



Государственное бюджетное образовательное учреждение высшего образования
Московской области

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

КОЛЛЕДЖ КОСМИЧЕСКОГО МАШИНОСТРОЕНИЯ И ТЕХНОЛОГИЙ



РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.06 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

09.02.04 «Информационные системы (по отраслям)»

Базовой подготовки

Королев, 2020

Автор: Соколова Наталья Валентиновна. Рабочая программа дисциплины «Основы алгоритмизации и программирования». – Королев МО: МГОТУ, 2020 - 22 с.

Рабочая программа учебной дисциплины составлена в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования (далее - ФГОС СПО) и учебного плана по специальности 09.02.04 «Информационные системы (по отраслям)».

Рабочая программа рассмотрена и одобрена на заседании цикловой комиссии по специальности 09.02.04 «Информационные системы (по отраслям)» 28.08.2020 г., протокол № 1.

Рабочая программа утверждена на заседании УМС ГБОУ ВО МО «Технологический университет» 31.08.2020 г., протокол № 1.

СОДЕРЖАНИЕ

	стр.
1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	7
3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ	17
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	19

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

Основы алгоритмизации и программирования

1.1. Область применения рабочей программы

Рабочая программа учебной дисциплины «Основы алгоритмизации и программирования» является частью программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности 09.02.04. «Информационные системы (по отраслям)» СПО

1.2. Общие и профессиональные компетенции, полученные в результате освоения учебной дисциплины

Общие компетенции

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

Профессиональные компетенции

ПК.1.2 Взаимодействовать со специалистами смежного профиля при разработке методов, средств и технологий применения объектов профессиональной деятельности.

ПК.1.3 Производить модификацию отдельных модулей информационной системы в соответствии с рабочим заданием, документировать произведенные изменения.

ПК.2.2 Программировать в соответствии с требованиями технического задания.

ПК.2.3 Применять методики тестирования разрабатываемых приложений.

1.3. Место учебной дисциплины в структуре основной профессиональной образовательной программы:

Учебная дисциплина «Основы алгоритмизации и программирования» является общепрофессиональной дисциплиной, формирующей базовый уровень знаний для освоения специальных дисциплин.

1.4. Цели и задачи учебной дисциплины – требования к результатам освоения учебной дисциплины:

- **в результате освоения дисциплины обучающийся должен уметь:**
 - использовать языки программирования, строить логически правильные и эффективные программы;
- **в результате освоения дисциплины обучающийся должен знать:**
 - общие принципы построения алгоритмов, основные алгоритмически конструкции;
 - понятие системы программирования;
 - основные элементы процедурного языка программирования, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;

- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования, понятие классов и объектов, их свойств и методов.

1.5. Рекомендуемое количество часов на освоение рабочей программы учебной дисциплины:

максимальной учебной нагрузки студента 362 часа, в том числе:

- обязательной аудиторной учебной нагрузки обучающегося 245 часов;
- самостоятельной работы обучающегося 117 часов.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	362
Обязательная аудиторная учебная нагрузка (всего)	245
в том числе:	
лабораторные занятия	112
практические занятия	0
Самостоятельная работа обучающегося (всего)	117
в том числе:	
внеаудиторная самостоятельная работ: работа над материалом учебников [1], [2] [3], [4], конспектом лекций;	30
выполнение индивидуальных заданий, творческие работы разных видов, поиск информации в сети Интернет, подготовка материала для исследовательской (проектной) деятельности (тематика самостоятельной работы);	21
подготовка к лабораторным и практическим занятиям, оформление отчетов по выполненным работам	66
<i>Итоговая аттестация в форме экзамена</i>	

2.2. Тематический план и содержание учебной дисциплины «Основы алгоритмизации и программирования»

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся	Объем часов	Уровень освоения
1	2	3	4
Введение в дисциплину "Основы алгоритмизации и программирования". Связь с другими дисциплинами		1	
Раздел 1. Основы алгоритмизации		72	
Тема 1.1. Алгоритм. Основные алгоритмические конструкции	Содержание учебного материала		
	1.1.1. Введение в дисциплину "Основы алгоритмизации и программирования". Связь с другими дисциплинами Алгоритмы. Свойства алгоритмов. Способы описания алгоритмов.		
	1.1.2. Основные конструкции алгоритмического языка: линейный алгоритм, ветвление, цикл.		
	Лабораторные занятия 1. Построение блок-схем и трассировочных таблиц линейных и разветвляющихся алгоритмов в программе Dia. 2. Построение блок-схем и трассировочных таблиц циклических алгоритмов в программе Dia		
	Самостоятельная работа Подготовка к лабораторным занятиям, оформление отчетов по выполненным работам. Работа над материалом учебников [1], [2] конспектом лекций, решение задач по образцу.		
Тема 1.2. Алгоритмы решения типовых задач	Содержание учебного материала		
	1.2.1. Алгоритм вычисления значения функции		
	1.2.2. Алгоритм вычисления суммы числового ряда		
	1.2.3. Алгоритм заполнения массива.		
	1.2.4. Алгоритм поиска минимального элемента		
	1.2.5. Алгоритмы сортировки		
	1.2.6. Алгоритмы сдвига и перестановок в массиве		
	1.2.7. Алгоритмы обработки строк		
Лабораторные занятия			

	<ol style="list-style-type: none"> 1. Построение блок-схем и трассировочных таблиц алгоритма вычисления значения функции 2. Построение блок-схем и трассировочных таблиц алгоритма вычисления суммы числового ряда 3. Построение блок-схем и трассировочных таблиц алгоритма заполнения массива. 4. Построение блок-схем и трассировочных таблиц алгоритма поиска минимального элемента 5. Построение блок-схем и трассировочных таблиц алгоритмов сортировки 6. Построение блок-схем и трассировочных таблиц алгоритмов сдвига и перестановок в массиве 7. Построение блок-схем и трассировочных таблиц алгоритмов обработки строк 		
	<p>Самостоятельная работа Подготовка к лабораторным и практическим занятиям, оформление отчетов по выполненным работам. Работа над материалом учебников [1], [2] конспектом лекций, решение задач по образцу.</p>		
<p>Раздел 2. Язык программирования C#</p>		72	
<p>Тема 2.1. Этапы решения задач на ПК. Базовые средства языка C#</p>	<p>Содержание учебного материала</p> <p>2.1.1. Основные этапы решения задач на ПК. Постановка задачи и спецификация программы. Тестирование, анализ и отладка программ. Корректность программ. Документирование программного продукта.</p> <p>2.1.2. Платформа Microsoft .Net Framework, Visual Studio, язык C#, Решения, проекты, пространства имен. Консольные и Windows-приложения C#.</p> <p>2.1.3. Классы, объекты, свойства, события. Методы, конструкторы, события и обработчики событий языка программирования C#.</p> <p>2.1.4. Система типов. Типы-значения и ссылочные типы. Встроенные типы. Преобразования типов. Преобразования внутри арифметического типа. Преобразования строкового типа. Класс Convert и его методы.</p> <p>2.1.5. Объявление переменных. Синтаксис объявления. Инициализация. Время жизни и область видимости. Построение выражений. Операции и их</p>		

	<p>приоритеты. Описание операций. Присваивание. Классы Math, Random и встроенные функции.</p>		
	<p>2.1.6. Операторы языка C#. Оператор присваивания. Составной оператор. Пустой оператор. Операторы выбора. If-оператор. Switch-оператор. Операторы перехода. Оператор goto. Операторы break, continue. Операторы цикла. For-оператор. Циклы while. Цикл foreach.</p>		
	<p>Лабораторные занятия 1. Разработка алгоритма вычисления значения функции, составление программы, тестирование и отладка.</p>		
	<p>Самостоятельная работа Подготовка к лабораторным занятиям, оформление отчетов по выполненным работам. Работа над материалом учебников [1], [2], [3], конспектом лекций, решение задач по образцу. Реферат по темам «Эволюция языков программирования», «Классификация языков программирования» и др.</p>		
<p>Тема 2.2. Реализация базовых алгоритмов на языке C#</p>	<p>Содержание учебного материала</p> <p>2.2.1. Тестирование и отладка программного кода. Составление набора тестовых данных.</p> <p>2.2.2. Оптимизация программного кода. Ограничения по входным параметрам. Форматы вывода результатов. Сравнительный анализ эффективности алгоритмов.</p> <p>2.2.3. Массивы в C#. Алгоритмы заполнения и вывода одномерного массива.</p> <p>2.2.4. Алгоритмы поиска минимального (максимального) элемента одномерного массива.</p> <p>2.2.5. Разработка алгоритма обработки одномерного массива (поиск по критерию, перестановки)</p> <p>2.2.6. Обработка числовых серий в массиве</p> <p>2.2.7. Алгоритмы сортировки</p> <p>2.2.8. Сравнительный анализ алгоритмов сортировки</p> <p>2.2.9. Поиск в неупорядоченном и упорядоченном массиве</p> <p>2.2.10. Многомерные массивы. Алгоритмы заполнения и вывода</p> <p>2.2.11. Алгоритмы обработки многомерных массивов (поиск по критерию, перестановки, сдвиги).</p>		

	2.2.12. Алгоритмы обработки совокупности массивов (сопоставление значений элементов различных массивов, формирование значений элементов массива на основе данных нескольких исходных массивов)		
	2.2.13. Строковый тип данных. Алгоритмы обработки строк.		
	Обобщение теоретического материала. Подведение итогов		
	<p>Лабораторные занятия</p> <ol style="list-style-type: none"> 1. Разработка алгоритма обработки целочисленных числовых значений, составление программы, тестирование и отладка. 2. Разработка алгоритма с использованием логических переменных, логических конструкции, составление программы, тестирование и отладка. 3. Разработка алгоритма использованием операторов ветвления, составление программы, тестирование и отладка 4. Разработка алгоритма поиск минимального-максимального элемента в наборе, составление программы, тестирование и отладка. 5. Разработка алгоритма вычисления суммы и произведения конечной числовой последовательности или частичной суммы и произведения бесконечной числовой последовательности, составление программы, тестирование и отладка. 6. Разработка алгоритма поиска минимального элемента конечной числовой последовательности, проверка на упорядоченность, составление программы, тестирование и отладка. 7. Разработка алгоритма заполнения одномерного массива, составление программы, тестирование и отладка. 8. Разработка алгоритма обработки одномерного массива (поиск суммы, произведения), составление программы, тестирование и отладка. 9. Разработка алгоритма обработки одномерного массива (поиск по критерию, перестановки), составление программы, тестирование и отладка. 10. Разработка алгоритма сортировки одномерного массива, составление программы, тестирование и отладка. 11. Разработка алгоритма заполнения двумерного массива, составление программы, тестирование и отладка. 12. Разработка алгоритма обработки двумерного массива (поиск по критерию), составление программы, тестирование и отладка. 13. Разработка алгоритма обработки двумерного массива (работа со строками и столбцами массива), составление программы, тестирование и 		

	отладка. 14. Разработка алгоритма обработки строк, составление программы, тестирование и отладка.		
	Самостоятельная работа Подготовка к лабораторным занятиям, оформление отчетов по выполненным работам. Работа над материалом учебников [1], [2], [3], конспектом лекций, решение задач по образцу.		
Раздел 3. Создание пользовательского интерфейса в Visual C#		72	
Тема 3.1. Программирование для Windows. Форма и элементы управления C#	Содержание учебного материала		
	3.1.1. Добавление элементов управления. Задание свойств. Обработка событий.		
	3.1.2. Отображение текста в форме Windows Forms. Использование элемента управления "TextBox" для получения вводимых данных Использование элемента управления "Button"		
	3.1.3. Переключатель RadioButton. Флажок CheckBox. Вывод сообщений. Функция MessageBox		
	3.1.4. Объекты - PictureBox, MenuStrip, ToolStrip, Timer.		
	3.1.5. Позиционирование объектов на форме.		
	3.1.6. Отображение даты и времени (Calendar, DateTimePicker)		
	3.1.7. Добавление файлов мультимедиа в приложение		
	3.1.8. Массив из объектов (элементов управления). Создание, удаление		
	3.1.9. Обработка массивов из объектов		
	3.1.10. Метки, индикаторы прогресса и бегунки		
	3.1.11. Списки и деревья (ListView и Tree View)		
	3.1.12. Список изображений ImageList		
	3.1.13. Использование ImageList и ListView или Tree View		
3.1.14. Полосы прокрутки.			

	<p>Лабораторные занятия</p> <ol style="list-style-type: none"> 1. Разработка приложений, реализующих вычисление значений функций. 2. Разработка проекта "Калькулятор" 3. Разработка приложений с использованием RadioButton, CheckBox. 4. Разработка проекта "Тестирование" 5. Разработка приложений с использованием PictureBox, MenuStrip, ToolStrip, Timer. 6. Разработка проекта "Анимация" 7. Разработка проекта "Определение интервала между двумя датами" 8. Разработка приложений с использованием файлов мультимедиа 9. Разработка приложений с использованием массив из объектов PictureBox 10. Работа со спискам 11. Работа с деревом 		
	<p>Самостоятельная работа</p> <p>Подготовка к лабораторным и практическим занятиям, оформление отчетов по выполненным работам. Работа над материалом учебников [1], [2], [3], конспектом лекций, решение задач по образцу.</p>		
<p>Тема 3.2. Проектирование и разработка пользовательского интерфейса</p>	<p>Содержание учебного материала</p> <p>3.2.1. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя. Программирование приложения. Тестирование, отладка приложения. Создание документации.</p> <p>3.2.2. Файловый ввод-вывод. Создание и удаление с диска.</p> <p>3.2.3. Меню. Создание головного и вложенного меню. Обработка сообщений меню. Контекстное меню.</p> <p>3.2.4. Разработка функционального интерфейса и схемы работы многооконного приложения.</p> <p>3.2.5. Создание процедур обработки событий. Компиляция и запуск приложения.</p> <p>Обобщение материала. Подведение итогов. Контроль знаний.</p>		
	<p>Лабораторные занятия</p>		

	<p>1. Разработка меню приложения</p> <p>2. Проектирование экранной формы (однооконного приложения). Индивидуальные задания.</p> <p>3. Разработка и тестирование приложения по индивидуальному заданию.</p>		
	<p>Самостоятельная работа</p> <p>Подготовка к лабораторным и практическим занятиям, оформление отчетов по выполненным работам. Работа над материалом учебников [1], [2], [3], конспектом лекций, решение задач по образцу.</p>		
Раздел 4. Объектно-ориентированное программирование в С#		72	
4.1. Реализация базовых принципов ООП в С#	<p>Содержание учебного материала</p> <p>4.1.1. Объектно-ориентированное программирование. Классы и объекты. Объявление классов и создание объектов. Методы. Конструкторы. Оператор this</p> <p>4.1.2. Наследование в С#. Конструктор базового класса.</p> <p>4.1.3. Полиморфизм. Виртуальные методы в С#. Переопределение методов</p> <p>Разработка приложения с использованием собственных классов</p> <p>4.1.4. Абстрактные классы, методы и свойства. Перегрузка методов в С#.</p> <p>4.1.5. Интерфейсы в С#. Множественное наследование</p> <p>Разработка приложения с использованием классов "наследования"</p> <p>4.1.6. Инкапсуляция в С#. Модификаторы доступа. Перегрузка операторов</p> <p>4.1.7. Переопределение методов Equals и GetHashCode в С#</p> <p>4.1.8. Регулярные выражения в С#. Класс Regex</p> <p>Лабораторные занятия</p> <p>1. Разработка приложения с использованием собственных классов</p> <p>2. Разработка приложения с использованием классов "наследования"</p> <p>3. Проектирование приложения по индивидуальным заданиям (с использованием принципов ООП)</p> <p>4. Разработка приложения по индивидуальным заданиям (с использованием принципов ООП)</p>		

	<p>Самостоятельная работа Подготовка к лабораторным занятиям, оформление отчетов по выполненным работам. Работа над материалом учебников [1], [2], [3], конспектом лекций, решение задач по образцу.образцу.</p>		
5. Работа с базами данных на языке С#		72	
5.1. Работа с таблицами данных программирования	Содержание учебного материала		
	5.1.1. Объекты: DataSet, DataTable и DataColumn		
	5.1.2. Объекты: DataRow, DataGridView, DataView		
	5.1.3. Использование объектов работы с таблицами при создании интерфейса приложения БД		
	<p>Лабораторные занятия 1. Разработка приложения с использованием объектов DataSet, DataTable и DataColumn 2. Отладка и тестирование приложения с использованием объектов DataSet, DataTable и DataColumn 3. Разработка приложения с использованием объектов DataRow, DataGridView, DataView 4. Отладка и тестирование приложения с использованием объектов DataRow, DataGridView, DataView</p>		
	<p>Самостоятельная работа Подготовка к лабораторным занятиям, оформление отчетов по выполненным работам. Работа над материалом учебников [1], [4]конспектом лекций, решение задач по образцу.</p>		
5.2. Создание приложений баз данных	Содержание учебного материала		
	5.2.1. Пример простейшего приложения баз данных. Обзор объектов ADO .NET		
	5.2.2. Источник данных DataSet. Таблицы и поля (объекты DataTable и DataColumn). Объекты DataRelation. Строки (объект DataRow). DataAdapter. Объекты DBConnection и DBCommand		

	5.2.3. Server Explorer. Разбор примера создания приложения БД «вручную»		
	5.2.4. Соединение с базой данных. Командная строка соединения SqlConnection. Управление соединением. Объект Connection. Работа с пулом соединений.		
	5.2.5. Хранимые процедуры. Запросы к базе данных		
	5.2.6. Работа с таблицами данных		
	Обобщение теоретического материала. Подведение итогов		
	Лабораторные занятия 1. Проектирование приложения БД (по индивидуальным заданиям) 2. Разработка приложения БД (по индивидуальным заданиям) 3. Тестирование и отладка приложения БД (по индивидуальным заданиям) 4. Обработка исключений при работе с MS SQL 5. Выполнение соединения с базой данных 6. Создание хранимых процедур в Visual Studio Программное создание объектов DataTable и DataColumn 7. Программное создание и изменение записей таблицы данных. 8. Вывод двух связанных таблиц данных в один элемент DataGridView 9. Вывод связанных таблиц данных в два элемента DataGridView 10. Документирование и подготовка презентации по проекту		
	Самостоятельная работа Подготовка к лабораторным занятиям, оформление отчетов по выполненным работам; Работа над материалом учебников [1], [4], конспектом лекций.		
ВСЕГО:		361	

Для характеристики уровня освоения учебного материала используются следующие обозначения:

1 – ознакомительный (узнавание ранее изученных объектов, свойств);

2 – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством)

3 – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация учебной дисциплины требует наличия кабинета программирования и баз данных, лаборатории инструментальных средств разработки.

Технические средства обучения: ПК, проектор, интерактивная доска

Оборудование лаборатории и рабочих мест лаборатории:

- посадочные места студентов;
- рабочее место преподавателя;
- рабочая доска;
- наглядные пособия (учебники, терминологические словари разных типов, опорные конспекты-плакаты, стенды, карточки, раздаточный материал, комплекты практических работ).

3.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники:

1. Основы алгоритмизации и программирования: Учебное пособие / Колдаев В.Д; Под ред. проф.Л.Г. Гагариной - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2016. - 416 с.: 60x90 1/16. - (Профессиональное образование) (Переплёт 7БЦ) ISBN 978-5-8199-0279-0. Режим доступа: <http://znanium.com/bookread2.php?book=537513>

2. Алгоритмизация и программирование : Учебное пособие / С.А. Канцедал. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014. - 352 с.: ил.; 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-8199-0355-1, 500 экз. Режим доступа: <http://znanium.com/catalog.php?bookinfo=429576>

3. Евдокимов, П.В. С# на примерах [Электронный ресурс] — Электрон. дан. — Санкт-Петербург : Наука и Техника, 2016. — 304 с. Режим доступа: <https://e.lanbook.com/book/90233>

Дополнительные

1. Тюкачев, Н.А. С#. Алгоритмы и структуры данных [Электронный ресурс] / Н.А. Тюкачев, В.Г. Хлебостроев. — Электрон. дан. — Санкт-Петербург : Лань, 2017. — 232 с. Режим доступа: <https://e.lanbook.com/book/94748>

2. Тюкачев, Н.А. С#. Основы программирования [Электронный ресурс] / Н.А. Тюкачев, В.Г. Хлебостроев. — Электрон. дан. — Санкт-Петербург : Лань, 2017. — 272 с. Режим доступа: <https://e.lanbook.com/book/94749>

3. Задачи по программированию [Электронный ресурс] : учеб. пособие / С.М. Окулов [и др.]. — Электрон. дан. — Москва : Издательство "Лаборатория знаний", 2017. — 826 с. Режим доступа: <https://e.lanbook.com/book/94162>

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения учебной дисциплины осуществляется преподавателем в процессе проведения лабораторных работ, тестирования, а также выполнения обучающимися индивидуальных заданий, проектов, исследований.

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
Умения:	
Работа в среде программирования.	Комбинированный: лабораторные практикумы, тестирование работа поиск информации на профессиональных сайтах. Индивидуальный и групповой: проектная (исследовательская работа).
Реализация построенных алгоритмов в виде программ на конкретном языке программирования.	Комбинированный: практические и лабораторные занятия, тестирование на профессиональных сайтах, участие в олимпиадах. Индивидуальный и групповой: проектная (исследовательская работа).
Знания:	
Этапы решения задачи на компьютере.	Комбинированный: тестирование, устный опрос, лабораторные практикумы.
Типы данных.	Комбинированный: тестирование, устный опрос, лабораторные практикумы.

Базовые конструкции изучаемых языков программирования.	Комбинированный: тестирование, устный опрос, лабораторные практикумы.
Принципы структурного и модульного программирования.	Комбинированный: тестирование, устный опрос, лабораторные практикумы.
Принципы объектно-ориентированного программирования	Комбинированный: тестирование, устный опрос, лабораторные практикумы.
	Индивидуальный: экзамен

4.1. ПЕРЕЧЕНЬ ОСНОВНЫХ ВОПРОСОВ, ВЫНОСИМЫХ НА ЭКЗАМЕН ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Раздел 1. Основы алгоритмизации

1. Алгоритм и его свойства.
2. Способы представления алгоритмов: словесный, формульно-словесный, блок-схемный.
3. Базовые канонические структуры, используемые при проектировании алгоритмов (линейные, ветвление, циклы).
4. Базовые алгоритмы заполнения и обработки массивов (поиск по критерию, сумма/произведение/количество, сортировки и т.п.)

Раздел 2. Этапы решения задач на ПК. Базовые средства языка C#

5. Основные этапы решения задач на ПК. Классификация языков программирования.
6. Платформа Microsoft .Net Framework, Visual Studio, язык C#
7. Консольные и Windows-приложения C#.
8. Арифметические, строковые и логические выражения в языке C#. Приоритеты выполнения операций в выражениях.
9. Операторы языка C#.(Операторы выбора. Операторы break, continue. Операторы цикла.)
10. Массивы в C#.

11. Реализация базовых алгоритмы заполнения и обработки массивов (поиск по критерию, сумма/произведение/количество, сортировки и т.п.) в С#.

Раздел 3. Создание пользовательского интерфейса в Visual C#

12. Форма и элементы управления С#. Задание свойств. Обработка событий.

13. Использование элемента управления при проектировании пользовательского интерфейса (TextBox, Button, RadioButton, CheckBox, MessageBox, PictureBox, MenuStrip, ToolStrip, Timer и т.п.)

14. Массив из объектов (элементов управления). Создание, удаление, обработка.

15. Списки и деревья (ListView и Tree View)

16. Файловый ввод-вывод. Создание и удаление с диска.

17. Меню. Создание головного и вложенного меню. Обработка сообщений меню. Контекстное меню. Многооконные приложения.

18. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя. Программирование приложения. Тестирование, отладка приложения. Создание документации.

Раздел 4. Объектно-ориентированное программирование в С#

19. Объектно-ориентированное программирование. Классы и объекты. Объявление классов и создание объектов. Методы. Конструкторы.

20. Инкапсуляция. Наследование. Полиморфизм. Конструкторы и деструкторы.

Раздел 5. Работа с базами данных на языке С#

21. Объекты: DataSet, DataTable, DataColumn, DataRow, DataGridView, DataView

22. Использование объектов работы с таблицами при создании интерфейса приложения БД

23. Соединение с базой данных. Командная строка соединения ConnectionString. Управление соединением.

24. Хранимые процедуры. Запросы к базе данных

25. Разработка функциональной схемы работы приложения.

4.2. КРИТЕРИИ ОЦЕНКИ ОТВЕТОВ

При оценке ответов дополнительно должны быть учтены качество сообщения, отражающего основные моменты и ответы на вопросы, заданные по теме вопроса.

Результаты защиты определяются оценками *«отлично»*, *«хорошо»*, *«удовлетворительно»*, *«неудовлетворительно»*.

Оценки *«отлично»* заслуживает ответ, в котором полно и всесторонне раскрыто теоретическое содержание темы, дан глубокий критический анализ действующей практики учетно-аналитической работы. Студент при ответе дал аргументированные ответы на все вопросы преподавателя, проявил творческие способности в понимании и изложении ответов на вопросы.

Оценка *«хорошо»* выставляется за ответ, который имеет убедительный ответ. При его этом студент показывает знания вопросов темы, оперирует данными, вносит предложения по теме ответа, во время ответа использует наглядные пособия, без особых затруднений отвечает на поставленные вопросы.

Оценка *«удовлетворительно»* выставляется за ответ, в котором имеются замечания по содержанию ответа и методике анализа. В теоретических, выводы в основном правильные, предложения представляют интерес, но недостаточно убедительно аргументированы и не на все вопросы студент дал правильные ответы.

Оценка *«неудовлетворительно»* выставляется за ответ, который в основном отвечает предъявляемым вопросам, но студент не дал правильных ответов на большинство заданных вопросов, т.е. обнаружил серьезные пробелы в профессиональных знаниях.