

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Узунова Галина Петровна

Должность: Директор

Дата подписания: 19.09.2023 10:21:58

Уникальный программный ключ:

ec29c88afcd483fc3f14efec2359d2c1514e1daf0b74e9391ec46ce98af9ce5f

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
«ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ»
«ОТКРЫТЫЙ ТАВРИЧЕСКИЙ КОЛЛЕДЖ»

УТВЕРЖДАЮ

Директор

АНО «ПОО» «Открытый
Таврический колледж»

Г.П. Узунова
2022 г.



РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

ПО СПЕЦИАЛЬНОСТИ

09.02.07 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

БАЗОВЫЙ УРОВЕНЬ ПОДГОТОВКИ

(базовый, углубленный)

Программист

(квалификация)

ФОРМА ОБУЧЕНИЯ

ОЧНАЯ

Симферополь, 2022 г.

РАССМОТРЕНА и ОДОБРЕНА
на заседании цикловой комиссии
Протокол № 1
от «07» июня 2022 г.
Председатель цикловой комиссии
Ришма А. А. Р. Р.
(Подпись, Ф.И.О.)

Разработана на основе Федерального
государственного образовательного
стандарта по специальности 09.02.07
Информационные системы и
программирование

СОДЕРЖАНИЕ

стр.

1. ПАСПОРТ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	6
3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ	14
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	16

1. ПАСПОРТ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

«ОП.04 Основы алгоритмизации и программирования»

1.1. Область применения программы

Рабочая программа учебной дисциплины является частью программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности СПО

09.02.07 Информационные системы и программирование.

1.2. Место дисциплины в структуре программы подготовки специалистов среднего звена

Дисциплина «Основы алгоритмизации и программирования» входит в цикл общепрофессиональных дисциплин.

1.3. Цель и планируемые результаты освоения дисциплины:

В результате освоения дисциплины обучающийся должен уметь:

У1 разрабатывать алгоритмы для конкретных задач.

У2 использовать программы для графического отображения алгоритмов.

У3 определять сложность работы алгоритмов.

У4 работать в среде программирования.

У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

У6 оформлять код программы в соответствии со стандартом кодирования.

У7 выполнять проверку, отладку кода программы.

В результате освоения дисциплины обучающийся должен знать:

З1 понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.

З2 эволюцию языков программирования, их классификацию, понятие системы программирования.

З3 основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

З4 подпрограммы, составление библиотек подпрограмм.

35 объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

В результате освоения дисциплины обучающийся осваивает элементы компетенций:

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

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

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

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

ОК 9 Использовать информационные технологии в профессиональной деятельности

ОК 10 Пользоваться профессиональной документацией на государственном и иностранном языках.

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

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

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

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

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

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

1.4. Рекомендуемое количество часов на освоение программы дисциплины:

максимальной учебной нагрузки обучающегося 158 часов, в том числе:
обязательной аудиторной учебной нагрузки обучающихся 154 часа;
самостоятельной работы обучающихся 4 часа.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем в часах
Объем образовательной программы	<i>158</i>
в том числе:	
теоретическое обучение	<i>78</i>
практические занятия	<i>76</i>
Самостоятельная работа	<i>4</i>
<i>Промежуточная аттестация в форме дифференцированного зачета</i>	

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

Наименование разделов и тем	Содержание учебного материала, лабораторные и практические работы, самостоятельная работа обучающихся	Объем часов	Коды компетенции, формированию которых способствует элемент программы
1	2	3	4
Раздел 1. Введение в программирование			
Тема 1.1. Языки программирования	Содержание учебного материала	10	OK 10
	<i>31 понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.</i>		
	<i>32 эволюцию языков программирования, их классификацию, понятие системы программирования.</i>		
	<i>33 основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</i>		
	1 Развитие языков программирования. Обзор языков программирования. Области применения языков программирования.		
	2 Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.		
	3 Жизненный цикл программы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере.		
Лабораторные занятия	-		
Практические занятия	-		
Тема 1.2. Типы данных	Содержание учебного материала	4	OK 5
	<i>31 понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.</i>		
	<i>33 основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</i>		
	1 Переменные и константы. Объявление объектов данных. Внутренне представление данных в памяти компьютера		
	2 Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.		
	Лабораторные занятия	-	
	Практические занятия	-	
Раздел 2. Основные конструкции языков программирования			

Тема 2.1. Операторы языка программирования	Содержание учебного материала	12	ОК 1
	<i>З1 понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.</i>		
	<i>З3 основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</i>		
	1 Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.		
	2 Условный оператор. Оператор выбора.		
	3 Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы.		
	4 Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.		
	5 Структурированный тип данных – множество. Операции над множествами.		
	6 Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа		
	Лабораторные занятия		
<i>У1 разрабатывать алгоритмы для конкретных задач.</i>	30		
<i>У3 определять сложность работы алгоритмов.</i>			
<i>У4 работать в среде программирования.</i>			
<i>У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.</i>			
<i>У6 оформлять код программы в соответствии со стандартом кодирования.</i>			
<i>У7 выполнять проверку, отладку кода программы.</i>			
Практические занятия			
Знакомство со средой программирования.			
Составление программ линейной структуры.			
Составление программ линейной структуры.			
Составление программ разветвляющейся структуры.			
Составление программ разветвляющейся структуры.			
Составление программ циклической структуры.			
Составление программ циклической структуры.			
Обработка одномерных массивов.			
Обработка двумерных массивов.			
Работа со строками.			
Работа со строками.			
Работа с данными типа множество.			
Составление программ на файлы последовательного доступа.			
Составление программ на типизированные файлы.			
Составление программ на нетипизированные файлы.			
Самостоятельная работа обучающихся	1		
«Виды и описание алгоритмов». (Составление таблицы).			
Раздел 3. Структурное и модульное программирование			

Тема 3.1. Процедуры и функции	Содержание учебного материала		4	OK 4
	<i>34 подпрограммы, составление библиотек подпрограмм.</i>			
	1	Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций.		
	2	Рекурсия. Программирование рекурсивных алгоритмов.		
	Лабораторные занятия		-	
<i>У1 разрабатывать алгоритмы для конкретных задач. У3 определять сложность работы алгоритмов. У4 работать в среде программирования. У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. У6 оформлять код программы в соответствии со стандартом кодирования. У7 выполнять проверку, отладку кода программы.</i>		6		
Практические занятия Организация процедур. Организация функций. Применение рекурсивных функций.				
Тема 3.2. Структуризация в программировании	Содержание учебного материала		4	OK 10
	<i>31 понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. 33 основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</i>			
	1	Основы структурного программирования. Методы структурного программирования.		
	Лабораторные занятия		-	
	Практические занятия		-	
Тема 3.3. Модульное программирование	Содержание учебного материала		4	OK 9
	<i>31 понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. 33 основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</i>			
	1	Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы.		
	2	Стандартные модули.		
	Лабораторные занятия		-	
	<i>У1 разрабатывать алгоритмы для конкретных задач. У3 определять сложность работы алгоритмов. У4 работать в среде программирования. У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. У6 оформлять код программы в соответствии со стандартом кодирования. У7 выполнять проверку, отладку кода программы.</i>		4	
	Практические занятия Программирование модуля.			

	Создание библиотеки подпрограмм.										
Раздел 4. Основные конструкции языков программирования											
Тема 4.1 Указатели	<p>Содержание учебного материала</p> <p><i>33 основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</i></p> <table border="1"> <tr> <td>1</td> <td>Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.</td> </tr> <tr> <td>2</td> <td>Структуры данных на основе указателей.</td> </tr> <tr> <td>3</td> <td>Задача о стеке.</td> </tr> </table> <p>Лабораторные занятия</p> <p><i>У1 разрабатывать алгоритмы для конкретных задач. У2 использовать программы для графического отображения алгоритмов. У3 определять сложность работы алгоритмов. У4 работать в среде программирования. У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. У6 оформлять код программы в соответствии со стандартом кодирования. У7 выполнять проверку, отладку кода программы.</i></p> <p>Практические занятия Использование указателей для организации связанных списков.</p>	1	Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.	2	Структуры данных на основе указателей.	3	Задача о стеке.	6	ОК 2		
1	Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.										
2	Структуры данных на основе указателей.										
3	Задача о стеке.										
Раздел 5. Объектно-ориентированное программирование (ООП)											
Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)	<p>Содержание учебного материала</p> <p><i>35 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.</i></p> <table border="1"> <tr> <td>1</td> <td>История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.</td> </tr> <tr> <td>2</td> <td>Основные принципы ООП: инкапсуляция, наследование, полиморфизм.</td> </tr> <tr> <td>3</td> <td>Классы объектов. Компоненты и их свойства.</td> </tr> <tr> <td>4</td> <td>Событийно-управляемая модель программирования. Компонентно-ориентированный подход.</td> </tr> </table> <p>Лабораторные занятия</p> <p>Практические занятия</p> <p>Самостоятельная работа обучающихся Компоненты и их свойства. (Создание электронного пособия).</p>	1	История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.	2	Основные принципы ООП: инкапсуляция, наследование, полиморфизм.	3	Классы объектов. Компоненты и их свойства.	4	Событийно-управляемая модель программирования. Компонентно-ориентированный подход.	8	ПК 1.1.
1	История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.										
2	Основные принципы ООП: инкапсуляция, наследование, полиморфизм.										
3	Классы объектов. Компоненты и их свойства.										
4	Событийно-управляемая модель программирования. Компонентно-ориентированный подход.										
Тема 5.2 Интегрированная среда разработчика	<p>Содержание учебного материала</p> <p><i>35 объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и</i></p>	6	ПК 1.2.								

	<i>объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.</i>			
1	Требования к аппаратным и программным средствам интегрированной среды разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.			
2	Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.			
3	Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.			
Лабораторные занятия		-		
<i>У1 разрабатывать алгоритмы для конкретных задач. У2 использовать программы для графического отображения алгоритмов. У3 определять сложность работы алгоритмов. У4 работать в среде программирования. У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. У6 оформлять код программы в соответствии со стандартом кодирования. У7 выполнять проверку, отладку кода программы.</i>		6		
Практические занятия Изучение интегрированной среды разработчика. Создание проекта с использованием компонентов для работы с текстом. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.				
Тема 5.3. Визуальное событийно-управляемое программирование	Содержание учебного материала	8	<i>ПК 1.3.</i>	
	<i>35 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.</i>			
	1	Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.		
	2	Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств.		
	3	Назначения свойств и их влияние на результат. Управление объектом через свойства.		
	4	События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.		
	Лабораторные занятия			-
	<i>У1 разрабатывать алгоритмы для конкретных задач. У2 использовать программы для графического отображения алгоритмов. У3 определять сложность работы алгоритмов. У4 работать в среде программирования. У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. У6 оформлять код программы в соответствии со стандартом кодирования. У7 выполнять проверку, отладку кода программы.</i>			14
	Практические занятия События компонентов (элементов управления), их сущность и назначение.			

	Создание процедур на основе событий. Создание проекта с использованием кнопочных компонентов. Создание проекта с использованием кнопочных компонентов. Создание проекта с использованием компонентов стандартных диалогов и системы меню. Создание проекта с использованием компонентов стандартных диалогов и системы меню. Разработка функциональной схемы работы приложения.		
	Самостоятельная работа обучающихся Создание проекта с использованием компонентов стандартных диалогов и системы меню. (Создание приложения).	2	
Тема 5.4 Разработка оконного приложения.	Содержание учебного материала	4	ПК 1.4.
	<i>35 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.</i>		
	1 Разработка функционального интерфейса приложения. Создание интерфейса приложения.		
	2 Разработка функциональной схемы работы приложения. Разработка игрового приложения.		
	Лабораторные занятия	-	
	<i>У1 разрабатывать алгоритмы для конкретных задач. У2 использовать программы для графического отображения алгоритмов. У3 определять сложность работы алгоритмов. У4 работать в среде программирования. У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. У6 оформлять код программы в соответствии со стандартом кодирования. У7 выполнять проверку, отладку кода программы.</i>	4	
Практические занятия Разработка оконного приложения с несколькими формами. Разработка игрового приложения.			
Тема 5.5 Этапы разработки приложений.	Содержание учебного материала	4	ПК 1.5.
	<i>35 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.</i>		
	1 Разработка приложения. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя. Тестирование, отладка приложения.		
	Лабораторные занятия	-	
	<i>У1 разрабатывать алгоритмы для конкретных задач. У2 использовать программы для графического отображения алгоритмов. У3 определять сложность работы алгоритмов. У4 работать в среде программирования. У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. У6 оформлять код программы в соответствии со стандартом кодирования. У7 выполнять проверку, отладку кода программы.</i>	4	
	Практические занятия Создание процедур обработки событий. Компиляция и запуск приложения.		

	Разработка интерфейса приложения. Тестирование, отладка приложения.			
Тема 5.6 Иерархия классов.	Содержание учебного материала	4	ПК 2.4.	
	<i>35 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.</i>			
	1	Классы ООП: виды, назначение, свойства, методы, события.		
	2	Перегрузка методов. Тестирование и отладка приложения. Решение задач		
	Лабораторные занятия			-
	<i>У1 разрабатывать алгоритмы для конкретных задач. У2 использовать программы для графического отображения алгоритмов. У3 определять сложность работы алгоритмов. У4 работать в среде программирования. У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. У6 оформлять код программы в соответствии со стандартом кодирования. У7 выполнять проверку, отладку кода программы.</i>			6
Практические занятия Программирование приложений: виды, назначение, свойства, методы, события. Объявления класса. Создание наследованного класса. Программирование приложений. Перегрузка методов.				
	Дифференцированный зачет			
	Всего:	158		

3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Лаборатория «Программирования баз данных», оснащенная необходимым для реализации программы учебной дисциплины оборудованием, приведенным ниже.

- автоматизированные рабочие места на 14 обучающихся;
- автоматизированное рабочее место на 1 преподавателя;
- сервер в лаборатории (8-х ядерный процессор, оперативная память 16 Гб, жесткие диски общим объемом 1 Тб, программное обеспечение: WindowsServer 2012)
- проектор;
- экран;
- маркерная доска;
- программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО: EclipseIDEforJavaEEDevelopers, .NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, MicrosoftVisioProfessional, MicrosoftVisualStudio, MySQLInstallerforWindows, NetBeans, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio, IntelliJIDEA.

3.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники:

1. Голицына О.Л., Попов И.И. Программирование на языках высокого уровня: учебное пособие [Гриф] - М.: ФОРУМ ,2018.
2. Семакин И.Г. Шестаков А.П. Основы алгоритмизации и программирования. – М.: ОИЦ «Академия», 2016.

Дополнительные источники:

1. Кашаев С.М. Программирование в Microsoft Excel на примерах – СПб.: БХВ-Петербург, 2007.
2. Культин Н.Б. Delphi в задачах и примерах – СПб.: БХВ-Петербург, 2006.
3. Гофман В.Э. Delphi. Экспресс-курс – БХВ-Петербург, 2005.
4. Кузин А.В., Демин В.М. Разработка баз данных Microsoft Access: учебник [Гриф] - М.: ФОРУМ, 2012 .
5. Голицына О.Л., Попов И.И. Основы алгоритмизации и программирования: учебное пособие [Гриф] – М.: ФОРУМ ИНФРА-М, 2005.
6. Агальцев В.П., Волдайская И.В. Математические методы в программировании: учебник - М.: ФОРУМ ИНФРА-М, 2005.

7. Кузин А.В., Демин В.М. Разработка баз данных в системе Microsoft Access: учебник [Гриф] - М.: ФОРУМ ,2005.

Интернет-ресурсы:

1. Лекции по алгоритмизации и программированию [Электронный ресурс] – Режим доступа: <http://www.chemisk.narod.ru/html/algorithm01.html>;
2. Видеолекции [Электронный ресурс] – Режим доступа: <http://video.yandex.ru>;
3. Дистанционное обучение [Электронный ресурс] – Режим доступа: <http://do.rksi.ru/site/help/help8.xml>;
4. Сводная энциклопедия [Электронный ресурс] – Режим доступа: <http://ru.wikipedia.org/wiki/>;
5. Паскаль (язык программирования) Материал из Википедии — свободной Энциклопедии [Электронный ресурс] – Режим доступа: <http://ru.wikipedia.org/wiki/%CF>;
6. Лекции по программированию [Электронный ресурс] – Режим доступа: <http://www.mari-el.ru/mmlab/home/lisp/LECTION6/lec6.htm>;
7. Информационный портал. Все о компьютере и программировании для начинающих info-comp.ru - Здесь есть все! [Электронный ресурс] – Режим доступа: <http://info-comp.ru/programmirovanie/67-turbopascal-.html>.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе тестирования, а также выполнения обучающимися индивидуальных заданий.

Результаты обучения	Критерии оценки	Формы и методы оценки
<p>Перечень умений, осваиваемых в рамках дисциплины:</p> <ul style="list-style-type: none"> - 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. 	<p>«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.</p> <p>«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.</p>	<p>Тестирование; Самостоятельная работа; Выполнение проекта; Наблюдение за выполнением практического задания (деятельностью студента); Оценка выполнения практического задания (работы); Подготовка и выступление с докладом, сообщением, презентацией; Письменный опрос; Устный опрос; Решение ситуационной задачи; Дифференцированный зачет.</p>
<p>Перечень знаний, осваиваемых в рамках дисциплины:</p> <ul style="list-style-type: none"> - эволюцию языков программирования, их классификацию, понятие системы программирования. - подпрограммы, составление библиотек подпрограмм - объектно- 	<p>«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным</p>	

<p>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.</p>	<p>материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.</p> <p>«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p>	
<p>- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</p> <p>- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.</p>		