

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ  
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»  
**КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)**

Среднее профессиональное образование

**ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**  
**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Специальность  
*09.02.05 Прикладная информатика (по отраслям)*

Квалификация выпускника:  
*Техник-программист*

Казань 2019

Рабочая программа учебной дисциплины «Основы алгоритмизации и программирования» (для 2019 года набора) разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.05 Прикладная информатика (по отраслям), утвержденного Министерством образования и науки Российской Федерации от 13 августа 2014 г. № 1001 и учебного плана, утвержденного Ученым советом Российского университета кооперации.

*Разработчики:*

Ахмедова А.М., преподаватель кафедры естественных дисциплин, сервиса и туризма Казанского кооперативного института (филиала) Российского университета кооперации.

**Рабочая программа:**

**обсуждена и рекомендована** к утверждению решением кафедры естественных дисциплин, сервиса и туризма Казанского кооперативного института (филиала) от 13.03.2019, протокол № 7

**одобрена** Научно-методическим советом Казанского кооперативного института (филиала) от 03.04.2019, протокол №5

**утверждена** Ученым советом Российского университета кооперации от 18.04.2019, протокол №4

©Казанский кооперативный институт (филиал) Российского университета кооперации, 2019  
© Ахмедова А.М., 2019

## СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» .....	4
1.1. Область применения программы.....	4
1.2. Место дисциплины в структуре основной профессиональной образовательной программы.....	4
1.3. Цель и задачи учебной дисциплины – требования к результатам освоения учебной дисциплины.....	4
1.4. Количество часов на освоение рабочей программы учебной дисциплины (по ФГОС):.....	6
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	6
2.1. Объем учебной дисциплины и виды учебной работы.....	6
2.2. Тематический план и содержание учебной дисциплины «Основы алгоритмизации и программирования» .....	7
3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ .....	14
3.1. Требования к минимальному материально-техническому обеспечению	14
3.2. Информационное обеспечение обучения .....	14
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ	15

# **1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

## **1.1. Область применения программы**

Рабочая программа учебной дисциплины является частью программы подготовки специалистов среднего звена в соответствии с Федеральным государственным образовательным стандартом от 13.08.2014 г. №1001, и учебным планом, утвержденным Ученым советом Российского университета кооперации по специальности СПО 09.02.05 Прикладная информатика (по отраслям).

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

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

## **1.3. Цель и задачи учебной дисциплины – требования к результатам освоения учебной дисциплины**

Сформировать базовые теоретические знания и практические навыки обучающегося в области программирования для освоения общих и профессиональных компетенций по специальности «Прикладная информатика (по отраслям)».

В результате изучения дисциплины обучающийся должен

### **уметь:**

- 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.

### **знать:**

- 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.

Изучение дисциплины направлено на формирование следующих компетенций:

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.

ПК 1.1 Обрабатывать статический информационный контент.

ПК 1.2 Обрабатывать динамический информационный контент.

ПК 2.2 Разрабатывать и публиковать программное обеспечение и информационные ресурсы отраслевой направленности со статическим и динамическим контентом на основе готовых спецификаций и стандартов.

ПК 2.3 Проводить отладку и тестирование программного обеспечения отраслевой направленности.

ПК 3.3 Проводить обслуживание, тестовые проверки, настройку программного обеспечения отраслевой направленности.

**1.4. Количество часов на освоение рабочей программы учебной дисциплины (по ФГОС):**

	<b>Очная форма</b>	<b>Заочная форма</b>
Максимальная учебная нагрузка обучающегося:	58	58
- обязательная аудиторная учебная нагрузка	34	10
- самостоятельная работа обучающегося	24	48

**2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

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

*очная форма обучения*

<b>Вид учебной работы</b>	<b>Объем часов</b>
<b>Максимальная учебная нагрузка (всего)</b>	58
<b>Обязательная аудиторная учебная нагрузка (всего)</b>	34
теоретическое обучение	18
лабораторные занятия	16
<b>Самостоятельная работа обучающегося (всего)</b>	24
<b>Промежуточная аттестация</b> (5 семестр) на базе основного общего образования (3 семестр) на базе среднего общего образования	Экзамен

*заочная форма обучения*

<b>Вид учебной работы</b>	<b>Объем часов</b>
<b>Максимальная учебная нагрузка (всего)</b>	58
<b>Обязательная аудиторная учебная нагрузка (всего)</b>	10
теоретическое обучение	4
лабораторные занятия	6
<b>Самостоятельная работа обучающегося (всего)</b>	48
<b>Промежуточная аттестация</b> (3 курс) на базе основного общего образования	Экзамен

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

*очная форма обучения*

Наименование разделов и тем	Содержание учебного материала, лабораторные занятия, самостоятельная работа обучающихся	Объем часов	Уровень освоения
1	2	3	4
<b>Раздел 1. Основные принципы алгоритмизации и программирования</b>			
<b>Тема 1.1</b> Основные понятия алгоритмизации. Языки и системы программирования	<b>Содержание учебного материала</b> Понятие алгоритма. Свойства алгоритмов. Формы записей алгоритмов. Общие принципы построения алгоритмов. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические. Данные: понятие и типы. Основные базовые типы данных и их характеристика. Структурированные типы данных и их характеристика. Классификация языков программирования. Элементы языков программирования. Понятие системы программирования. Исходный, объектный и загрузочный модули. Интегрированная среда программирования.	2	1
	<b>Лабораторные занятия</b> Составление блок-схем линейных алгоритмов, разветвляющихся алгоритмов, циклических алгоритмов	2	2
	<b>Контрольные работы по теме «Составление блок-схем алгоритмов»</b>		
	<b>Самостоятельная работа обучающихся</b> Составление блок-схем линейных алгоритмов Составление блок-схем разветвляющихся алгоритмов Составление блок-схем циклических алгоритмов	4	3
<b>Раздел 2. Программирование на алгоритмическом языке</b>			
<b>Тема 2.1.</b> Основные элементы языка	<b>Содержание учебного материала</b> История развития языка программирования. Структурная схема программы на алгоритмическом языке. Лексика языка. Переменные и константы. Типы данных. Выражения и операции. Синтаксис операторов: присваивания, ввода-вывода. Синтаксис операторов: безусловного и условного переходов. Синтаксис операторов: циклов. Составной оператор	2	1
	<b>Лабораторные занятия</b> Составление программ линейной структуры. Составление программ разветвляющейся структуры.	2	2
	<b>Самостоятельная работа обучающихся</b> Типовой расчет «Задачи линейной и ветвящейся конструкции»	4	3

	Типовой расчет «Задачи циклической конструкции»		
<b>Тема 2.2.</b> Массивы	<b>Содержание учебного материала</b>		
	Массивы, как структурированный тип данных. Объявление массива. Стандартные функции для массива целых и вещественных чисел. Ввод и вывод одномерных массивов. Ввод и вывод двумерных массивов. Обработка массивов.	2	1
	<b>Лабораторные занятия</b> Обработка одномерных массивов. Обработка двумерных массивов. Использование стандартных функций для работы с массивами.	2	2
	<b>Самостоятельная работа обучающихся</b> Типовой расчет «Одномерные массивы» Типовой расчет «Двумерные массивы»	2	3
<b>Тема 2.3.</b> Строки и множества Процедуры и функции	<b>Содержание учебного материала</b>		
	Структурированные типы данных: строки и множества. Объявление строковых типов данных. Операции со строками. Стандартные функции и процедуры для работы со строками. Объявление множества. Операции над множествами. Понятие подпрограммы. Организация процедур, стандартные процедуры. Процедуры, определенные пользователем: синтаксис, передача аргументов. Процедуры с параметрами, описание процедур. Функции: способы организации и описание. Вызов функций, рекурсия.	2	1
	<b>Лабораторные занятия</b> Работа со строковыми переменными. Работа с данными типа множество. Разработка программ со структурированными типами данных. Организация процедур. Использование процедур. Организация функций. Использование функций.	2	2
	<b>Самостоятельная работа обучающихся</b> Типовой расчет «Строки и символы» Типовой расчет «Множества» Типовой расчет «Комбинированный тип»	4	3
<b>Тема 2.4.</b> Организация ввода-вывода данных. Работа с файлами	<b>Содержание учебного материала</b>		
	Типы файлов. Организация доступа к файлам. Файлы последовательного доступа. Открытие и закрытие файла последовательного доступа. Запись в файл и чтение из файла последовательного доступа. Файлы произвольного доступа. Порядок работы с файлами произвольного доступа. Создание	2	1



	структуры записи. Открытие и закрытие файла произвольного доступа.		
	<b>Самостоятельная работа обучающихся</b> Типовой расчет «Типизированные файлы» Типовой расчет «Текстовые файлы»	2	3
<b>Раздел 3. Программирование в объектно-ориентированной среде</b>			
<b>Тема 3.1.</b> Основные принципы объектно-ориентированного программирования (ООП)	<b>Содержание учебного материала</b>	2	1
	История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. Классы объектов. Компоненты и их свойства. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.		
	<b>Лабораторные занятия</b> Изучение интегрированной среды разработчика. Создание простого проекта.		
<b>Тема 3.2.</b> Иерархия классов	<b>Содержание учебного материала</b>	2	1
	Классы объектно-ориентированного языка программирования: виды, назначение, свойства, методы, события. Объявление класса, свойств и методов экземпляра класса. Наследование. Перегрузка методов.		
	<b>Лабораторные занятия</b> Объявление класса, создание экземпляров класса. Создание наследованного класса. Перегрузка методов.		
<b>Тема 3.3.</b> Визуальное событийно-управляемое программирование	<b>Содержание учебного материала</b>	2	1
	Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Свойства компонентов (элементов управления). Виды свойств. Синтаксис определения свойств. Категория свойств. Назначение свойств и их влияние на результат. Управление объектом через свойства. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий. Процедуры, определенные пользователем: синтаксис, передача аргументов. Вызов событий.		
	<b>Лабораторные занятия</b> Создание проекта с использованием кнопочных компонентов.		

	Создание проекта с использованием компонентов для работы с текстом. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.		
	<b>Самостоятельная работа обучающихся</b> Создание программного продукта: калькулятор Создание программного продукта: текстовый редактор	4	3
<b>Тема</b> 3.4. Разработка оконного приложения	<b>Содержание учебного материала</b> Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Создание процедур обработки событий. Компиляция и запуск приложения.	2	1
	<b>Лабораторные занятия</b> Разработка оконного приложения. Разработка оконного приложения с несколькими формами. Разработка многооконного приложения.	2	2
	<b>Контрольные работы по теме «Объектно-ориентированное программирование»</b>		
	<b>Самостоятельная работа обучающихся</b> Создание программного продукта: график функции. Создание программного продукта: обучающее - контролирующая программа.	4	3
<b>Итого часов по дисциплине: 58</b>		<b>18/16/24</b>	

*заочная форма обучения*

Наименование разделов и тем	Содержание учебного материала, лабораторные занятия, самостоятельная работа обучающихся	Объем часов	Уровень освоения
1	2	3	4
<b>Раздел 1. Основные принципы алгоритмизации и программирования</b>			
<b>Тема 1.1</b> Основные понятия алгоритмизации. Языки и системы программирования	<b>Содержание учебного материала</b> Понятие алгоритма. Свойства алгоритмов. Формы записей алгоритмов. Общие принципы построения алгоритмов. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические. Данные: понятие и типы. Основные базовые типы данных и их характеристика. Структурированные типы данных и их характеристика. Классификация языков программирования. Элементы языков программирования. Понятие системы программирования. Исходный, объектный и загрузочный модули. Интегрированная среда программирования.	2	1
	<b>Лабораторные занятия</b> Составление блок-схем линейных алгоритмов, разветвляющихся алгоритмов, циклических алгоритмов	2	2

	<b>Контрольные работы по теме «Составление блок-схем алгоритмов»</b>		
	<b>Самостоятельная работа обучающихся</b> Составление блок-схем линейных алгоритмов Составление блок-схем разветвляющихся алгоритмов Составление блок-схем циклических алгоритмов	4	3
<b>Раздел 2. Программирование на алгоритмическом языке</b>			
<b>Тема 2.1.</b> Основные элементы языка	<b>Содержание учебного материала</b>		
	История развития языка программирования. Структурная схема программы на алгоритмическом языке. Лексика языка. Переменные и константы. Типы данных. Выражения и операции. Синтаксис операторов: присваивания, ввода-вывода. Синтаксис операторов: безусловного и условного переходов. Синтаксис операторов: циклов. Составной оператор	2	1
	<b>Лабораторные занятия</b> Составление программ линейной структуры. Составление программ разветвляющейся структуры.	2	2
	<b>Самостоятельная работа обучающихся</b> Типовой расчет «Задачи линейной и ветвящейся конструкции» Типовой расчет «Задачи циклической конструкции»	4	3
<b>Тема 2.2.</b> Массивы	<b>Самостоятельная работа обучающихся</b>		
	Массивы, как структурированный тип данных. Объявление массива. Стандартные функции для массива целых и вещественных чисел. Ввод и вывод одномерных массивов. Ввод и вывод двумерных массивов. Обработка массивов.	2	3
	<b>Лабораторные занятия</b> Обработка одномерных массивов. Обработка двумерных массивов. Использование стандартных функций для работы с массивами.	2	2
	<b>Самостоятельная работа обучающихся</b> Типовой расчет «Одномерные массивы» Типовой расчет «Двумерные массивы»	2	3
<b>Тема 2.3.</b> Строки и множества Процедуры и функции	<b>Самостоятельная работа обучающихся</b>		
	Структурированные типы данных: строки и множества. Объявление строковых типов данных. Операции со строками. Стандартные функции и процедуры для работы со строками. Объявление множества. Операции над множествами. Понятие подпрограммы. Организация процедур, стандартные процедуры. Процедуры, определенные пользователем: синтаксис, передача аргументов. Процедуры с параметрами,	2	3

	описание процедур. Функции: способы организации и описание. Вызов функций, рекурсия.		
	Работа со строковыми переменными. Работа с данными типа множество. Разработка программ со структурированными типами данных. Организация процедур. Использование процедур. Организация функций. Использование функций.	2	
	Типовой расчет «Строки и символы» Типовой расчет «Множества» Типовой расчет «Комбинированный тип»	4	
<b>Тема 2.4.</b> Организация ввода-вывода данных. Работа с файлами	<b>Самостоятельная работа обучающихся</b> Типы файлов. Организация доступа к файлам. Файлы последовательного доступа. Открытие и закрытие файла последовательного доступа. Запись в файл и чтение из файла последовательного доступа. Файлы произвольного доступа. Порядок работы с файлами произвольного доступа. Создание структуры записи. Открытие и закрытие файла произвольного доступа.	2	3
	Типовой расчет «Типизированные файлы» Типовой расчет «Текстовые файлы»	2	
<b>Раздел 3. Программирование в объектно-ориентированной среде</b>			
<b>Тема 3.1.</b> Основные принципы объектно-ориентированного программирования (ООП)	<b>Самостоятельная работа обучающихся</b> История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. Классы объектов. Компоненты и их свойства. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.	2	3
	Изучение интегрированной среды разработчика. Создание простого проекта.	2	
<b>Тема 3.2.</b> Иерархия классов	<b>Самостоятельная работа обучающихся</b> Классы объектно-ориентированного языка программирования: виды, назначение, свойства, методы, события. Объявление класса, свойств и методов экземпляра класса. Наследование. Перегрузка методов.	2	3

	Объявление класса, создание экземпляров класса. Создание наследованного класса. Перегрузка методов.	2	
<b>Тема 3.3.</b> Визуальное событийно- управляемое программирование	<b>Самостоятельная работа обучающихся</b>		3
	Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Свойства компонентов (элементов управления). Виды свойств. Синтаксис определения свойств. Категория свойств. Назначение свойств и их влияние на результат. Управление объектом через свойства. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий. Процедуры, определенные пользователем: синтаксис, передача аргументов. Вызов событий.	2	
	Создание проекта с использованием кнопочных компонентов. Создание проекта с использованием компонентов для работы с текстом. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.	2	
	Создание программного продукта: калькулятор Создание программного продукта: текстовый редактор	4	
<b>Тема 3.4.</b> Разработка оконного приложения	<b>Самостоятельная работа обучающихся</b>		3
	Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Создание процедур обработки событий. Компиляция и запуск приложения.	2	
	Разработка оконного приложения. Разработка оконного приложения с несколькими формами. Разработка многооконного приложения.	2	
	Создание программного продукта: график функции. Создание программного продукта: обучающее - контролирующая программа.	4	
<b>Итого часов по дисциплине: 58</b>		<b>4/6/48</b>	

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 ПРОГРАММЫ ДИСЦИПЛИНЫ

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

Учебная дисциплина реализуется в лаборатории разработки, внедрения и адаптации программного обеспечения отраслевой направленности.

*Оборудование лаборатории:*

Стандартная учебная мебель:

Стол письменный;

Столы компьютерные;

Столы аудиторные двухместные;

Стул;

Стулья ученические;

Доска аудиторная;

Кафедра.

*Демонстрационное оборудование и учебно-наглядные пособия:*

Стенд СШ-525 «Развитие вычислительной техники»;

Стенд СШ-519 «Великие люди в мире информатики».

*Оборудование, технические средства обучения:*

Системные блоки;

Мониторы;

Клавиатура;

Мыши.

*Программное обеспечение:*

1. Desktop School ALNG LicSAPk MVL.

a. Office ProPlus All Lng Lic/SA Pack MVL Partners in Learning

b. Windows 8/

2. Ide Lazarus;

3. Free Pascal;

4. FreeBASIC;

5. Система тестирования INDIGO.

6. Интернет-браузеры Google Chrome, Firefox.

#### 3.2. Информационное обеспечение обучения

**Основная литература:**

1. Колдаев, В.Д. Основы алгоритмизации и программирования [Электронный ресурс]: учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2019. - 416 с.: ил. - Режим доступа: <http://znanium.com/catalog/product/980416>

### Дополнительная литература:

1. Фризен И.Г. Основы алгоритмизации и программирования (среда PascalABC.NET) [Электронный ресурс]: учеб. пособие / И.Г. Фризен. — М. : ФОРУМ : ИНФРА-М, 2019. — 392 с. - Режим доступа: <http://znanium.com/catalog/product/1007486>

2. Ночка, Е.И. Основы алгоритмизации и программирования на языке Питон [Электронный ресурс]: учебник / Е.И. Ночка- М.:КУРС, 2017. - 208 с. - Режим доступа: <http://znanium.com/catalog/product/795688>

3. Гуриков, С.Р. Основы алгоритмизации и программирования на Python [Электронный ресурс]: учеб. пособие / С.Р. Гуриков. — М. : ФОРУМ : ИНФРА-М, 2018. — 343 с. - Режим доступа: <http://znanium.com/catalog/product/961522>

4. Ночка, Е.И. Основы алгоритмизации и программирования. Ответы на контрольные вопросы [Электронный ресурс] / Е.И. Ночка - М.:КУРС, НИЦ ИНФРА-М, 2017. - 59 с. - Режим доступа: <http://znanium.com/catalog.php?bookinfo=772548>

## 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

**Контроль и оценка результатов освоения учебной дисциплины осуществляется преподавателем в процессе проведения лабораторных занятий, проверки выполнения самостоятельной работы, решения задач.**

Оценка качества освоения учебной программы включает текущий контроль успеваемости, промежуточную аттестацию по итогам освоения дисциплины.

Результаты обучения: умения, знания	Форма контроля и оценивания
<b>Уметь:</b>	
- разрабатывать алгоритмы для конкретных задач;	Лабораторные работы, контрольная работа. Экзамен
- использовать программы для графического отображения алгоритмов;	Лабораторные работы, контрольная работа. Экзамен
- определять сложность работы алгоритмов;	Лабораторные работы, контрольная работа. Экзамен
- работать в среде программирования;	Лабораторные работы, контрольная работа. Экзамен
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;	Лабораторные работы, контрольная работа. Экзамен
- оформлять код программы в соответствии со стандартом кодирования;	Лабораторные работы, контрольная работа. Экзамен
- выполнять проверку, отладку кода программы.	Лабораторные работы, контрольная работа. Экзамен
<b>Знать:</b>	
- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;	Лабораторные работы, контрольная работа. Экзамен

- эволюцию языков программирования, их классификацию, понятие системы программирования;	Лабораторные работы, контрольная работа. Экзамен
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;	Лабораторные работы, контрольная работа. Экзамен
- подпрограммы, составление библиотек подпрограмм;	Лабораторные работы, контрольная работа. Экзамен
- 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.	Лабораторные работы, контрольная работа. Экзамен