

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский Томский государственный университет»

Механико-математический факультет



УТВЕРЖДАЮ

Проректор по образовательной
деятельности

Е.В. Луков

(подпись)

« 9 » сентября 2024

**ОСНОВНАЯ ПРОГРАММА
ПРОФЕССИОНАЛЬНОГО ОБУЧЕНИЯ**

программа профессиональной подготовки по профессиям рабочих, должностям служащих

«Помощник программиста»

Код профессии 06.001

(по приказу от 20.07.2022 № 424н)

Квалификационный разряд, класс, категория: 3

Трудоемкость: 148 ч.

Форма обучения: очная

СОГЛАСОВАНО:

Начальник учебного управления

М.А. Игнатьева

Декан механико-математического факультета

Л.В. Гензе

Томск 2024

Основная программа профессионального обучения разработана в соответствии с требованиями профессионального стандарта **06.001 Программист** по профессии рабочего **«Помощник программиста»**

Организация-разработчик: Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Томский государственный университет»

Разработчик: Лещинский Дмитрий Викторович, старший преподаватель кафедры вычислительной математики и компьютерного моделирования ММФ ТГУ.

Руководитель: Лещинский Дмитрий Викторович, старший преподаватель кафедры вычислительной математики и компьютерного моделирования ММФ ТГУ.

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

основной программы профессионального обучения

«Помощник программиста»

№ п/п	Наименование модулей / разделов / тем	Общая трудоёмкость, ч	Контактные часы		СРС, ч	Формы контроля
			Лекции	Практические занятия		
1	Раздел 1. Теоретические основы программирования и алгоритмизации. Язык программирования C++.	38	16	10	12	Зачет
1.1	История развития вычислительной техники. Классы современных компьютеров.	2	2	-	-	
1.2	Принципиальное устройство компьютеров. Оперативная память. Процессор.	2	2	-	-	
1.3	Понятие алгоритма, его свойства. Способы представления алгоритмов. Представление алгоритмов с помощью блок-схем.	6	2	2	2	Задание 1
1.4	Структура программы на языке C/C++. Понятия: типов данных, переменных, условных операторов, операторов присваивания, операторов ввода-вывода.	6	2	2	2	
1.5	Массивы. Указатели. Операторы цикла с пред-, и постусловием.	6	2	2	2	
1.6	Заголовочные файлы и их подключение к проекту. Перегрузка функций.	6	2	2	2	
1.7	Тестирование и отладка программ. Приемы, часто применяемые при отладке. Основные принципы структурного программирования.	6	2	2	2	Задание 2
1.8	Система управления версиями Git.	4	2	-	2	

2	Раздел 2. Структуры данных, строки, функции, файлы на языке C++. Алгоритмы поиска, сортировки, хеширования, оценка их эффективности и сложности.	24	10	4	10	Зачет
2.1	Эффективность и сложность алгоритмов. Классификация алгоритмов по их сложности.	4	2	-	2	
2.2	Работа с информацией, расположенной в таблице. Постановка задач поиска, вставки и удаления элементов таблицы.	6	2	2	2	
2.3	Алгоритмы поиска, вставки и удаления в информации в неупорядоченном массиве. Оценка их эффективности.	6	2	2	2	
2.4	Постановка задачи сортировки одномерного массива. Алгоритмы сортировки. Оценка их эффективности.	6	2	2	2	Задание 3
2.5	Хеширование как метод размещения данных в массиве. Преимущества и недостатки технологии хеширования.	6	2	2	2	
3	Раздел 3. Динамические структуры данных: стек, очередь, список, деревья. Объектно-ориентированное программирование.	24	8	8	8	Зачет
3.1	Стеки, очереди, списки и деревья как структуры данных. Логическое описание. Варианты их реализаций и операций с ними.	12	4	4	4	
3.2	Основы объектно-ориентированного программирования. Классы. Конструкторы. Шаблоны.	10	2	4	4	Задание 4
3.3	Понятие операционной системы и её основные элементы. Принципы взаимодействия прикладных программ с операционными системами.	2	2	-	-	
4	Раздел 4. Базы данных.	26	6	6	14	Зачет
4.1	Архитектура баз данных. Этапы проектирования базы данных. Инфологические и даталогические	8	2	2	4	

	модели данных. Варианты организации хранения базы данных.					
4.2	Язык запросов SQL. Типы данных в SQL. Операторы SQL.	12	2	4	6	Задание 5
4.3	Проблемы безопасности баз данных. Основные методы защиты данных. Понятие транзакции.	6	2	-	4	
5	Раздел 5. Язык программирования Python. Внешние модули	32	8	8	16	Зачет
5.1	Введение в среду программирования Python. Основы синтаксиса, типы данных, операторы, условные операторы, циклы, встроенные функции.	8	2	2	4	
5.2	Решение классических задач программирования для освоения основ синтаксиса языка Python. Знакомство с IDE Jupyter Notebook, Google Colab.	8	2	2	4	
5.3	Модули: numpy, matplotlib, pandas, scipy.	16	4	4	8	Задание 6
Итоговая аттестация		4	-	2	2	Квалификационный экзамен
Итого		148	46	40	62	

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ХАРАКТЕРИСТИКА

основной программы профессионального обучения «Помощник программиста»

ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Нормативно-правовая база

Настоящая программа разработана в соответствии с:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказ от 26 августа 2020 года № 438 «Об утверждении Порядка организации и осуществления образовательной деятельности по основным программам профессионального обучения»;
- Приказ Министерства просвещения РФ от 14 июля 2023 г. № 534 «Об утверждении Перечня профессий рабочих, должностей служащих, по которым осуществляется профессиональное обучение»;
- Профессиональный стандарт 06.001 «Программист» утвержден приказом Министерства труда и социальной защиты Российской Федерации от 20.07.2022г. № 424н.
- локальные нормативные акты ТГУ

1.2. Цель реализации программы

Целью реализации программы является формирование у обучающихся профессиональных знаний, умений и навыков по профессии «Помощник программиста» в рамках 3 уровня квалификации вида профессиональной деятельности «Программист», предусмотренного профессиональным стандартом 06.001 «Программист», с присвоением 3 квалификационного разряда «Помощник программиста».

1.3. Планируемые результаты обучения

Виды деятельности	Профессиональные компетенции ППО	Практический опыт	Умения	Знания
Разработка, отладка, проверка работоспособности, модификация компьютерного программного обеспечения	ПК 1.1. Формализация и алгоритмизация поставленных задач для разработки программного кода (А/01.3)	<ul style="list-style-type: none"> – Составление формализованных описаний решений поставленных задач в соответствии с требованиями технического задания или внутренних документов организации. – Разработка алгоритмов решения поставленных задач в соответствии с требованиями технического задания или внутренних документов организации. – Проверка корректности алгоритмов решения поставленных задач. – Оценка и согласование сроков выполнения поставленных задач. 	<ul style="list-style-type: none"> – Использовать методы и приемы формализации поставленных задач. – Использовать методы и приемы алгоритмизации поставленных задач. – Использовать программное обеспечение для графического отображения алгоритмов. – Применять алгоритмы решения типовых задач в соответствующих областях. – Осуществлять коммуникации с заинтересованными сторонами. 	<ul style="list-style-type: none"> – Методы и приемы формализации поставленных задач. – Языки формализации функциональных спецификаций. – Методы и приемы алгоритмизации поставленных задач. – Нотации и программное обеспечение для графического отображения алгоритмов. – Алгоритмы решения типичных задач, области и способы их применения.
	ПК 1.2. Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных (А/02.3)	<ul style="list-style-type: none"> – Создание программного кода в соответствии с техническим заданием (готовыми спецификациями). – Оптимизация программного кода с использованием специализированных программных средств. – Оценка и согласование сроков выполнения поставленных задач. 	<ul style="list-style-type: none"> – Применять выбранные языки программирования для написания программного кода. – Использовать выбранную среду программирования и средства системы управления базами данных. – Использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода. – Осуществлять коммуникации с заинтересованными сторонами. 	<ul style="list-style-type: none"> – Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования. – Методологии разработки компьютерного программного обеспечения. – Методологии и технологии проектирования и использования баз данных. – Технологии программирования. – Особенности выбранной среды программирования и системы управления базами данных.

				<ul style="list-style-type: none"> – Компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними.
	<p>ПК 1.3. Оформление программного кода в соответствии с установленными требованиями (А/03.3)</p>	<ul style="list-style-type: none"> – Приведение наименований переменных, функций, классов, структур данных и файлов в соответствие с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода. – Структурирование исходного программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода. – Комментирование и разметка программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода. – Форматирование исходного программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода. – Оформление технической документации на компьютерное 	<ul style="list-style-type: none"> – Применять нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода. – Применять инструментарий для создания и актуализации исходных текстов программ. – Применять заданные стандарты и шаблоны для составления и оформления технической документации. – Осуществлять коммуникации с заинтересованными сторонами. 	<ul style="list-style-type: none"> – Инструментарий для создания и актуализации исходных текстов программ. – Методы повышения читаемости программного кода. – Системы кодировки символов, форматы хранения исходных текстов программ. – Нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода. – Основные стандарты оформления технической документации на компьютерное программное обеспечение.

		программное обеспечение по заданному стандарту или шаблону.		
	ПК 1.4. Работа с системой управления версиями программного кода (А/04.3)	<ul style="list-style-type: none"> – Регистрация изменений исходного текста программного кода в системе управления версиями. – Слияние, разделение и сравнение исходных текстов программного кода. – Сохранение сделанных изменений программного кода в соответствии с регламентом управления версиями. 	<ul style="list-style-type: none"> – Использовать выбранную систему управления версиями. – Использовать вспомогательные инструментальные программные средства для обработки исходного текста программного кода. – Выполнять действия, соответствующие установленному регламенту используемой системы управления версиями. – Осуществлять коммуникации с заинтересованными сторонами. 	<ul style="list-style-type: none"> – Возможности используемой системы управления версиями и вспомогательных инструментальных программных средств. – Установленный регламент использования системы управления версиями.
	ПК 1.5. Проверка и отладка программного кода (А/05.3)	<ul style="list-style-type: none"> – Анализ и проверка исходного программного кода. – Отладка программного кода на уровне программных модулей. – Отладка программного кода на уровне межмодульных взаимодействий и взаимодействий с окружением. – Оценка и согласование сроков выполнения поставленных задач. 	<ul style="list-style-type: none"> – Выявлять ошибки в программном коде. – Применять методы и приемы отладки программного кода. – Интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов. – Применять современные компиляторы, отладчики и оптимизаторы программного кода. – Осуществлять коммуникации с заинтересованными сторонами. 	<ul style="list-style-type: none"> – Методы и приемы отладки программного кода. – Типы и форматы сообщений об ошибках, предупреждений. – Способы использования технологических журналов, форматы и типы записей журналов. – Современные компиляторы, отладчики и оптимизаторы программного кода. – Сообщения о состоянии аппаратных средств.

При реализации настоящей программы в рамках конкретной ОПОП соответствие компетенций устанавливается приложением 1.

1.4. Требования к поступающим, категория слушателей: к освоению данной образовательной программы допускаются лица, имеющие среднее общее, среднее профессиональное или высшее образование.

1.5. Трудоемкость обучения: 148 часов, включая все виды аудиторной и самостоятельной работы слушателя и время, отводимое на контроль качества освоения программы

1.6. Форма обучения: очная, в соответствии с календарным графиком основной образовательной программы высшего образования.

МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

Наименование кабинета (мастерской, лаборатории и т.д.)	Вид занятий	Материально-техническое оснащение (наименование оборудования, программного обеспечения)
Учебная аудитория	Лекционные занятия	Технические средства обучения: компьютер, мультимедийный проектор, экран, маркерная доска
Учебная аудитория с компьютерами	Практические занятия	Технические средства обучения: компьютер для преподавателя, мультимедийный проектор, экран, маркерная доска, компьютеры по числу студентов в подгруппе Программное обеспечение: операционная система Windows 7 или Windows 10, интегрированные среды для работы на C/C++, IDE Jupyter Notebook, Google Colab, офисный пакет Microsoft Office 2010

ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ (формы аттестации, оценочные и методические материалы)

Промежуточная аттестация

Промежуточная аттестация проводится по результатам освоения учебного плана в формате выполнения практических заданий по наиболее важным, с точки зрения практических навыков, темам. Формы и процедуры промежуточной аттестации по каждой теме доводятся до сведения обучающихся перед началом учебного процесса. Для допуска к итоговой аттестации (квалификационному экзамену) слушателю необходимо получить зачеты по модулям программы. Каждому слушателю для допуска к итоговой аттестации необходимо выполнить 6 практических заданий.

Итоговая аттестация

Итоговая аттестация результатов подготовки обучающихся осуществляется экзаменационной комиссией в форме квалификационного экзамена. Квалификационный экзамен проводится для определения соответствия полученных знаний, умений и навыков программе профессионального обучения по профессии «Помощник программиста». Квалификационный экзамен включает в себя практическую квалификационную работу в виде написания программы, реализующей предложенный алгоритм и проверку теоретических знаний в виде устного экзамена в пределах квалификационных требований.

Лицам, успешно сдавшим квалификационный экзамен, по результатам освоения программы профессионального обучения выдается свидетельство по профессии «Помощник программиста».

Примеры вопросов теоретической части квалификационного экзамена (теста)

Результаты (освоенные профессиональные компетенции)	Вопросы	Основные показатели оценки результата
ПК 1.1. Формализация и алгоритмизация поставленных задач для разработки программного кода (А/01.3)	Алгоритм и его свойства. Что означает такое свойство алгоритма как "массовость"?	Знание основных подходов к составлению алгоритмов. Понимание подходов к адаптации алгоритма для решения всех задач подобного типа.
	Какие виды памяти используются в ЭВМ?	Знание видов памяти ЭВМ. Понимание в чем заключается иерархия памяти и в чем особенности каждого уровня.

Критерии оценки теоретической части экзамена:

Таблица 1

Баллы	Критерии соответствия
10	Правильно и развернуто изложен материал каждого вопроса и отчетов по индивидуальным заданиям соответствующего раздела. Студент полно, четко и логично излагает материал вопроса и защищаемый материал задания.
8	Правильно изложен материал каждого вопроса и отчетов по индивидуальным заданиям, но не весь материал изложен развернуто и логически структурировано.
6	В целом правильно изложен материал каждого вопроса и защищаемого отчетов по заданию, но изложение носит поверхностный характер и с нарушением логики изложения.
4	Материал ответа на каждый вопрос и защищаемых отчетов по заданиям представлен очень поверхностно и с нарушением логики изложения. Студент очень плохо владеет основными концепциями дисциплины. Допущены существенные терминологические и фактические ошибки.
2	Неверно изложен материал на вопросы билета, каждый отчет по индивидуальным заданиям написан с грубыми ошибками или отчеты вообще не подготовлены к защите.

Теоретическая часть квалификационного экзамена предполагает ответы на 3 теоретических вопроса устно. Ответы на вопросы даются в развернутой форме. Ответы на теоретические вопросы оцениваются согласно критериям оценки (Таблица 1).

Примеры заданий практической части квалификационного экзамена

Результаты (освоенные профессиональные компетенции)	Задания	Основные показатели оценки результата
ПК 1.2. Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных (А/02.3)	Создание таблиц, доменов и индексов в реляционной БД с помощью SQL.	Знание языка запросов SQL и типов данных используемых в нем. Умение применять операторы CREATE, INSERT, DELETE, UPDATE, SELECT.
ПК 1.3. Оформление программного кода в соответствии с установленными требованиями (А/03.3)	На языке С-псевдокода описать алгоритм решения задачи: Пусть x и y – действительные корни квадратного уравнения. Вычислить его коэффициенты.	Знание синтаксиса и функций языка C/C++. Умение описать построенный алгоритм/программный код с помощью С-псевдокода.

Практическая часть квалификационного экзамена предполагает выполнение практического задания по написанию алгоритма или программного кода с комментариями учащегося.

При ответе на теоретические вопросы и выполнении практического задания оценивается полнота и точность ответа, логичность и аргументированность изложения материала, в соответствии с критериями оценки (Таблица 1).

В случае, если слушатель не справился с выполнением теоретической или практической частями квалификационного экзамена, допускается его пересдача в резервный день аттестации.

Максимальное количество баллов за теоретическую и практическую части итогового квалификационного экзамена – 40 баллов.

Итоговая оценка

По результатам квалификационного экзамена выставляется оценка по пятибалльной шкале и присваивается квалификация. На основании баллов, полученных за теоретическую часть квалификационного экзамена и ее защиту, практическую часть квалификационного экзамена слушателю выставляется итоговая оценка по следующим критериям:

«Отлично» – выставляется, если количество баллов за теоретическую и практическую часть составляет 35-40 баллов.

«Хорошо» – выставляется, если количество баллов за теоретическую и практическую часть составляет 29-34 баллов.

«Удовлетворительно» – выставляется, если количество баллов за теоретическую и практическую часть составляет 23-28 баллов.

«Неудовлетворительно» – выставляется, если количество баллов составляет менее 22 баллов (**программа не освоена слушателем**).

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

РАБОЧАЯ ПРОГРАММА Помощник программиста

Рабочая программа Помощник программиста является частью основной программы профессионального обучения «Программирование» по профессии «Программист» и определяет результаты, содержание и условия обучения, обеспечивающие освоение вида деятельности (ВД): Разработка, отладка, проверка работоспособности, модификация компьютерного программного обеспечения.

Результатом освоения программы теоретической подготовки является овладение обучающимися видом деятельности: Разработка, отладка, проверка работоспособности, модификация компьютерного программного обеспечения, в том числе профессиональными (ПК) компетенциями, в соответствии с профессиональным стандартом 06.001 «Программист», с присвоением рабочей профессии «Помощник программиста»:

Код	Наименование результата обучения
ПК 1.1. Формализация и алгоритмизация поставленных задач для разработки программного кода (А/01.3)	Знать: <ul style="list-style-type: none">– Методы и приемы формализации поставленных задач.– Языки формализации функциональных спецификаций.– Методы и приемы алгоритмизации поставленных задач.– Нотации и программное обеспечение для графического отображения алгоритмов.– Алгоритмы решения типичных задач, области и способы их применения. Уметь: <ul style="list-style-type: none">– Использовать методы и приемы формализации поставленных задач.– Использовать методы и приемы алгоритмизации поставленных задач.– Использовать программное обеспечение для графического отображения алгоритмов.– Применять алгоритмы решения типовых задач в соответствующих областях.– Осуществлять коммуникации с заинтересованными сторонами.
ПК 1.2. Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных (А/02.3)	Знать: <ul style="list-style-type: none">– Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования.– Методологии разработки компьютерного программного обеспечения.– Методологии и технологии проектирования и использования баз данных.– Технологии программирования.– Особенности выбранной среды программирования и системы управления базами данных.– Компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними. Уметь: <ul style="list-style-type: none">– Применять выбранные языки программирования для написания программного кода.– Использовать выбранную среду программирования и средства системы управления базами данных.– Использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода.– Осуществлять коммуникации с заинтересованными сторонами.
ПК 1.3. Оформление программного кода в соответствии с	Знать: <ul style="list-style-type: none">– Инструментарий для создания и актуализации исходных текстов программ.

установленными требованиями (А/03.3)	<ul style="list-style-type: none"> – Методы повышения читаемости программного кода. – Системы кодировки символов, форматы хранения исходных текстов программ. – Нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода. – Основные стандарты оформления технической документации на компьютерное программное обеспечение <p>Уметь:</p> <ul style="list-style-type: none"> – Применять нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода. – Применять инструментарий для создания и актуализации исходных текстов программ. – Применять заданные стандарты и шаблоны для составления и оформления технической документации. – Осуществлять коммуникации с заинтересованными сторонами.
ПК 1.4. Работа с системой управления версиями программного кода (А/04.3)	<p>Знать:</p> <ul style="list-style-type: none"> – Возможности используемой системы управления версиями и вспомогательных инструментальных программных средств. – Установленный регламент использования системы управления версиями. <p>Уметь:</p> <ul style="list-style-type: none"> – Использовать выбранную систему управления версиями. – Использовать вспомогательные инструментальные программные средства для обработки исходного текста программного кода. – Выполнять действия, соответствующие установленному регламенту используемой системы управления версиями. – Осуществлять коммуникации с заинтересованными сторонами.
ПК 1.5. Проверка и отладка программного кода (А/05.3)	<p>Знать:</p> <ul style="list-style-type: none"> – Методы и приемы отладки программного кода. – Типы и форматы сообщений об ошибках, предупреждений. – Способы использования технологических журналов, форматы и типы записей журналов. – Современные компиляторы, отладчики и оптимизаторы программного кода. – Сообщения о состоянии аппаратных средств. <p>Уметь:</p> <ul style="list-style-type: none"> – Выявлять ошибки в программном коде. – Применять методы и приемы отладки программного кода. – Интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов. – Применять современные компиляторы, отладчики и оптимизаторы программного кода. – Осуществлять коммуникации с заинтересованными сторонами.

СОДЕРЖАНИЕ ОБУЧЕНИЯ

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических занятий, с указанием формата работы (кол-во часов)	Виды СРС (кол-во часов)
Раздел 1. Теоретические основы программирования и алгоритмизации. Язык программирования С++. (38 ч)			
Тема 1.1 История развития вычислительной техники. Классы современных компьютеров. (2 ч)	История развития вычислительной техники. Классы современных компьютеров. (2 ч)		

Тема 1.2 Принципиальное устройство компьютеров. Оперативная память. Процессор. (2 ч)	Принципиальное устройство компьютеров. Оперативная память. Процессор. (2 ч)		
Тема 1.3 Понятие алгоритма, его свойства. Способы представления алгоритмов. Представление алгоритмов с помощью блок-схем. (6 ч)	Понятие алгоритма, его свойства. Способы представления алгоритмов. Представление алгоритмов с помощью блок-схем. (2 ч)	Практическая работа по представлению алгоритмов в различных видах (блок-схем и псевдокодов и др.) (2 ч)	Изучение литературы и методических материалов по теме. Выполнение самостоятельного практического задания. (2 ч)
Тема 1.4 Структура программы на языке C/C++. Понятия: типов данных, переменных, условных операторов, операторов присваивания, операторов ввода-вывода. (6 ч)	Структура программы на языке C/C++. Понятия: типов данных, переменных, условных операторов, операторов присваивания, операторов ввода-вывода. (2 ч)	Практическая работа по изучению синтаксиса языка C/C++ и различных структур, используемых в нем. (2 ч)	Изучение литературы и методических материалов по теме. (2 ч)
Тема 1.5 Массивы. Указатели. Операторы цикла с пред-, и постусловием. (6 ч)	Массивы. Указатели. Операторы цикла с пред-, и постусловием. (2 ч)	Практическая работа по изучению синтаксиса использования массивов, указателей и операторов цикла на языке C/C++. (2 ч)	Изучение литературы и методических материалов по теме. (2 ч)
Тема 1.6 Заголовочные файлы и их подключение к проекту. Перегрузка функций (6 ч)	Заголовочные файлы и их подключение к проекту. Перегрузка функций (2 ч)	Практическая работа по применению заголовочных файлов к проектам на языке C/C++. (2 ч)	Изучение литературы и методических материалов по теме. (2 ч)
Тема 1.7 Тестирование и отладка программ. Приемы, часто применяемые при отладке. Основные принципы структурного программирования. (4 ч)	Тестирование и отладка программ. Приемы, часто применяемые при отладке. Основные принципы структурного программирования. (2 ч)		Изучение литературы и методических материалов по теме. Выполнение самостоятельного практического задания. (2 ч)
Тема 1.8 Система управления версиями Git. (6 ч)	Система управления версиями Git. (2 ч)	Практическая работа по использованию системы управления версиями – Git. (2 ч)	Изучение литературы и методических материалов по теме. (2 ч)
Раздел 2. Структуры данных, строки, функции, файлы на языке C++. Алгоритмы поиска, сортировки, хеширования, оценка их эффективности и сложности. (24 ч)			
Тема 2.1 Эффективность и сложность алгоритмов. Классификация алгоритмов по их сложности. (4 ч)	Эффективность и сложность алгоритмов. Классификация алгоритмов по их сложности. (2 ч)		Изучение литературы и методических материалов по теме. (2 ч)
Тема 2.2 Работа с информацией, расположенной в таблице. Постановка задач поиска, вставки и удаления элементов таблицы. (6 ч)	Работа с информацией, расположенной в таблице. Постановка задач поиска, вставки и удаления элементов таблицы. (2 ч)	Практика по работе с таблицами. (2 ч)	Изучение литературы и методических материалов по теме. (2 ч)

Тема 2.3 Алгоритмы поиска, вставки и удаления в информации в неупорядоченном массиве. Оценка их эффективности. (6 ч)	Алгоритмы поиска, вставки и удаления в информации в неупорядоченном массиве. Оценка их эффективности. (2 ч)	Практическое занятие по изученным алгоритмам поиска, вставки и удаления. (2 ч)	Изучение литературы и методических материалов по теме. (2 ч)
Тема 2.4 Постановка задачи сортировки одномерного массива. Алгоритмы сортировки. Оценка их эффективности. (6 ч)	Постановка задачи сортировки одномерного массива. Алгоритмы сортировки. Оценка их эффективности. (2 ч)	Применение алгоритмов сортировки. (2 ч)	Изучение литературы и методических материалов по теме. Выполнение самостоятельного практического задания. (2 ч)
Тема 2.5 Хеширование как метод размещения данных в массиве. Преимущества и недостатки технологии хеширования. (6 ч)	Хеширование как метод размещения данных в массиве. Преимущества и недостатки технологии хеширования. (2 ч)	Освоение технологии хеширования (2 ч)	Изучение литературы и методических материалов по теме. (2 ч)
Раздел 3. Динамические структуры данных: стек, очередь, список, деревья. Объектно-ориентированное программирование. (24 ч)			
Тема 3.1 Стеки, очереди, списки и деревья как структуры данных. Логическое описание. Варианты их реализаций и операций с ними. (12 ч.)	Стеки, очереди, списки и деревья как структуры данных. Логическое описание. Варианты их реализаций и операций с ними. (4 ч)	Практическая работа по применению динамических структур данных. (4 ч)	Изучение литературы и методических материалов по теме. (4 ч)
Тема 3.2. Основы объектно-ориентированного программирования. Классы. Конструкторы. Шаблоны. (10 ч)	Основы объектно-ориентированного программирования. Классы. Конструкторы. Шаблоны. (2 ч)	Практическая работа по изучению синтаксиса языка C/C++ для объектно-ориентированного программирования. (4 ч)	Изучение литературы и методических материалов по теме. Выполнение самостоятельного практического задания. (4 ч)
Тема 3.3. Понятие операционной системы и её основные элементы. Принципы взаимодействия прикладных программ с операционными системами. (2 ч)	Понятие операционной системы и её основные элементы. Принципы взаимодействия прикладных программ с операционными системами. (2 ч)		
Раздел 4. Базы данных. (26 ч)			
Тема 4.1. Архитектура баз данных. Этапы проектирования базы данных. Инфологические и даталогические модели данных. Варианты организации хранения базы данных. (8 ч)	Архитектура баз данных. Этапы проектирования базы данных. Инфологические и даталогические модели данных. Варианты организации хранения базы данных. (2 ч)	Практическая работа по изученному материалу. (2 ч)	Изучение литературы и методических материалов по теме. (4 ч)
Тема 4.2. Язык запросов SQL. Типы данных в SQL. Операторы SQL. (12 ч)	Язык запросов SQL. Типы данных в SQL. Операторы SQL. (10 ч)	Практическая работа по языку запросов SQL. Освоение синтаксиса. (4 ч)	Изучение литературы и методических материалов по теме. Выполнение самостоятельного практического задания. (6 ч)

Тема 4.3. Проблемы безопасности баз данных. Основные методы защиты данных. Понятие транзакции. (6 ч)	Проблемы безопасности баз данных. Основные методы защиты данных. Понятие транзакции. (2 ч)		Изучение литературы и методических материалов по теме. (4 ч)
Раздел 5. Язык программирования Python. Внешние модули. (32 ч)			
Тема 5.1. Введение в среду программирования Python. Основы синтаксиса, типы данных, операторы, условные операторы, циклы, встроенные функции. (8 ч)	Введение в среду программирования Python. Основы синтаксиса, типы данных, операторы, условные операторы, циклы, встроенные функции. (2 ч)	Практическая работа по изучению синтаксиса языка Python. (2 ч)	Изучение литературы и методических материалов по теме. (4 ч)
Тема 5.2. Решение классических задач программирования для освоения основ синтаксиса языка Python. Знакомство с IDE Jupyter Notebook, Google Colab. (8 ч)	Решение классических задач программирования для освоения основ синтаксиса языка Python. Знакомство с IDE Jupyter Notebook, Google Colab. (2 ч)	Практическая работа по применению языка Python при решении классических задач. (2 ч)	Изучение литературы и методических материалов по теме. (4 ч)
Тема 5.3. Модули: numpy, matplotlib, pandas, scipy. (16 ч)	Модули: numpy, matplotlib, pandas, scipy. (4 ч)	Практическая работа по применению numpy, matplotlib, pandas, scipy в языке Python. (4 ч)	Изучение литературы и методических материалов по теме. Выполнение самостоятельного практического задания. (8 ч)

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ (организационно-педагогические)

Материально-технические условия реализации программы:

Обучение по программе реализовано в очном формате, с применением активных технологий совместного обучения в электронной среде. Материал для самостоятельного изучения представляется в виде текстовых материалов, презентаций, размещаемых в LMS Moodle. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин.

Учебно-методическое и информационное обеспечение программы:

Методические рекомендации и пособия по изучению курса

Программа реализуется в формате очного обучения, с применением активных технологий совместного обучения в электронной среде LMS Moodle. Обучение в очной части реализуется в виде лекционных и практических занятий.

По данной программе имеется электронный учебно-методический комплекс в LMS Moodle. УМК содержит: систему навигации по программе (учебно-тематический план, интерактивный график работы по программе, сведения о результатах обучения, о преподавателях программы, чат для объявлений и вопросов преподавателям), набор видео-лекций, презентации к лекциям, набор ссылок на внешние образовательные ресурсы и инструменты, систему заданий с подробными установкам, инструкции по работе с цифровыми сервисами, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи, а также онлайн-площадки для взаимного обучения.

Литература

1. Керниган Б., Ритчи Д. Язык программирования Си. СПб.: «Невский диалект», 2001.
2. Т. А. Павловская. С, С++. Программирование на языке высокого уровня. СПб.: ПИТЕР, 2002.
3. В. В. Подбельский Язык С++. М.: Финансы и статистика, 1999.
4. Т. А. Павловская, Ю. А. Щупак. С, С++. Структурное программирование: Практикум. СПб.:ПИТЕР, 2002.
5. Васильев А.Н. Программирование на С++ в примерах и задачах.
6. Конова Е., Поллак Г. Алгоритмы и программы. С++. СПб.: Издательство «Лань», 2017.384 с.
7. Ульман Дж. Д., Уидом Дж. Введение с системы баз данных. - М.: Издательство «Лори», 2000.
8. Ю.Б. Бакаревич, Н.В.Пушкина. Самоучитель MS Office 2016. – СПб., БХВ-Петербург, 2017, 418 с
9. Любанович Б. Простой Python. Современный стиль программирования. 2-е. изд. – СПб.: Питер, 2021. – 592 с.
10. Чан. Дж. Python: быстрый старт. — СПб.: Питер, 2021. — 224 с.
11. Стивенсон Б. Python. Сборник упражнений / пер. с англ. А. Ю. Гинько. – М.: ДМК Пресс, 2021. – 238 с.
12. Лутц М.. Python. Карманный справочник. – М.: Вильямс, 2019. – 320 с.
13. Вабищевич Н. П. Численные методы. Вычислительный практикум. Практическое применение численных методов при использовании алгоритмического языка Python. – Москва: USSR, 2020. – 320 с.
14. Хайбрахманов С. А. Основы научных расчётов на языке программирования Python : УП. — Челябинск : Изд-во Челяб. гос. ун-та, 2019. — 96 с.
15. Криволапов С. Я., Хрипунова М. Б. Математика на Python: учебник. – Москва: КНОРУС, 2022. – 456 с.

Дополнительная литература

1. Вопросы и ответы по С и С++. Москва: «МикроАрт», 1997.
2. Брайан Оверленд. С++ без страха. М.; изд-во «Триумф», 2005.
3. Йодан Э. Структурное программирование и конструирование программ. М.:Мир, 1979.
4. Дейкстра. Дисциплина программирования. М.: Мир, 1978.
5. Вирт Н. Алгоритмы и структуры данных. М: Мир, 1985.
6. Мейер В., Бодуэн К. Методы программирования. М.:Мир,1982, Т.1,2.
7. Могилев А.В., Пак Н.И., Хеннер Е.К. Информатика. М.- Академия, 1999.
8. Дейт К. Дж. Введение в системы баз данных. – К.; М.; СПб.: Издательский дом «Вильямс», 1999
9. Задачник по программированию для математиков. Томск: Изд-во Том. ун-та, 2001.
10. Страуструп Б. Программирование. Принципы и практика использования С++. Москва:Вильямс , 2011 – 1238 стр.
11. Дейтел П., Дейтел Х. Python: Искусственный интеллект, большие данные и облачные вычисления. — СПб.: Питер, 2020. — 864 с.
12. Нуньес-Иглесиас Х., Уолт ван дер Ш., Дэшноу Х. Элегантный SciPy. Научное программирование на Python. – М. ДМК Пресс, 2018. – 266 с.
13. Брюс, П. Практическая статистика для специалистов Data Science. 2-е. изд., перераб. и доп. – СПб.: БХВ-Петербург, 2021. – 352 с.

14. Кристиан Хилл. Научное программирование на Python / пер. с англ. А. В. Снастина. – М.: ДМК Пресс, 2021. – 646 с.

КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОГРАММЫ (формы аттестации, оценочные и методические материалы)

Для получения зачета по данному модулю слушатель должен выполнить все практические задания (практические задания 1-6) и суммарно набрать не менее 6 баллов.

Практические задания оцениваются по следующей шкале:

– 0 баллов – задание не принято: задание не выполнено или выполнено не полностью;

– 1 балл – задание принято: присутствуют небольшие недочеты, некритичные для данной работы;

– 2 балла – задание принято: работа выполнена полностью в соответствии с заданием.

Перечень практических работ:

1. Практическое задание №1. «Представление алгоритмов с помощью блок-схем»

Составить блок-схему алгоритма для вычисления корней квадратного уравнения в зависимости от вида его коэффициентов.

2. Практическое задание №2. «Написание программного кода на языке C/C++»

Написать программу на языке C/C++ для вычисления корней квадратного уравнения в зависимости от вида его коэффициентов.

3. Практическое задание №3. «Алгоритмы сортировки»

Реализовать на языке C/C++ «пузырьковый» алгоритм сортировки для предложенного массива данных.

4. Практическое задание №4. «Объектно-ориентированное программирование»

Реализовать алгоритм для вычисления корней квадратного уравнения в зависимости от вида его коэффициентов средствами объектно-ориентированного программирования.

5. Практическое задание №5. «Базы данных»

Таблицы: пассажиры, маршруты, самолеты, авиакомпании, билеты. Реализовать возможность на основе нескольких таблиц распечатать посадочный талон, в котором указывается направление полета, дата и время вылета и прилета, пассажир, посадочное место, цена полета и дополнительные услуги.

6. Практическое задание №6. «Написание программного кода на языке Python»

Написать программу на языке Python для вычисления корней квадратного уравнения в зависимости от вида его коэффициентов. Показать результат нахождения корней графически с использованием библиотеки matplotlib.