходный код конвертируется в промежуточную форму, а затем выполняется интерпретатором.

Ниже перечислены главные отличия между компилятором и интерпретатором:

- Интерпретатор берет одну инструкцию, транслирует и выполняет ее, а затем берет следующую инструкцию. Компилятор же транслирует всю программу сразу, а потом выполняет ее.
- Компилятор генерирует отчет об ошибках после трансляции всего, а то время как интерпретатор прекратит трансляцию после первой найденной ошибки.
- Компилятор по сравнению с интерпретатором требует больше времени для анализа и обработки языка высокого уровня.
- Помимо времени на обработку и анализ, общее время выполнения кода компилятора быстрее в сравнении с интерпретатором.

< Предыдущая страница    Содержание    Следующая страница >

Рисунок 5-Пример перехода на страницы

Coreapp.ai - decentralized online education platform

### Промежуточный контроль по подтеме

Ответьте на следующие вопросы

#### Что представляет собой язык программирования?

выбери один правильный ответ

- Правильный ответ**  
Искусственный язык для записи компьютерных программ.
- Естественный разговорный язык, понятный компьютерам.
- Физический механизм обработки данных в компьютере.
- Инструмент автоматического перевода текста на естественный язык.

Блок пройден верно ■

КОММЕНТАРИЙ К ЗАДАНИЮ  
Молодец! Ты был внимателен

#### Как называются языки программирования, близкие к командам процессора и зависящие от архитектуры компьютера?

Рисунок 6 - Контроль знаний по подтеме

После каждой подтемы студенту предлагается пройти промежуточный контроль.

В конце основной темы представлены вопросы для самоконтроля в виде тестов.

Вопросы для самоконтроля

**Какой этап жизненного цикла программного обеспечения предусматривает описание требований, постановку задач и утверждение технического задания?**  
выбери один верный ответ

**Неправильный ответ**  
Проектирование

Написание кода

Анализ требований

Тестирование и отладка

Блок в процессе прохождения ■

**Что означает аббревиатура CI/CD, применимая в практике DevOps?**  
выберите один верный ответ

Конфигурация инфраструктуры / Контроль доставки

Рисунок 7 - Пример вопроса для самоконтроля

Далее студент подводит итоги - рефлексия. Студент отвечает на вопросы преподавателя, и прикрепляет файл для анализа им.

## Рефлексия



1. Что нового и важного узнали на сегодняшнем занятии? Какие знания были особенно полезны или вызвали наибольший интерес?
2. Какие трудности возникли в процессе освоения материала? Что именно было сложно понять или усвоить, и почему?
3. Как новый материал повлиял на мировоззрение или отношение к изучаемой дисциплине?
4. Как полученные знания можно применить в реальной жизни? Приведите конкретные примеры ситуаций, где новые знания окажутся полезными.
5. Какой аспект занятий мне хотелось бы изменить или улучшить в будущем? Возможно, добавить больше практических заданий, групповых обсуждений или наглядных примеров?

### Ответь на вопросы педагога

Введите ответ



Кристина Бычок

Прикрепить файл

## Рисунок 8– Рефлексия

Далее, студенту предлагается задание для самостоятельного выполнения. Студент прикрепляет файл с выполненным заданием. Также, может оставить аудио-ответ или комментарий по предложенному заданию. Для каждого задания свой блок ответа.

**1** Задание для блока №1. Исследование популярных языков программирования  
Цель: познакомиться с основными языками программирования и их особенностями.  
Инструкция:  
1. Составьте список наиболее востребованных языков программирования на сегодняшний день (например, Python, JavaScript, C++, Java).  
2. Для каждого языка укажите:  
- Краткую историю появления и развития.  
- Основные области применения.  
- Примеры известных проектов или компаний, использующих этот язык.  
3. Подготовьте сравнительную таблицу с ключевыми характеристиками выбранных вами языков программирования.

Задание для блока №2. Практическое задание: написание простой программы  
Цель: попробовать самостоятельно написать простую программу на одном из языков программирования.  
Инструкция:  
Выберите один из предложенных языков программирования (Python, JavaScript, C++). Напишите небольшую программу, решающую одну из предложенных ниже задач:  
1. Программа должна запрашивать у пользователя ввод двух чисел и выводить их сумму.  
2. Программа должна вычислять факториал введенного числа.  
3. Программа должна проверять, является ли введенное число простым числом.  
Предоставьте код вашей программы вместе с пояснениями, как работает каждая строка вашего решения.

#### Выполненные задания, БЛОК №1

Прикрепите файл с выполненным заданием

Щелкните, чтобы написать ответ

Прикрепить файл

Записать аудио-ответ

#### Выполненные задания, БЛОК №2

Прикрепите файл с выполненным заданием

Рисунок 9 - Задание для самостоятельного выполнения

Аналогичная схема работы для студента представлена в следующей теме Раздела 1.

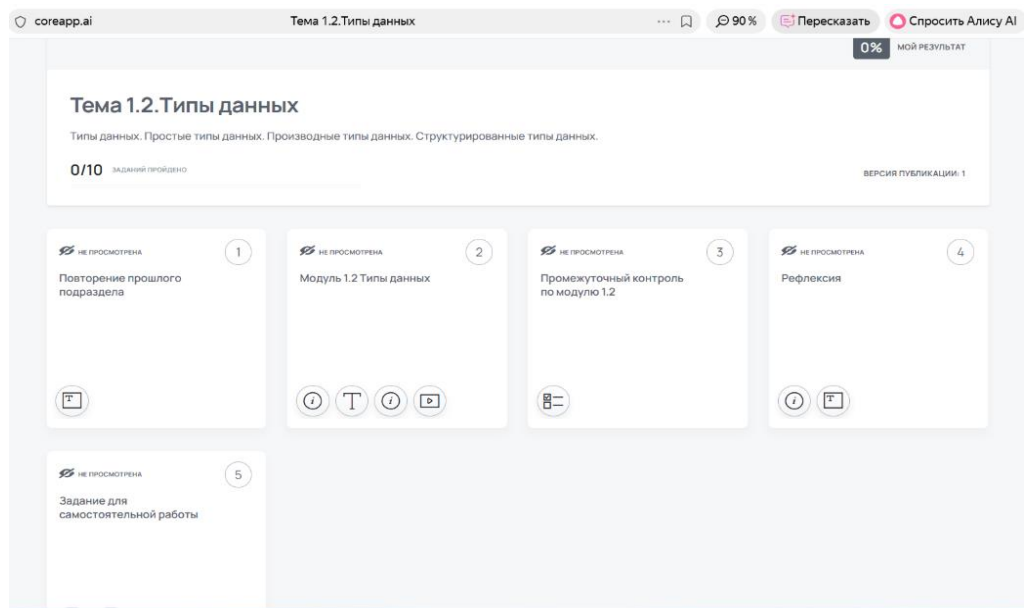


Рисунок 10 - Главная страница второго урока

### Повторение прошлого подраздела

**Перечислите ключевые стадии жизненного цикла программы и дайте характеристику каждой из них. Расскажите, каким образом тестирование влияет на качество конечного продукта.**

**Развитие языков программирования: Назовите основные тенденции и направления, определяющие развитие современных языков программирования. Чем обусловлен рост популярности высокоуровневых языков?**

Рисунок 11 - Повторение прошлого подраздела

## Модуль 1.2 Типы данных



Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.

В данной лекции мы будем рассматривать основные понятия, связанные с типами данных в программировании. Мы начнем с общего представления о типах данных и затем перейдем к их классификации на простые и производные типы. Давайте начнем!

Что такое типы данных?

Типы данных - это средства, используемые в программировании для определения характеристик данных, которые могут быть использованы программой. Они определяют набор операций, которые могут быть применены к данным, а также ограничения этих операций.

Простые типы данных

Простые типы данных - это базовые типы данных, которые предоставляются программным языком. Они обычно представляют основные типы данных, такие как целочисленные значения, числа с плавающей точкой, символы и логические значения. Давайте подробнее рассмотрим каждый из них.

Целочисленные значения

Целочисленные значения - это тип данных, который представляет целые числа, как положительные, так и отрицательные. Этот тип данных обычно представлен в виде чисел в десятичной системе счисления, но он также может быть представлен в других системах счисления, таких как двоичная или шестнадцатеричная. Примеры целочисленных значений: 0, 1, -10, 100.

Числа с плавающей точкой

Числа с плавающей точкой - это тип данных, который представляет дробные числа. Он обычно используется для представления чисел с плавающей точкой или научных значений. Примеры чисел с плавающей точкой: 3.14, -0.01, 1.0e-6.

Символы

## Рисунок 12 - Теоретическая часть

Coreapp.ai - decentralized online education platform

содержать переменные для хранения улицы, номера дома, города и почтового индекса.

Классы

Классы - это тип данных, который позволяет объединять переменные и функции в одну единицу, называемую объектом. Они являются основой объектно-ориентированного программирования и позволяют создавать сложные структуры и функции, инкапсулируя их в объекты.

▶ Посмотрим видеоурок

Основы программирования с нуля. Типы данных и переменные

### Типы данных

Тип данных – способ хранения и представления данных на компьютере.

```
graph TD; Root[Данные] --> Числовые[Числовые]; Root --> Символьные[Символьные (a, b, д, «BytesMagic»)]; Root --> Логические[Логические (boolean), true, false]; Числовые --> Целые[Целые числа (1, 2, -6, 100)]; Числовые --> Плавающие[Числа с плавающей точкой (1.2, 0.5, 65.7)];
```

Смотреть на YOUTUBE

Рисунок 13 - Представленный видеоурок по подтеме

## Промежуточный контроль по модулю 1.2

### Что такое типы данных в программировании?

выберите один верный ответ

- Правильный ответ**  
Средства для определения характеристик данных и ограничений операций над ними.
- Механизмы шифрования данных.
- Регулярные выражения для поиска совпадений.
- Библиотеки готовых функций.

Блок пройден верно ■

### Выберите простой тип данных:

выберите один верный ответ

- Целочисленные значения
- Структуры
- Массивы
- Классы

Блок не пройден

Рисунок 14 - Аналогичный промежуточный контроль по подтеме

## Рефлексия



1. Что нового и важного узнали на сегодняшнем занятии? Какие знания были особенно полезны или вызвали наибольший интерес?
2. Какие трудности возникли в процессе освоения материала? Что именно было сложно понять или усвоить, и почему?
3. Как новый материал повлиял на мировоззрение или отношение к изучаемой дисциплине?
4. Как полученные знания можно применить в реальной жизни? Приведите конкретные примеры ситуаций, где новые знания окажутся полезными.
5. Какой аспект занятий мне хотелось бы изменить или улучшить в будущем? Возможно, добавить больше практических заданий, групповых обсуждений или наглядных примеров?

### Ответь на вопросы педагога

Щелкните, чтобы написать ответ

Прикрепить файл

Записать аудио-ответ

## Рисунок 15— Рефлексия

### Задание для самостоятельной работы

**1** Задание для блока ответа №1

Типы данных  
Определите тип каждого выражения ниже и укажите размер памяти, занимаемый данным типом данных в байтах (предположим, что мы работаем на платформе x86\_64):

1. `char ch = 'A';`
2. `unsigned int num = 42;`
3. `float pi = 3.14f;`
4. `long long bigNum = 123456789L;`
5. `bool flag = true;`

Задание для блока ответа №2

Производные и структурированные типы данных  
Создайте структуру `Person`, содержащую поля: фамилия (`lastName`), возраст (`age`) и рост (`height`). Затем объявите массив из трёх объектов типа `Person`. Заполните каждый объект соответствующими значениями и выведите их на экран в следующем формате:  
Фамилия: Иванов  
Возраст: 30  
Рост: 180 см  
Для решения используйте любые доступные вам языки программирования (например, C++, Python, Java).

**Выполненные задания для блока №1**  
прикрепите файл с заданием №1

Щелкните, чтобы написать ответ

Прикрепить файл      Записать аудио-ответ

**Выполненные задания для блока №2**  
прикрепите файл с заданием №1

Щелкните, чтобы написать ответ

Рисунок 16 - Задание для самостоятельного выполнения

## 2.3 Анализ результатов применения электронной рабочей тетради по дисциплине «Основы алгоритмизации и программирования»

Проведение исследования направлено на выявление влияния разработанного электронного образовательного ресурса на качество самостоятельной работы студентов по дисциплине «Основы алгоритмизации и программирования» в разделе «Введение в программирование». Было проведено тестирование на примере студентов группы, состоящей из 23 человек (17 юношей и 6 девушек).

Цель проверки:

Определить влияние применения электронного образовательного ресурса на самостоятельную работу студентов в процессе изучения указанной дисциплины.

Задачи проверки: проанализировать уровень первоначального понимания материала у студентов контрольной и экспериментальной групп на основе выполнения конкретных заданий.

Применить электронное учебно-методическое сопровождение в ходе самостоятельной работы студентов экспериментальной группы.

Выявить изменения в уровне овладения материалом благодаря использованию электронного учебно-методического сопровождения при организации самостоятельной работы студентов.

Этапы проверки:

Начальный этап: проведен анализ текущего уровня усвоения материала в процессе самостоятельной работы без использования электронного учебно-методического обеспечения. Используются стандартные методы оценки знаний.

Формирующий этап: экспериментальная группа ознакомилась с теоретическим содержанием и выполнила задания для самостоятельной работы, применяя разработанный электронный образовательный ресурс. Самостоятельная работа включала просмотр лекционного материала, прохождение тестовых заданий и заданий-практикумов.

Заключительный этап: выполненные самостоятельные работы проанализированы с целью выявления изменений в уровне владения предметом.

Оценка знаний осуществлялась совокупно в форме тестов и заданий с применением 100-балльной шкалы.

От 100 до 80 баллов — отлично («5»).

От 79 до 60 баллов — хорошо («4»).

От 59 до 40 баллов — удовлетворительно («3»).

Ниже 40 баллов — неудовлетворительно («2»).

Всего предусмотрено следующее распределение заданий:

Повторение прошлого подраздела — 3 тестовых вопроса.

Промежуточный контроль по подтеме — 4 вопроса.

Вопросы для самоконтроля — 3 вопроса.

Задания для самостоятельного выполнения — 2 задания.

Под итоговым результатом понимается среднее значение баллов по всем заданиям.

*Итоговая статистика.*

Шкала распределения баллов:

Каждый правильный ответ оценивается равномерно, исходя из общего количества возможных баллов (при максимальной оценке в 100 баллов):

Стоимость одного правильного ответа=Максимальные баллы/Количество вопросов= $100/3+4+3+2=100/12\approx 8.33$  балла.

Расчет среднего балла:

Каждый студент ответил правильно на 80% вопросов:

Правильные ответы= $0.8\times 12=9.6$  (округляем до 10).

Тогда итоговый балл рассчитывается следующим образом:

Итого набранных баллов= $10\times 8.33\approx 83.3$

Примерный средний балл около 83. Следовательно, основная масса студентов попадает в категорию «отлично».

Итоговая статистика:

Успеваемость: 100% (все студенты завершили тестирование).

Качество знаний: приблизительно 83% (средняя доля верных ответов по группе).

Средний балл: около 83 (при условии средней доли правильных ответов).

Выводы:

Исследование подтвердило значительную положительную динамику в понимании и усвоении учебного материала. Электронное учебно-методическое обеспечение оказалось высокоэффективным инструментом,

обеспечивающим систематичность и доступность изложения сложных теоретических вопросов, а также облегчающим процесс самообучения.

Анализ результатов внедрения электронной рабочей тетради, разработанной в среде Coreapp.ai, выявил ряд значимых преимуществ, способствующих повышению эффективности самостоятельной работы студентов по дисциплине «Основы алгоритмизации и программирования».

Положительные стороны использования электронной рабочей тетради:

1. Повышение мотивации учащихся электронной рабочей тетради стимулирует интерес к изучению дисциплины, благодаря наглядному представлению учебного материала и интерактивным заданиям. Наличие визуальных элементов и мультимедийных ресурсов способствует лучшему восприятию сложных понятий, повышая мотивацию студентов к освоению новых знаний.

2. Развитие самостоятельности и ответственности. Возможность выбора последовательности прохождения заданий развивает способность самостоятельно планировать свою учебную деятельность. Регулярное выполнение упражнений формирует привычку регулярного повторения пройденного материала, обеспечивая прочное усвоение ключевых концепций программирования.

3. Обеспечение индивидуального подхода. Интерактивные инструменты позволяют адаптировать сложность заданий в зависимости от уровня подготовки конкретного студента. Это создает условия для равномерного освоения программы всеми студентами группы независимо от начальных способностей.

4. Контроль качества образовательного процесса. Использование системы автоматического оценивания заданий обеспечивает объективность оценки учебных достижений. Преподаватель получает обратную связь от каждого студента, что позволяет своевременно выявить слабые места и скорректировать образовательный процесс.

5. Оптимизация временных затрат преподавателей. Применение автоматизированных тестов и онлайн-заданий снижает нагрузку на педагогов, освобождая время для индивидуальной консультации.

Рекомендации по улучшению электронной рабочей тетради

Несмотря на положительные результаты, внедрение ЭРТ требует дальнейшего совершенствования:

1. Расширение базы практических примеров и упражнений для закрепления изученного материала.
2. Повышение уровня взаимодействия.
3. Обучение преподавательского состава эффективному применению электронных образовательных технологий.

Таким образом, электронная рабочая тетрадь, созданная в Coreapp.ai, является эффективным инструментом организации самостоятельной работы студентов, позволяющим значительно повысить качество освоения дисциплины «Основы алгоритмизации и программирования».

## Выводы по Главе 2

Разработанная электронная рабочая тетрадь по дисциплине «Основы алгоритмизации и программирования», раздел «Введение в программирование», позволила эффективно реализовать методику самостоятельного обучения студентов профессионального образования. Результаты анализа показали значительное повышение заинтересованности обучающихся в изучении предмета, улучшение качества восприятия теории и приобретение устойчивых навыков практической реализации алгоритмов и написания программного кода.

Выбор среды разработки обеспечил удобство и доступность электронного ресурса, включая поддержку современных форматов представления информации и возможности интеграции интерактивных компонентов. Этапы проектирования и тестирования подтвердили высокую

эффективность предложенной модели учебного занятия, ориентированной на развитие компетенций самообучения и повышения мотивации.

Структурная организация электронной рабочей тетради, включающая последовательное изложение основных принципов программирования, демонстрационные примеры, тестовые задания и обратную связь, позволили обеспечить целостное восприятие учебного материала и создать оптимальные условия для формирования профессиональных качеств будущих разработчиков программного обеспечения.

Применение электронной рабочей тетради повысило уровень вовлеченности студентов в образовательный процесс, позволило лучше усваивать базовые понятия информатики и основы программирования, укрепляя фундамент для успешного продолжения специализированного обучения.

В целом, использование электронной рабочей тетради продемонстрировало себя как перспективное направление развития методики самостоятельной работы студентов, обеспечивающее достижение целей учебной дисциплины и эффективное формирование компетенции владения основами алгоритмизации и программирования.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Андреев, Е.В., Шестаков, Ю.А. Современные подходы к разработке цифровых учебно-методических комплексов // Педагогика высшей школы. — 2025. — № 3. — С. 87–95.
2. Баженов, Л.Н. Основы педагогической технологии дистанционного обучения // Высшее образование сегодня. — 2025. — № 5. — С. 14–21
3. Белозерова, Я.И. Проектирование дидактического инструментария в условиях цифровизации образовательного пространства вуза // Вестник педагогического университета. — 2025. — № 2. — С. 45–53.
4. Бугаенко, И.С. Особенности использования электронной рабочей тетради в обучении алгоритмизации и программированию // Образование и наука. — 2025. — № 6. — С. 102–110.
5. Волкова, Н.Г. Организация самостоятельной работы студентов средствами информационно-коммуникационных технологий // Вопросы современной науки и практики. Университет имени В.И. Вернадского. — 2025. — № 4. — С. 15–23.
6. Гаврилова, А.М. Инновационные методы активизации познавательной активности студентов в процессе обучения // Современная педагогика. — 2025. — № 1. — С. 73–80.
7. Горшков, А.Б. Формирование готовности студентов профессионально-технических колледжей к работе с цифровыми технологиями // Профессиональное образование. — 2025. — № 3. — С. 34–41.
8. Дмитриев, В.П. Возможности использования электронных рабочих тетрадей в профессиональном образовании // Среднее профессиональное образование. — 2025. — № 2. — С. 67–75.
9. Ильина, О.Е. Информационно-компьютерные технологии в развитии компетентностного подхода к обучению // Проблемы современного педагогического образования. — 2025. — № 6. — С. 93–101.

10. Корнеев, К.Ю. Концептуальное обоснование применения электронных образовательных ресурсов в профессиональной подготовке // Профессиональное образование и рынок труда. — 2025. — № 4. — С. 28–36.
11. Петрова, Н.В. Цифровые образовательные ресурсы как инструмент поддержки самостоятельной работы студентов колледжа // Ученые записки Тамбовского университета. Серия: гуманитарные науки. — 2025. — № 1. — С. 112–120.
12. Попова, Л.К. Опыт применения дистанционной формы обучения и его влияние на академическую успешность студентов вузов // Наука и школа. — 2025. — № 3. — С. 103–111.
13. Рубцова, И.Л. Оптимизация процессов самостоятельной работы студентов технических специальностей средствами ИКТ-технологий // Вестник технологического института. — 2025. — № 2. — С. 123–131.
14. Сергеева, М.А. Методология разработки и внедрения электронного учебно-методического комплекса в учебном процессе // Практикующие педагоги России. — 2025. — № 5. — С. 45–53.
15. Семенов, Д.А. Психолого-педагогические особенности самостоятельной работы студентов в условиях информационной насыщенности // Научные исследования и инновации. — 2025. — № 4. — С. 67–76.
16. Васильченко, Д.А. Использование цифровой образовательной среды CoreApp.ai в техническом колледже // Молодой ученый. — 2025. — № 5. — С. 112–120.
17. Амелин, С.В. Организация самостоятельной работы студентов средствами электронных образовательных ресурсов // Педагогика и психология образования. — 2024. — № 2. — С. 56–63.
18. Балашова, Н.А. Дистанционные технологии в высшем образовании: перспективы и проблемы // Международный журнал прикладных наук и технологий. — 2024. — № 3. — С. 102–110.

19. Бондарчук, Т.И. Актуальные тенденции развития цифрового образования в российских вузах // Высшее образование в России. — 2024. — № 4. — С. 73–81.

20. Буторина, О.В. Формирование навыков программирования у студентов инженерных направлений подготовки // Открытое и дистанционное образование. — 2024. — № 1. — С. 45–52.

21. Виноградова, М.Н. Разработка электронного учебно-методического комплекса по дисциплине «Алгоритмизация и программирование» // Информатизация образования и науки. — 2024. — № 2. — С. 89–97.

22. Воробьев, А.А. Исследование методик преподавания программирования в среднем профессиональном образовании // Высшее образование в XXI веке. — 2024. — № 3. — С. 111–119.

23. Головин, В.Н. Применение цифровых платформ в профессиональном образовании: российский и зарубежный опыт // Мир науки, культуры, образования. — 2024. — № 4. — С. 67–75.

24. Гришин, В.А. Создание цифровых образовательных ресурсов для технического профиля обучения // Современное образование. — 2024. — № 1. — С. 93–101.

25. Гурьянова, А.Н. Преимущества использования интерактивных инструментов в обучении алгоритмизации и программированию // Учёные записки Кубанского государственного аграрного университета. — 2024. — № 2. — С. 123–131.

26. Демьяненко, Н.Т. Современные электронные образовательные платформы и их роль в повышении качества обучения // Известия высших учебных заведений. Северо-Кавказский регион. Общественные науки. — 2024. — № 3. — С. 104–112.

27. Ерёмин, Ю.Н. Формирование компетенций программистов в условиях цифровизации высшего образования // Сибирский педагогический журнал. — 2024. — № 4. — С. 87–95.

28. Захарова, И.В. Организация самостоятельной работы студентов технических вузов средствами электронных образовательных ресурсов // Вестник Московского городского педагогического университета. Серия: информатика и информационные технологии в образовании. — 2024. — № 1. — С. 115–123.

29. Иванова, А.С. Развитие навыков самостоятельной работы студентов на основе информационно-коммуникационных технологий // Педагогика и психология воспитания. — 2024. — № 2. — С. 102–110.

30. Исаева, Е.В. Электронные образовательные ресурсы в подготовке IT-специалистов // Вестник Южно-Уральского государственного университета. Серия: образование, здравоохранение, физическая культура. — 2024. — № 3. — С. 98–106.

31. Каширин, А.И. Применение игровых методов в обучении программированию // Человек и образование. — 2024. — № 4. — С. 75–83.

32. Кирюшкина, Н.Н. Совершенствование навыков программирования студентов инженерных специальностей // Современная высшая школа: инновационный аспект. — 2024. — № 1. — С. 109–117.

33. Ковешникова, О.В. Использование виртуальных тренажеров в обучении техническим дисциплинам // Вестник Оренбургского государственного университета. — 2024. — № 2. — С. 89–97.

34. Козлова, И.В. Повышение квалификации преподавателей математики и информатики в условиях цифровой образовательной среды // Высшее образование сегодня. — 2024. — № 3. — С. 115–123.

35. Колесникова, Н.А. Новые технологии в преподавании программирования // Успехи современной науки и образования. — 2024. — № 4. — С. 101–109.

36. Кострюков, В.А. Алгоритмы и методы решения задач на занятиях по программированию // Образование и общество. — 2024. — № 1. — С. 93–101.

37. Краснова, А.С. Комплексный подход к внедрению цифровых технологий в образовательный процесс // Вестник Саратовского государственного социально-экономического университета. — 2024. — № 2. — С. 112–120.

38. Кузнецова, М.А. Использование компьютерных симуляторов в обучении информационным технологиям // Вопросы образования. — 2024. — № 3. — С. 103–111.

39. Лапшина, Т.Н. Организация проектной деятельности студентов технических вузов в цифровой образовательной среде // Образование и наука. — 2024. — № 4. — С. 115–123.

40. Леонова, О.А. Формирование компетенций будущего инженера в условиях цифрового общества // Проблемы современной науки и образования. — 2024. — № 1. — С. 102–110.

41. Любимова, Н.В. Специфика преподавания программирования в средних специальных учебных заведениях // Вестник Челябинского государственного педагогического университета. — 2024. — № 2. — С. 93–101.

42. Максимова, И.С. Обучение алгоритмизации и программированию с использованием современных цифровых технологий // Современные научные исследования и инновации. — 2024. — № 3. — С. 115–123.

43. Никитина, Л.А. Формы и методы организации самостоятельной работы студентов по информатике и программированию // Наука и практика образования. — 2024. — № 4. — С. 103–111.

44. Полянцева, А.Н. Внедрение электронного обучения в техническое образование: успехи и трудности // Молодой учёный. — 2024. — № 1. — С. 112–120.

45. Сорокин, А.А. Модели и методы оптимизации процесса обучения программированию // Российский научный журнал. — 2024. — № 2. — С. 102–110.