

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**Колледж**

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**ОП.04 Основы алгоритмизации и программирования**

по программе подготовки специалистов среднего звена (ППССЗ) среднего  
профессионального образования

Специальность:	10.02.05. Обеспечение информационной безопасности автоматизированных систем.
Обучение:	по программе базовой подготовки
Уровень образования, на базе которого осваивается ППССЗ:	Основное общее образование
Квалификация:	Техник по защите информации
Форма обучения:	Очная

Рабочая программа дисциплины: «Основы алгоритмизации и программирования» разработана на основе требований Федерального государственного образовательного стандарта (далее ФГОС) для среднего профессионального образования (СПО) по специальности 10.02.05. «Обеспечение информационной безопасности автоматизированных систем» от 09.12.2016 №1553 для реализации основной профессиональной образовательной программы СПО на базе основного общего образования с получением среднего общего образования.

**Организация-разработчик:** колледж федерального государственного бюджетного образовательного учреждение высшего образования «Дагестанский государственный университет» (Колледж ДГУ)

**Разработчики:**

Сайгитмагомедова Хадижат Садрутдиновна - преподаватель кафедры общепрофессиональных дисциплин Колледжа ДГУ

Рабочая программа дисциплины одобрена на заседании кафедры общепрофессиональных дисциплин колледжа ДГУ

протокол № 7 от «1» марта 2021г.

Зав. кафедрой Шериф - Магомедова П.Р.

Рабочая программа дисциплины согласована с учебно-методическим управлением «И» 03 2021г.

Начальник УМУ, д.б.н., проф. А.Г. Гасангаджиева А.Г

## **Содержание**

<b>1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ: «Основы алгоритмизации и программирования» .....</b>	<b>4</b>
<b>2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ .....</b>	<b>6</b>
<b>3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ .....</b>	<b>18</b>
<b>4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....</b>	<b>19</b>

# 1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

## 1.1. Область применения программы

Рабочая программа дисциплины является частью основной профессиональной образовательной программы в соответствии с ФГОС по специальности СПО 10.02.05.

Обеспечение информационной безопасности автоматизированных систем, для очного обучения студентов, имеющих основное общее образование, по программе базовой подготовки

## 1.2. Место дисциплины в структуре основной профессиональной образовательной программы:

Учебная дисциплина «Основы алгоритмизации и программирования» относится к общепрофессиональным дисциплинам профессионального цикла ППСЗ.

## 1.3. Цели и задачи дисциплины – требования к результатам освоения дисциплины:

Содержание программы учебной дисциплины «Основы алгоритмизации и программирования» направлено на достижение следующей цели:

**Целью дисциплины «Основы алгоритмизации и программирования»** является изучение и применение на практике основных методов и средств обработки, хранения, передачи и поиска информации с использованием различных программных средств,

**Задачи учебной дисциплины «Основы алгоритмизации и программирования» в профессиональной деятельности:**

1 Понимать сущность и социальную значимость своей будущей профессии, обладать высокой мотивацией к выполнению профессиональной деятельности в области обеспечения информационной безопасности

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

3 Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития

4 Использовать информационно-коммуникационные технологии в профессиональной деятельности

5 Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации

6 Ориентироваться в условиях частой смены технологий в профессиональной деятельности

7 Владеть основными методами и средствами разработки программного обеспечения

8 Решать частные технические задачи, возникающие при проведении всех видов плановых и внеплановых контрольных проверок, при аттестации объектов, помещений, программ, алгоритмов

В результате изучения дисциплины: «Основы алгоритмизации и программирования» в профессиональной деятельности обучающийся должен обладать следующими компетенциями:

Код	Наименование результата обучения
ОК-01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК-02	Осуществлять поиск, анализ и интерпретацию информации, необходимой

	для выполнения задач профессиональной деятельности
ОК-03	Планировать и реализовывать собственное профессиональное и личностное развитие.
ПК-2.1	Осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации.
ПК-2.2	Обеспечивать защиту информации в автоматизированных системах отдельными программными, программно-аппаратными средствами.
ПК-2.3	Осуществлять тестирование функций отдельных программных и программно-аппаратных средств защиты информации.
ПК-2.4	Осуществлять обработку, хранение и передачу информации ограниченного доступа
ПК-2.6	Осуществлять регистрацию основных событий в автоматизированных (информационных) системах, в том числе с использованием программных и программно-аппаратных средств обнаружения, предупреждения и ликвидации последствий компьютерных атак.

В результате освоения дисциплины обучающийся должен знать:

- - типы данных, базовые конструкции изучаемых языков программирования;
- интегрированные среды изучаемых языков программирования.

В результате освоения дисциплины обучающийся должен уметь:

- использовать языки программирования высокого уровня;
- работать в интегрированной среде программирования.

При реализации содержания учебной дисциплины «Основы алгоритмизации и программирования» в пределах освоения ОПОП СПО на базе основного общего образования с получением среднего общего образования учебная нагрузка студентов составляет 196 часов, из них аудиторная (обязательная) учебная нагрузка, включая практические занятия, — 154 часов; внеаудиторная самостоятельная работа студентов — 41 часов

## 2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 2.1. Объем учебной дисциплины и виды учебной работы

<b>Вид учебной работы</b>	<b>Объем часов</b>
<b>Максимальная учебная нагрузка (всего)</b>	<i>196</i>
<b>Обязательная аудиторная учебная нагрузка (всего)</b>	<i>154</i>
в том числе:	
Лекции	<i>78</i>
практические занятия	<i>76</i>
контрольные работы	-
курсовая работа (проект) ( <i>если предусмотрено</i> )	-
<b>Самостоятельная работа обучающегося (всего)</b>	<i>42</i>
<i>промежуточная аттестация в форме: ДФК, экзамена</i>	

## 2.2. Тематический план и содержание дисциплины

2	Содержание учебного материала, лабораторные работы, практические занятия, самостоятельная работа обучающихся	Объем часов
1	2	3
<b>Раздел 1. Основные принципы алгоритмизации и программирования</b>		
<b>Тема 1.1.</b> Алгоритм и его свойства	<i><b>Лекционные занятия:</b></i> Введение. Понятие алгоритма; свойства алгоритма; способы описания алгоритма (словесный, формульно-словесный, графический). Понятие алгоритмического языка. Алгоритмизация как базовая составляющая технологического процесса создания программного изделия	<b>4</b>
	<b>Практическое занятие.</b> Общие принципы построения алгоритмов. Основные принципы структурной методологии: принцип абстракции, принцип формальности, принцип «разделяй и властвуй», принцип иерархического упорядочения. Построение алгоритмов в словесной форме.	4
	<i><b>Самостоятельная работа:</b></i> - домашнее задание на построение алгоритмов в словесной форме	4

Тема 1.2 Базовые алгоритмические конструкции	<i>Лекционные занятия:</i> Блок-схемы алгоритмов различной структуры (линейной, разветвляющейся, циклической) Понятие ветвления. Понятие алгоритмического цикла. Типы циклов	4
	<i>Практическое занятие:</i> Примеры алгоритмов различной структуры. Построение блок-схемы алгоритмов	4
	<i>Самостоятельная работа:</i> - домашние практические работы на построение алгоритмов решения простейших задач	4
Тема 1.3. Логические основы алгоритмизации	<i>Лекционные занятия:</i> Позиционные и непозиционные системы счисления. Алгоритмы перевода чисел из одной системы счисления в другие. Перевод чисел в системах счисления с кратными основаниями. Арифметические действия в двоичной системе счисления. Прямой код, обратный (инверсный) код, дополнительный код	4
	<i>Практическое занятие:</i> 1. Позиционные и непозиционные системы счисления. Двоичная, восьмеричная, шестнадцатеричная системы счисления. Алгоритмы перевода чисел из одной системы счисления в другие Перевод чисел в системах счисления с кратными основаниями. Арифметические действия в двоичной системе счисления. Прямой код, обратный (инверсный) код, дополнительный код	4
	Самостоятельная работа. Прямой код, обратный (инверсный) код, дополнительный код	4
<b>Раздел 2. Язык программирования</b>		

Тема 2.1. Языки и системы программирования	<p><b>Лекционное занятие.</b> Развитие языков программирования. Обзор языков программирования. Объекты применения языков программирования. Стандарты языков программирования.</p>	4
	<p><b>Практические занятия:</b> История развития языков программирования. Сравнительный обзор высокоуровневых языков программирования и общие понятия высокоуровневых языков программирования. Движущие силы развития языков программирования. Классификация языков программирования.</p>	2
	<p><b>Самостоятельная работа.</b> - подготовка сообщений, рефератов по теме занятия. Примерная тематика:</p> <ul style="list-style-type: none"> <li>– эволюция языков программирования;</li> <li>– машинно-ориентированные языки;</li> <li>– процедурные языки программирования;</li> <li>– логические языки программирования;</li> <li>– интерпретаторы и компиляторы;</li> <li>– интегрированные среды;</li> <li>– визуальные среды программирования;</li> <li>– встроенные языки программирования;</li> <li>– технология RAD;</li> </ul> <p>CASE-технологии</p>	2
Тема 2.2. Методы и принципы программирования	<p><b>Лекционное занятие.</b> Методы программирования, виды программного обеспечения, жизненный цикл программного обеспечения, элементы языка программирования</p>	2

	<p><b>Практические занятия:</b></p> <ol style="list-style-type: none"> <li>1. Понятие программного обеспечения</li> <li>2. Методы программного обеспечения, принципы программного обеспечения,</li> <li>3. Жизненный цикл программного обеспечения, специализация языка программирования, Элементы языка программирования, структура языка программирования</li> </ol>	<b>2</b>
	<p><b>Самостоятельная работа:</b></p> <p>- выполнение индивидуальных заданий для закрепления навыков программирования по темам практических занятий</p>	<b>2</b>
<p><b>Тема 2.3. Типы данных</b></p>	<p><b>Лекционные занятия:</b> Данные. Входные, выходные и промежуточные данные. Постоянные (константы) и переменные данные. Идентификация переменных. Понятие типа данных. Скалярные типы данных: целочисленные, вещественные, литерные, логические. Значения переменных. Область допустимых значений переменных. Допустимые операции</p>	<b>2</b>
	<p><b>Практические занятия:</b></p> <ol style="list-style-type: none"> <li>1. <b>Типы данных определяемые программистом. Перечисляемые и интервальные типы.</b></li> <li>2. Выражения. Арифметические, логические и строковые выражения. Запись выражений в процедурных алгоритмических языках. Приоритеты операций в выражениях. Присваивание значений</li> <li>3. Построение алгоритмов решения задач с использованием различных типов данных</li> </ol>	<b>2</b>

	<b>Самостоятельная работа обучающихся:</b> - домашние практические работы на построение алгоритмов решения простейших задач с использованием данных различных типов	<b>2</b>
<b>Раздел 3 Основы программирования в интегрированной среде</b>		
<b>Тема 3.1</b> Знакомство с Python.	<b>Лекционное занятие</b> Краткая историческая справка и основные особенности языка. Дзен Питона. Режимы работы с интерпретатором. Создание скриптов. Схема запуска программ. Основные элементы.	<b>2</b>
	<b>Практические занятия:</b> Язык программирования Python. Знакомство и первая работа в среде разработки IDLE	<b>4</b>
	<b>Самостоятельная работа.</b> Подготовка реферата	<b>4</b>
<b>Тема 3.2</b> Типы данных. Ввод и вывод данных. Логические выражения и операторы	<b>Лекционные занятия:</b> Данные и их типы. Операции в программировании. Изменение типов данных. Логические выражения и логический тип данных. Логические операторы. Сложные логические выражения.	<b>4</b>
	<b>Практические занятия:</b> Данные и их типы. Операции в программировании. Изменение типов данных. Переменные. Обозначение переменных. Арифметические операторы.	<b>4</b>
	<b>Самостоятельная работа:</b> Данные и их типы. Операции в программировании. Изменение типов данных. Переменные. Вывод данных. Функция print(). Ввод данных. Функция input().	<b>4</b>

<b>Тема 3.3.</b> Линейные алгоритмы и программы	<b>Лекционные занятия:</b> Понятие линейных алгоритмов. Построение линейных алгоритмов	<b>2</b>
	<b>Практические занятия:</b> Программирование алгоритмов линейной структуры	<b>2</b>
	<b>Самостоятельная работа:</b> Программирование алгоритмов линейной структуры	<b>2</b>
<b>Тема 3.4</b> Ветвление. Ветвления и операторы выбора	<b>Лекционные занятия:</b> Понятие ветвления в языках программирования. Организация ветвлений в Python. Организация множественных ветвлений. Синтаксис if-elif-else. Вложенные условные конструкции. Каскадные условные конструкции.	<b>4</b>
	<b>Практические занятия:</b> Инструкция if – elif – else. Использование инструкции if – elif – else. Написание программ с использованием инструкции if – elif – else для определения	<b>4</b>
	<b>Самостоятельная работа:</b> Организация ветвлений в Python. Организация множественных ветвлений. Синтаксис if-elif-else. Вложенные условные конструкции. Каскадные условные конструкции.	<b>4</b>
<b>Раздел 4. Цикл в языке программирования Python</b>		

<b>Тема 4. 1</b> Циклические операторы For	<b>Лекционные занятия:</b> Цикл for. Требования к записи цикла. Работа цикла. Порядок выполнения программа	<b>4</b>
	<b>Практические занятия:</b> Решение задач с циклом for	<b>4</b>
	<b>Самостоятельная работа:</b> Рассмотрение циклических операторов.	<b>4</b>
<b>Тема 4. 2</b> Циклы в программировании. Цикл while	<b>Лекционные занятия:</b> Знакомство с организацией циклов в Python. Знакомство и применение цикла while.	<b>4</b>
	<b>Практические занятия:</b> «Решение задач с циклом while»	<b>4</b>
<b>Тема 4.3</b> Операторы break и continue.	<b>Лекционные занятия:</b> Знакомство и применение цикла for, функции range, операторов break и continue	<b>4</b>
	<b>Практические занятия:</b> Оператор прерывания цикла – break. Оператор перехода к следующему шагу цикла – continue. Синтаксис записи программы.	<b>4</b>
<b>Тема 4.4</b> Функции в программировании	<b>Лекционные занятия:</b> Понятие функции. Определение функции. Оператор def. Вызов функции. Структура программ с функциями.	<b>6</b>
	<b>Практические занятия:</b> Оператор def. Вызов функции. Структура программ с функциями	<b>6</b>

	<i><b>Самостоятельная работа:</b></i> выполнение домашних заданий	<b>2</b>
<b>Тема 4.5</b> Локальные и глобальные переменные	<i><b>Лекционные занятия:</b></i> Понятие локальные и глобальные переменные. Работа с локальными и глобальными переменными в Python.	<b>4</b>
	<i><b>Практические занятия:</b></i> Работа с локальными и глобальными переменными в Python.	<b>4</b>
<b>Тема 4.6</b> Возврат значений из функций. Оператор return	<i><b>Лекционные занятия:</b></i> Знакомство с функцией return. Возврат нескольких значений.	<b>4</b>
	<i><b>Практические занятия:</b></i> Знакомство с функцией return. Возврат нескольких значений.	<b>4</b>
<b>Раздел 5. Функции в Python</b>		
<b>Тема 5.1</b> Параметры и аргументы функций	<i><b>Лекционные занятия:</b></i> Понятие параметры в Python. Произвольное количество аргументов. Вызов функции. Обязательные аргументы функций. Аргументы заданные по умолчанию. Аргументы произвольной длины.	<b>2</b>
	<i><b>Практические занятия:</b></i> Понятие параметры в Python. Произвольное количество аргументов.	<b>2</b>
	<i><b>Самостоятельная работа:</b></i> выполнение домашних заданий	<b>2</b>
<b>Тема 5.2</b> Встроенные функции	<i><b>Лекционные занятия:</b></i> Понятие встроенные функции Python. Работа со встроенными функциями Python.	<b>2</b>

	<b>Практические занятия:</b> Понятие встроенные функции Python. Работа со встроенными функциями Python.	<b>2</b>
<b>Раздел 6. Коллекции в Python.</b>		
<b>Тема 6.1</b> Матрицы и множества	<b>Лекционные занятия:</b> Работа с матрицами. Работа с множествами.	<b>4</b>
	<b>Практические занятия:</b> Работа с матрицами. Работа со множествами.	<b>4</b>
	<b>Самостоятельная работа:</b> выполнение домашних заданий	<b>2</b>
<b>Тема 6.2</b> Списки	<b>Лекционные занятия:</b> Понятие списки в Python. Создание и работа со списками. Двумерные списки. Обработка и вывод списка. Сортировка списка.	<b>2</b>
	<b>Практические занятия:</b> Работа со списками. Операции над списками в Python.	<b>2</b>
<b>Тема 6.3</b> Строки	<b>Лекционные занятия:</b> Знакомство со строками в Python. Методы строк. Форматирование строк. Функции строк.	<b>2</b>
	<b>Практические занятия:</b> Знакомство со строками в Python. Методы строк	<b>2</b>
<b>Тема 6.4</b> Кортежи	<b>Лекционные занятия:</b> Знакомство с кортежами. Списки в кортежах	<b>2</b>

	<b>Практические занятия:</b> Знакомство с кортежами. Списки в кортежах. Решение задач.	<b>2</b>
<b>Тема 6.5</b> Файлы	<b>Лекционные занятия:</b> Знакомство с файлами. Открытие файла. Закрытие файла. Чтение и запись файлов. Переименование файлов. Позиция файлов. Методы файла.	<b>2</b>
		<b>1</b>
<b>Тема 6.6</b> Словари	<b>Лекционные занятия:</b> Знакомство со словарями. Перебор элементов словаря в цикле for. Методы словаря. Создание словаря.	<b>2</b>
	<b>Практические занятия:</b> Знакомство со словарями. Перебор элементов словаря в цикле for. Методы словаря	<b>1</b>
<b>Раздел 7. Библиотеки в Python.</b>		
<b>Тема 7.1</b> Модули	<b>Лекционные занятия:</b> Понятие модуль в Python. Подключение модулей в Python. Создание собственного модуля. Библиотека в программирование. Особенности импорта модулей. Встроенные модули.	<b>2</b>
	<b>Практические занятия:</b> Понятие модуль в Python. Подключение модулей в Python. Создание собственного модуля.	<b>2</b>

<b>Тема 7.2</b> Библиотеки python. Работа с изображениями	<i><b>Лекционные занятия:</b></i> Обработка изображений. Установка библиотек. Фильтры.Изменение размера изображения. Создание изображений в рисовании.	<b>2</b>
	<i><b>Практические занятия:</b></i> Разработка программы с изменением размера изображения, с применением фильтра.	<b>2</b>
<i><b>Лекционные занятия</b></i>		<b>78</b>
<i><b>Практические занятия</b></i>		<b>76</b>
<i><b>Самостоятельная работа</b></i>		<b>42</b>
<i><b>Итого</b></i>		<b>196</b>
<i><b>Форма контроля</b></i>		<b>ДФК, Экзамен</b>

## 4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

### 3.1. Требования к минимальному материально-техническому обеспечению

Реализация программы дисциплины требует наличия учебного кабинета. Для усвоения знаний и практических навыков студентами изучение дисциплины «Основы алгоритмизации и программирования» обеспечено, прежде всего, наличием научно-учебно-методического кабинета, в котором есть возможность проводить занятия, как в традиционной форме, так и с использованием интерактивных технологий и различных образовательных методик. Имеется библиотека, включающая литературу, как основного, так и дополнительного характера.

#### Оборудование учебного кабинета:

- посадочные места по количеству обучающихся;
- рабочее место преподавателя;
- комплект учебно-наглядных пособий;

#### Технические средства обучения:

- компьютер по количеству обучающихся с лицензионным программным обеспечением и с доступом к сети Интернет.
- проектор;
- интерактивная доска;

#### Программное обеспечение:

- интегрированная среда программирования (Python? Microsoft Visual Studio, и т.п.),

### 3.2. Информационное обеспечение обучения

#### Перечень рекомендуемых учебных изданий, Интернет-ресурсов,

##### а) основная литература:

- 1 Кудрина, Е. В. Основы алгоритмизации и программирования на языке С# : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2021. — 322 с.
- 2 Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — Москва : Издательство Юрайт, 2021. — 219 с.

##### б) дополнительная литература:

- 1 1. Зыков, С.В. Введение в теорию программирования. Курс лекций: учебное пособие / С.В. Зыков. – М.: ИНТУИТ.РУ 2004.
- 2 Керман. Программирование и отладка в Delphi. / Керман, пер.с англ. – М.: Изд. Дом Вильямс, 2003.
- 3 Любавин, С.А. Turbo Delphi для новичков и не только / С.А. Любавин. – М.:ИТ Пресс, 2008
- 4 Марченко, А.И. Программирование в среде Turbo Pascal 7.0./ А.И. Марченко, Л.А. Марченко. – Киев: ВЕК+, 1999 г.
- 5 Меняев. М.Ф. Информатика и основы программирования: учеб. пособие/ М.Ф. Меняев, – М.:Омега-Л, 2005.
- 6 Острейковский, В.А. Информатика. / В.А. Острейковский, – М.: Высш.шк. 1999.
- 7 Павловская, Т.А. Паскаль. Программирование на языке высокого уровня: учебник для вузов. – СПб: Питер, 2008
- 8 Семакин, И.Г. Основы программирования. / И.Г. Семакин, А.П. Шестаков. – М.: Академия, 2006.
- 9 Хорев, П.Б. Технологии объектно-ориентированного программирования: учеб.пособие для студ. высш. учеб.заведений / П.Б. Хорев. – М.: Академия, 2004.

### Средства обеспечения освоения дисциплины

1. Электронный каталог НБ ДГУ [Электронный ресурс]: база данных содержит сведения о всех видах литературы, поступающих в фонд НБ ДГУ / Дагестанский государственный университет. – Махачкала, 2010 – Режим доступа: <http://elib.dgu.ru>, свободный
2. Moodle [Электронный ресурс]: система виртуального обучения: [база данных] / Даг. Гос. ун-т. – Махачкала, г. – Доступ из сети ДГУ или, после регистрации из сети ун-та, из любой точки, имеющей доступ в интернет. –URL: <http://moodle.dgu.ru/>
3. Электронный каталог НБ ДГУ [Электронный ресурс]: база данных содержит сведения о всех видах литературы, поступающих в фонд НБ ДГУ / Дагестанский гос. ун-т. – Махачкала, 2010 – режим доступа: <http://elib.dgu.ru>, свободный
- 4/ Большой юридический справочник -[http://www.blblaw.ru/kodeksy\\_rf.html](http://www.blblaw.ru/kodeksy_rf.html)
5. Библиотека бесплатных программ [Электронный ресурс]: – Режим доступа: <http://biblprog.org.ua/ru>
6. Сайт «Все для программиста» [Электронный ресурс]: – Режим доступа: <http://www.codenet.ru/>

### Материально-техническое обеспечение дисциплин

1. Операционные системы Windows 7, SuseLinux 10.
2. Программы настройки и оптимизации операционной системы: NortonUtilities, SystemUtilities, Sandra, SystemMechanic.
3. Архиваторы: WinRar, WinZip, ZipMagic, WinAce.
4. Антивирусные средства: Drweb, Avp.
5. Программы для работы с изображением: ACDSee, AcrobatReader.
6. Программы для работы с Internet и электронной почтой: EtypeDialer, GetRight, TheBat!, Ace FTP, Opera, ICQ. 10
7. Программы-оболочки: FAR manager 2.0, Volkov Commander.
8. MS Office 2007/2010.
9. СУБД ACCESS.
10. СерверБД MS SQL 2005.
11. Программы для тестирования аппаратных устройств ПЭВМ.
12. Текстовые процессоры / редакторы: Word, LaTeX.
13. Программы для создания компьютерных презентаций MS PowerPoint

### 3. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий и 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
<b>В результате освоения дисциплины обучающийся должен знать:</b>	

<p>- Интегрированные среды изучаемых языков программирования обнаруживать и устранять неисправности системы защиты информации автоматизированной системы согласно документации (06.033 А/01.5) типы данных, базовые конструкции изучаемых языков программирования; интегрированные среды изучаемых языков программирования</p>	<p>- Оценка в рамках текущего контроля результатов деятельности обучающихся при выполнении:  - проверка конспектов лекций;  - тестирование;  - самостоятельная работа;  - выполнение индивидуальных практических заданий;  - индивидуальный и фронтальный опрос в ходе аудиторных занятий;  - защита практической работы</p>
<p><b>В результате освоения дисциплины обучающийся должен уметь:</b></p>	
<p>- использовать языки программирования высокого уровня;  - работать в интегрированной системе программирования;</p>	<p>- Оценка в рамках текущего контроля результатов деятельности обучающихся при выполнении:  - проверка конспектов лекций;  - тестирование;  - самостоятельная работа;  - выполнение индивидуальных Практических/лабораторных заданий;  - индивидуальный и фронтальный опрос в ходе аудиторных занятий;  - защита практической работы</p>