

МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖНОЙ  
ПОЛИТИКИ НИЖЕГОРОДСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
“НИЖЕГОРОДСКИЙ ГУБЕРНСКИЙ КОЛЛЕДЖ”

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ  
ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Специальность:

09.02.07 Информационные системы и программирование

Нижний Новгород  
2020 г.

Рассмотрена методической комиссией  
информатики и вычислительной техники  
протокол №\_1\_\_от 26.08.2020 г.  
Председатель МК \_\_\_\_\_ И.А. Мухин

УТВЕРЖДАЮ

Заместитель директора по учебной работе

\_\_\_\_\_ О.Ю. Евчинникова

« 01 » августа 2020 г.



Рабочая программа учебной дисциплины ОП.04 Основы алгоритмизации и программирования разработана на основе Федерального государственного образовательного стандарта по специальности 09.02.07 Информационные системы и программирование среднего профессионального образования.

Организация-разработчик: ГБПОУ Нижегородский Губернский колледж

Разработчик:

Мурева Ольга Николаевна, преподаватель информатики

## СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ .....	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ .....	5
3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ .....	12
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ .....	13

# 1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

## 1.1. Место дисциплины в структуре основной профессиональной образовательной программы

Учебная дисциплина «Основы алгоритмизации и программирования» принадлежит к общепрофессиональному циклу.

## 1.2. Цель и планируемые результаты освоения дисциплины:

Код ПК, ОК	Умения	Знания
ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5	<p>Разрабатывать алгоритмы для конкретных задач.</p> <p>Использовать программы для графического отображения алгоритмов.</p> <p>Определять сложность работы алгоритмов.</p> <p>Работать в среде программирования.</p> <p>Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.</p> <p>Оформлять код программы в соответствии со стандартом кодирования.</p> <p>Выполнять проверку, отладку кода программы.</p>	<p>Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.</p> <p>Эволюцию языков программирования, их классификацию, понятие системы программирования.</p> <p>Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</p> <p>Подпрограммы, составление библиотек подпрограмм</p> <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>

## **2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

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

<b>Вид учебной работы</b>	<b>Объем часов</b>
<b>Всего объем образовательной нагрузки</b>	<b>152</b>
<b>Всего занятий</b>	<b>148</b>
в том числе:	
– учебные занятия	64
– лабораторные, практические занятия	76
– консультации	8
<b>Самостоятельная работа обучающегося (всего)</b>	<b>4</b>
<b>Итоговая аттестация в форме экзамена – 8 часов</b>	

## 2.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающегося, курсовая работа (проект)	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
<b>Раздел 1. Введение в программирование</b>		<b>32</b>	
<b>Тема 1.1. Основы алгоритмизации</b>	<b>Содержание учебного материала</b>	<b>18</b>	ОК 2, ОК 4 , ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.5, ПК 2.4, 2.5
	1. Место и роль программирования и алгоритмизации в современных компьютерных технологиях. Понятие алгоритмизации и алгоритма. Свойства алгоритмов.	2	
	2. Общие принципы построения алгоритмов. Способы описания алгоритмов. Правила выполнения схем алгоритмов. Основные конструкции алгоритмического языка: линейный алгоритм, ветвление, цикл.	4	
	3. Этапы решения задач с помощью ЭВМ: постановка задачи, создание модели, алгоритм, кодирование алгоритма, анализ результатов. Правила постановки задачи. Способы описания алгоритмов: псевдокоды, блок-схема, программа.	4	
	<b>Практические занятия</b>	<b>8</b>	
	1. Блок-схема: основные элементы, правила составления. Примеры. Составление блок-схем линейных алгоритмов.	2	

	2.	Составление блок-схем разветвляющихся алгоритмов.	2	
	3.	Составление блок-схем циклических алгоритмов.	2	
	4.	Примеры решения нелинейных и линейных уравнений.	2	
<b>Тема 1.2. Языки программирования</b>	<b>Содержание учебного материала</b>		<b>8</b>	ОК 2, ОК 4 , ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.5, ПК 2.4, 2.5
	1.	Развитие языков программирования. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.	4	
	2.	Жизненный цикл программы программа. Программный продукт и его характеристики.	4	
<b>Тема 1.3. Типы данных</b>	<b>Содержание учебного материала</b>		<b>6</b>	ОК 2, ОК 4 , ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.5, ПК 2.4, 2.5
	1.	Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.	2	
	2.	Правила формирования и вычисления выражений. Типы выражений.	2	
	3.	Консультация <sup>1</sup> : Словарь языка программирования	2	
<b>Раздел 2. Основные конструкции языков программирования.</b>			<b>84</b>	
<b>Тема 2.1 Операторы языка программирования</b>	<b>Содержание учебного материала</b>		<b>14</b>	ОК 2, ОК 4 , ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.5, ПК 2.4, 2.5
	1.	Структура программы на языке программирования Python. Операции и переменные. Приоритет выполнения операций. Операнды: литералы, выражения, переменные.	4	
	2.	Динамическая типизация. Инструкция присваивания. Запись. Операции с текстовыми строками в интерпретаторе Python.	4	

<sup>1</sup> Проводится по графику учебного процесса

	3.	Консультация <sup>2</sup> : Тестирование и отладка кода в Python	2	
	<b>Практические занятия</b>		<b>4</b>	
	1.	Ввод данных-функция input(). Пример использования. Вывод данных – инструкция print().Пример использования.	2	
	2.	Функции преобразования в числовые типы int() и float().Пример использования. Форматирование вывода – функция format().Пример использования.	2	
<b>Тема 2.2 Структурное программирование</b>	<b>Содержание учебного материала</b>		<b>42</b>	ОК 2, ОК 4 , ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.5, ПК 2.4, 2.5
	1.	Предпрограммная подготовка задачи: постановка, формализация, создание алгоритма, программирование, тестирование и отладка.	2	
	2.	Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.	2	
	3.	Условный оператор. Оператор выбора.	2	
	4.	Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы.	2	
	5.	Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.	2	
	<b>Практические занятия</b>		<b>32</b>	
	1.	Линейные алгоритмы. Правила записи арифметических выражений. Инструменты обработки целых чисел. Тип данных float().Инструменты обработки вещественных чисел.	2	
	2.	Запись арифметических выражений в языке Python. Множественное присваивание. Операции обработки числовых данных. Функция модуля числа abs().	2	

<sup>2</sup> Проводится по графику учебного процесса



	3.	Операции: безостаточного деления ( $a/b$ ) и остаток от деления ( $a\%b$ ). Функция генерации целого случайного числа <code>random</code> . <code>randint(a,b)</code> . Функции модуля <code>Math</code> : <code>sqrt(x)</code> , <code>sin(x)</code> , <code>cos(x)</code> , <code>log(x)</code> , <code>exp(x)</code> . Функция округления к целому значению <code>round(x)</code> .	2	
	4.	Разветвляющие алгоритмы. Инструкция ветвления. Сложные условия и каскадные ветвления.	2	
	5.	Циклические алгоритмы. Виды циклов. Запись цикла с условием. Запись цикла с постусловием.	2	
	6.	Принцип работы цикла с параметром. Сложные циклические алгоритмы. Вложенные циклы.	2	
	7.	Одномерные массивы. Решение классических задач. Сортировка. Числа Фибоначчи.	4	
	8.	Двумерные массивы. Решение задач на графы.	4	
	9.	Строки.	4	
	10.	Файлы.	4	
	11.	Контейнеры и динамические массивы.	4	
<b>Тема 2.3 Модульное программирование, структуры данных</b>	<b>Содержание учебного материала</b>		<b>28</b>	ОК 2, ОК 4 , ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.5, ПК 2.4, 2.5
	1.	Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций.	2	
	2.	Рекурсия. Программирование рекурсивных алгоритмов.	2	
	3.	Стандартные модули. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы.	2	
	4.	Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.	2	

	5.	Структуры данных на основе указателей. Задача о стеке.	2	
	6.	Консультация <sup>3</sup> : Контейнеры и динамические массивы.	2	
	<b>Практические занятия</b>		<b>16</b>	
	1.	Программирование рекурсии	2	
	2.	Программирование и компиляция модулей	4	
	3.	Стек, очередь и дек. Их реализация.	4	
	4.	Итераторы. Умные указатели.	2	
	5.	Алгоритмы сортировки и поиска. Сложность алгоритма.	4	
<b>Раздел 3. Основы объектно-ориентированного программирования.</b>			<b>32</b>	
<b>Тема 3.1 Концепция объектно-ориентированного программирования.</b>	<b>Содержание учебного материала</b>		<b>18</b>	ОК 2, ОК 4 , ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.5, ПК 2.4, 2.5
	1.	История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.	2	
	2.	Основные принципы ООП: инкапсуляция, наследование, полиморфизм.	2	
	3.	Классы объектов. Компоненты и их свойства.	2	
	4.	Событийно-управляемая модель программирования. Компонентноориентированный подход.	2	
	5.	Принципы объектно-ориентированного программирования: наследование.	2	

<sup>3</sup> Проводится по графику учебного процесса

	6.	Консультация <sup>4</sup> : Принципы объектно-ориентированного программирования: полиморфизм.	2	
	<b>Практические занятия</b>		<b>6</b>	
	1.	Программирование объектной модели. Взаимодействие экземпляров различных классов.	2	
	2.	Программирование объектов и классов.	2	
	3.	Определение инкапсуляции. Применение инкапсуляции в программировании.	2	
<b>Тема 3.2 Разработка оконного приложения</b>	<b>Содержание учебного материала</b>		<b>14</b>	ОК 2, ОК 4 , ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.5, ПК 2.4, 2.5
	1.	Разработка функционального интерфейса приложения. Создание интерфейса приложения.	2	
	2.	Разработка функциональной схемы работы приложения.	2	
	<b>Практические занятия</b>		<b>10</b>	
	1.	Проектирование объектно-ориентированного приложения.	2	
	2.	Разработка игрового приложения.	4	
	3.	Создание интерфейса пользователя.	2	
	4.	Тестирование, отладка приложения.	2	
<b>Самостоятельная работа<sup>5</sup></b>	1.	Автоматизация типичных задач	4	ОК 2, ОК 4 , ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.5, ПК 2.4, 2.5

<sup>4</sup> Проводится по графику учебного процесса

<sup>5</sup> Проводится по графику учебного процесса

### **3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

- а. Для реализации программы учебной дисциплины должны быть предусмотрены следующие специальные помещения:

Лаборатория «**Программирование баз данных**», оснащенная необходимым для реализации программы учебной дисциплины оборудованием, приведенным в п 6.1.2.1 примерной программы по данной специальности.

#### **б. Информационное обеспечение реализации программы**

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, рекомендуемых для использования в образовательном процессе

#### **3.2.1. Печатные издания**

1. Семакин И.Г., Шестаков А.П. Основы алгоритмизации и программирования. –М.: ОИЦ «Академия», 2016

#### 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

Результаты обучения	Критерии оценки	Формы и методы оценки
<p><i>Перечень умений, осваиваемых в рамках дисциплины:</i></p> <ul style="list-style-type: none"> <li>Разрабатывать алгоритмы для конкретных задач.</li> <li>Использовать программы для графического отображения алгоритмов.</li> <li>Определять сложность работы алгоритмов.</li> <li>Работать в среде программирования.</li> <li>Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.</li> <li>Оформлять код программы в соответствии со стандартом кодирования.</li> <li>Выполнять проверку, отладку кода программы.</li> </ul>	<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><b>Примеры форм и методов контроля и оценки</b></p> <ul style="list-style-type: none"> <li>Компьютерное тестирование на знание терминологии по теме;</li> <li>Тестирование....</li> <li>Контрольная работа ....</li> <li>Самостоятельная работа.</li> <li>Защита реферата....</li> <li>Семинар</li> <li>Защита курсовой работы (проекта)</li> <li>Выполнение проекта;</li> <li>Наблюдение за выполнением практического задания. (деятельностью студента)</li> </ul>
<p><i>Перечень знаний, осваиваемых в рамках дисциплины:</i></p> <ul style="list-style-type: none"> <li>Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.</li> <li>Эволюцию языков программирования, их классификацию, понятие системы программирования.</li> <li>Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</li> <li>Подпрограммы, составление библиотек подпрограмм</li> </ul>	<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 выполнено, некоторые из</p>	<ul style="list-style-type: none"> <li>Оценка выполнения практического задания(работы)</li> <li>Подготовка и выступление с докладом, сообщением, презентацией</li> <li>Решение ситуационной задачи....</li> </ul>

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.

выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.