

Приложение 2.14
к ОПОП-П специальности
09.02.07 Информационные системы и программирование

Рабочая программа дисциплины
«ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

СОДЕРЖАНИЕ ПРОГРАММЫ

СОДЕРЖАНИЕ ПРОГРАММЫ	176
1. Общая характеристика	177
1.1. Цель и место дисциплины в структуре образовательной программы.....	177
1.2. Планируемые результаты освоения дисциплины	177
2. Структура и содержание ДИСЦИПЛИНЫ.....	185
2.1. Трудоемкость освоения дисциплины	185
2.2. Содержание дисциплины.....	186
3. Условия реализации ДИСЦИПЛИНЫ.....	191
3.1. Материально-техническое обеспечение.....	191
3.2. Учебно-методическое обеспечение	191
4. Контроль и оценка результатов освоения ДИСЦИПЛИНЫ.....	191

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

«ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» (наименование дисциплины)

1.1. Цель и место дисциплины в структуре образовательной программы

Цель дисциплины «Основы алгоритмизации и программирования»: изучение и освоение базовых понятий и приемов программирования, применяемых на основных этапах разработки программ; изучение методов программирования для овладения знаниями в области технологии программирования; подготовка к осознанному использованию как языков программирования, так и методов программирования.

Дисциплина «Основы алгоритмизации и программирования» включена в обязательную часть общепрофессионального цикла образовательной программы.

1.2. Планируемые результаты освоения дисциплины

Результаты освоения дисциплины соотносятся с планируемыми результатами освоения образовательной программы, представленными в матрице компетенций выпускника (п. 4.3 ОПОП-П).

В результате освоения дисциплины обучающийся должен¹:

Код ОК, <i>ПК</i>	Уметь	Знать	Владеть навыками
ОК.01	<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. 	<ul style="list-style-type: none"> – Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. – Эволюцию языков программирования, их классификацию, понятие системы программирования. – Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. – Подпрограммы, составление библиотек подпрограмм <p>Объектно-ориентированную модель</p>	-

¹ Берутся сведения, указанные по данному виду деятельности в п. 4.2.

		<p>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>	
ОК.02	<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. 	<ul style="list-style-type: none"> – Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. – Эволюцию языков программирования, их классификацию, понятие системы программирования. – Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. – Подпрограммы, составление библиотек подпрограмм <p>Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма,</p>	

		наследования и переопределения	
ОК.04	<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. 	<ul style="list-style-type: none"> – Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. – Эволюцию языков программирования, их классификацию, понятие системы программирования. – Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. – Подпрограммы, составление библиотек подпрограмм 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 	
ОК.05	<ul style="list-style-type: none"> – Разрабатывать алгоритмы для конкретных задач. – Использовать программы для графического отображения алгоритмов. 	<ul style="list-style-type: none"> – Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. 	

	<ul style="list-style-type: none"> – 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. 	<ul style="list-style-type: none"> – Эволюцию языков программирования, их классификацию, понятие системы программирования. – Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. – Подпрограммы, составление библиотек подпрограмм <p>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</p>	
ОК.09	<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. 	<ul style="list-style-type: none"> – Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. – Эволюцию языков программирования, их классификацию, понятие системы программирования. – Основные элементы языка, структуру программы, операторы и операции, 	-

	<ul style="list-style-type: none"> – Оформлять код программы в соответствии со стандартом кодирования. – Выполнять проверку, отладку кода программы. 	<p>управляющие структуры, структуры данных, файлы, классы памяти.</p> <ul style="list-style-type: none"> – Подпрограммы, составление библиотек подпрограмм <p>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</p>	
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Основные этапы разработки программного обеспечения	Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.
	Оформлять документацию на программные средства	Основные принципы технологии структурного и объектно-ориентированного программирования	
	Оценка сложности алгоритма	Актуальная нормативно-правовая база в области документирования алгоритмов	
ПК 1.2	Создавать программу по разработанному алгоритму как отдельный модуль	Основные этапы разработки программного обеспечения	Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля
	Оформлять документацию на программные средства	Основные принципы технологии структурного и объектно-ориентированного программирования	Разрабатывать мобильные приложения
	Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ	Знание API современных мобильных операционных систем	

ПК 1.3	Выполнять отладку и тестирование программы на уровне модуля	Основные принципы отладки и тестирования программных продуктов	Использовать инструментальные средства на этапе отладки программного продукта
	Оформлять документацию на программные средства	Инструментарий отладки программных продуктов	Проводить тестирование программного модуля по определенному сценарию
	Применять инструментальные средства отладки программного обеспечения		
ПК 1.4	Выполнять отладку и тестирование программы на уровне модуля	Основные виды и принципы тестирования программных продуктов	Проводить тестирование программного модуля по определенному сценарию
	Оформлять документацию на программные средства		Использовать инструментальные средства на этапе тестирования программного продукта
ПК 1.5	Выполнять оптимизацию и рефакторинг программного кода	Способы оптимизации и приемы рефакторинга	Анализировать алгоритмы, в том числе с применением инструментальных средств
	Работать с системой контроля версий	Инструментальные средства анализа алгоритма	Осуществлять рефакторинг и оптимизацию программного кода
		Методы организации рефакторинга и оптимизации кода	
		Принципы работы с системой контроля версий	
ПК 2.4	Использовать выбранную систему контроля версий	Модели процесса разработки программного обеспечения	Разрабатывать тестовые наборы (пакеты) для программного модуля
	Анализировать проектную и техническую документацию	Основные принципы процесса разработки программного обеспечения	Разрабатывать тестовые сценарии программного средства
	Выполнять тестирование интеграции	Основные подходы к интегрированию программных модулей	Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования
	Организовывать постобработку данных	Основы верификации и аттестации программного обеспечения	
	Использовать приемы работы в системах контроля версий	Методы и способы идентификации сбоев и ошибок при интеграции приложений	
	Оценивать размер минимального набора тестов	Методы и схемы обработки исключительных ситуаций	
	Разрабатывать тестовые пакеты и тестовые сценарии	Основные методы и виды тестирования программных продуктов	

	Выполнять ручное и автоматизированное тестирование программного модуля.	Приемы работы с инструментальными средствами тестирования и отладки	
	Выявлять ошибки в системных компонентах на основе спецификаций	Стандарты качества программной документации	
		Основы организации инспектирования и верификации	
		Встроенные и основные специализированные инструменты анализа качества программных продуктов	
		Методы организации работы в команде разработчиков	
ПК 2.5	Использовать выбранную систему контроля версий	Модели процесса разработки программного обеспечения	Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования
	Использовать методы для получения кода с заданной функциональностью и степенью качества	Основные принципы процесса разработки программного обеспечения	
	Анализировать проектную и техническую документацию	Основные подходы к интегрированию программных модулей	
	Организовывать постобработку данных	Основы верификации и аттестации программного обеспечения	
	Приемы работы в системах контроля версий	Стандарты качества программной документации	
	Выявлять ошибки в системных компонентах на основе спецификаций	Основы организации инспектирования и верификации	
		Встроенные и основные специализированные инструменты анализа качества программных продуктов	
		Методы организации работы в команде разработчиков	
		Использовать выбранную систему контроля версий	

1.3.Обоснование часов вариативной части ОПОП-П

№№ п/п	Дополнительные знания, умения, навыки (если указаны ПК)	№, наименование темы	Объем часов	Обоснование включения в рабочую программу
1	Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.	Создание проекта с использованием кнопочных компонентов	2	Расширение кругозора и уровня профессиональных знаний студентов
2	Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля	Создание проекта с использованием компонентов стандартных диалогов и системы меню	2	Расширение кругозора и уровня профессиональных знаний студентов
3	Разрабатывать мобильные приложения	Разработка функциональной схемы работы приложения	2	Расширение кругозора и уровня профессиональных знаний студентов
4	Использовать инструментальные средства на этапе отладки программного продукта	Разработка оконного приложения с несколькими формами	4	Расширение кругозора и уровня профессиональных знаний студентов
5	Проводить тестирование программного модуля по определенному сценарию	Разработка игрового приложения	4	Расширение кругозора и уровня профессиональных знаний студентов
6	Проводить тестирование программного модуля по определенному сценарию	Разработка приложения. Проектирование объектно-ориентированного приложения.	2	Расширение кругозора и уровня профессиональных знаний студентов
7	Использовать инструментальные средства на этапе тестирования программного продукта	Создание интерфейса пользователя. Тестирование, отладка приложения.	2	Расширение кругозора и уровня профессиональных знаний студентов
8	Анализировать алгоритмы, в том числе с применением инструментальных средств	Разработка интерфейса приложения	4	Расширение кругозора и уровня профессиональных знаний студентов
9	Осуществлять рефакторинг и оптимизацию программного кода	Тестирование, отладка приложения	4	Расширение кругозора и уровня профессиональных знаний студентов
10	Разрабатывать тестовые наборы (пакеты) для программного модуля	Классы ООП: виды, назначение, свойства, методы, события. Перегрузка методов.	2	Расширение кругозора и уровня профессиональных знаний студентов

		Тестирование и отладка приложения. Решение задач		профессиональных знаний студентов
11	Разрабатывать тестовые сценарии программного средства	Классы ООП: виды, назначение, свойства, методы, события	2	Расширение кругозора и уровня профессиональных знаний студентов
12	Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования	Объявления класса	2	Расширение кругозора и уровня профессиональных знаний студентов
13	Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования	Создание наследованного класса	2	Расширение кругозора и уровня профессиональных знаний студентов
14	Разрабатывать тестовые сценарии программного средства	Программирование приложений	4	Расширение кругозора и уровня профессиональных знаний студентов
15	Разрабатывать тестовые сценарии программного средства	Перегрузка методов	4	Расширение кругозора и уровня профессиональных знаний студентов

2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Трудоемкость освоения дисциплины

Наименование составных частей дисциплины	Объем в часах	В т.ч. в форме практ. подготовки
Учебные занятия ²	148	52
Курсовая работа (проект)	-	-
Самостоятельная работа	2	-
Промежуточная аттестация в форме экзамена	14	-
Всего	164	52

² Учебные занятия могут представлены в виде теоретических занятий, лабораторных и практических занятий

2.2. Содержание дисциплины

Наименование разделов и тем	Содержание учебного материала, практических и лабораторных занятий, <i>курсовая работа (проект)</i>	Объем, ак. ч. / в том числе в форме практической подготовки, ак. ч.	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
		150/96	
Раздел 1. Введение в программирование		10	
Тема 1.1. Языки программирования	Содержание Развитие языков программирования. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы. Жизненный цикл программы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере. Самостоятельная работа обучающихся Развитие языков программирования. Обзор языков программирования. Области применения языков программирования.	8 2 2 2	ОК.01, ОК.02, ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
Тема 1.2. Типы данных	Содержание Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.	2 2	ОК.01, ОК.02, ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
Раздел 2.		34	
Тема 2.1 Операторы языка программирования	Содержание Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор. Условный оператор. Оператор выбора. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные	34 2 2	ОК.01, ОК.02, ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5,

	циклы.		ПК 2.4, ПК 2.5
	Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.	2	
	Структурированный тип данных – множество. Операции над множествами. Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа	2	
	В том числе практических занятий	26	
	Знакомство со средой программирования.	1	
	Составление программ линейной структуры.	1	
	Составление программ разветвляющейся структуры	2	
	Составление программ циклической структуры	2	
	Обработка одномерных массивов.	2	
	Обработка двумерных массивов.	4	
	Работа со строками.	2	
	Работа с данными типа множество	2	
	Файлы последовательного доступа	2	
	Типизированные файлы	4	
	Нетипизированные файлы	4	
Раздел 3.		22	
Тема 3.1. Процедуры и функции	Содержание	2	ОК.01, ОК.02, ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
	Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций. Рекурсия. Программирование рекурсивных алгоритмов.	2	
Тема 3.2. Структуризация в программировании	Содержание	10	ОК.01, ОК.02, ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
	Основы структурного программирования. Методы структурного программирования.	2	
	В том числе практических занятий	8	
	Организация процедур	2	
	Организация функций	2	
	Применение рекурсивных функций	4	
Тема 3.3.	Содержание	10	ОК.01, ОК.02,

Модульное программирование	Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы. Стандартные модули.	2	ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
	В том числе практических занятий	8	
	Программирование модуля	4	
	Создание библиотеки подпрограмм	4	
Раздел 4. Основные конструкции языков программирования		8	
Тема 4.1. Указатели	Содержание	8	ОК.01, ОК.02, ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
	Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.	2	
	Структуры данных на основе указателей. Задача о стеке.	2	
	В том числе практических занятий	4	
	Использование указателей для организации связанных списков	4	
Раздел 5		78	
Тема 5.1. Основные принципы объектно-ориентированного программирования (ООП)	Содержание	10	ОК.01, ОК.02, ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
	История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.	2	
	Основные принципы ООП: инкапсуляция, наследование, полиморфизм.	2	
	Классы объектов. Компоненты и их свойства. Событийно-управляемая модель программирования. Компонентно-ориентированный подход.	2	
	В том числе практических занятий	4	
	Создание процедур обработки событий. Компиляция и запуск приложения	4	
Тема 5.2. Интегрированная среда разработчика	Содержание	12	ОК.01, ОК.02, ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
	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.	2	
	Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта. Настройка среды и параметров проекта.	2	
	В том числе практических занятий	6	
	Изучение интегрированной среды разработчика	2	
	Создание проекта с использованием компонентов для работы с текстом	2	

	Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени	2	
Тема 5.3. Визуальное событийно-управляемое программирование	Содержание	10	ОК.01, ОК.02, ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
	Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.	2	
	Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.	2	
	События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.	2	
	В том числе практических занятий	4	
	События компонентов (элементов управления), их сущность и назначение	2	
	Создание процедур на основе событий.	2	
Тема 5.4. Разработка оконного приложения	Содержание	18	ОК.01, ОК.02, ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
	Разработка функционального интерфейса приложения. Создание интерфейса приложения.	2	
	Разработка функциональной схемы работы приложения. Разработка игрового приложения.	2	
	В том числе практических занятий	14	
	Создание проекта с использованием кнопочных компонентов	2	
	Создание проекта с использованием компонентов стандартных диалогов и системы меню	2	
	Разработка функциональной схемы работы приложения	2	
	Разработка оконного приложения с несколькими формами	4	
	Разработка игрового приложения	4	
Тема 5.5. Этапы разработки приложений	Содержание	12	ОК.01, ОК.02, ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
	Разработка приложения. Проектирование объектно-ориентированного приложения.	2	
	Создание интерфейса пользователя. Тестирование, отладка приложения.	2	
	В том числе практических занятий	8	
	Разработка интерфейса приложения	4	
	Тестирование, отладка приложения	4	
Тема 5.6.	Содержание	16	ОК.01, ОК.02,

Иерархия классов	Классы ООП: виды, назначение, свойства, методы, события. Перегрузка методов. Тестирование и отладка приложения. Решение задач	2	ОК.04, ОК.05, ОК.09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
	В том числе практических занятий	14	
	Классы ООП: виды, назначение, свойства, методы, события	2	
	Объявления класса	2	
	Создание наследованного класса	2	
	Программирование приложений	4	
	Перегрузка методов	4	
Промежуточная аттестация		14	
Всего:		164	

3. УСЛОВИЯ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ

3.1. Материально-техническое обеспечение

Кабинет «Информационных технологий в профессиональной деятельности», оснащенный(е) в соответствии с приложением 3 ОПОП-П.

Лаборатория «Вычислительной техники, архитектуры персонального компьютера и периферийных устройств», «Программирования и баз данных», «Программного обеспечения и сопровождения компьютерных систем», оснащенные в соответствии с приложением 3 ОПОП-П.

3.2. Учебно-методическое обеспечение

3.2.1. Основные печатные и/или электронные издания

1. Основы Алгоритмизации и программирования: практикум / Е.Г. Серкова. – Ростов н/Д : Феникс, 2019. – 188, с. : ил. – (Среднее профессиональное образование).

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Результаты обучения	Показатели освоённости компетенций	Методы оценки
<p><i>Умеет:</i></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><i>Знает:</i></p> <ul style="list-style-type: none"> – Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные 	<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>«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий</p>	<p>Примеры форм и методов контроля и оценки</p> <ul style="list-style-type: none"> – Компьютерное тестирование на знание терминологии по теме; – Тестирование – Контрольная работа – Самостоятельная работа. – Защита реферата – Семинар – Выполнение проекта; – Наблюдение за выполнением практического задания. (деятельностью студента) – Оценка выполнения практического задания(работы) – Подготовка и выступление с докладом, сообщением, презентацией <p>Решение ситуационной задачи</p>

<p>алгоритмические конструкции.</p> <ul style="list-style-type: none"> – Эволюцию языков программирования, их классификацию, понятие системы программирования. – Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. – Подпрограммы, составление библиотек подпрограмм – 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. 	<p>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>	
--	---	--

