САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ «КОЛЛЕДЖ «КРАСНОСЕЛЬСКИЙ»

РАССМОТРЕНО И ПРИНЯТО
на заседании Педагогического Совета
СПб ГБПОУ «Колледж «Красносельский»

Протокол №__2__ от__10.03__ 2023 г.

УТВЕРЖДАЮ
Директор СПб ГБПОУ
«Колледж «Красносельский»
_____Г.И. Софина
«__10__» __03__ 2023 г.
Приказ №__24-ОД__ от __10.03__ 2023 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.04 Основы алгоритмизации и программирования

основной профессиональной образовательной программы среднего профессионального образования по специальности

09.02.07 Информационные системы и программирование

СОГЛАСОВАНО:
Генеральный директор
ООО «АРКТИКА СПБ
_____
Н.А. Пузанов
_____

Санкт-Петербург
2023 г.

СОДЕРЖАНИЕ

1. **ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ** **4**

2. **СТРУКТУРА УЧЕБНОЙ ДИСЦИПЛИНЫ** **5**

3. **УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ** **10**

4. **КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ** **11**

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

1.1.Область применения рабочей программы

Рабочая программа учебной дисциплины является частью примерной основной образовательной программы по специальности среднего профессионального образования 09.02.07 Информационные системы и программирование в соответствии с ФГОС СПО 09.02.07 Информационные системы и программирование.

1.2. Место дисциплины в структуре основной профессиональной образовательной программы

Учебная дисциплина «Основы алгоритмизации и программирования» принадлежит к общепрофессиональному циклу (ОП.00)

1.3. 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	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.	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

1.4. Количество часов на освоение программы дисциплины: учебной нагрузки студента 152 часа, в том числе:

- обязательной аудиторной учебной нагрузки студента во взаимодействии с преподавателем 150 часов;
- самостоятельной работы студента 2 часа.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	**Объем часов**
Объем образовательной программы	**152**
учебная нагрузка во взаимодействии с преподавателем	**150**
Самостоятельная работа	**2**
в том числе:	
теоретическое обучение	8
Лабораторные и практические занятия	134
консультация	2
экзамен	6
Самостоятельная работа	2
Промежуточная аттестация проводится в форме экзамена	

Наименование разделов и тем	*Содержание учебного материала и формы организации деятельности обучающихся*	**Коды компетенций, формированию которых способствует элемент программы**	*Объем часов*
1	*2*	*3*	*4*
Раздел 1. Введение в программирование			
Тема 1.1. Языки программирования. Типы данных	**Содержание учебного материала**		
	1. 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. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5, ПК 2.4, 2.5	*2*
	Практические занятия		10
	Знакомство со средой программирования		
	Самостоятельная работа обучающихся Эволюция языков программирования Классификация программного обеспечения ПК Типы данных, определяемые программистом Перечисляемый и интервальный типы		1
Раздел 2. Программирование на алгоритмическом языке			
Тема 2.1. Операторы языка программирования	**Содержание учебного материала**	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5, ПК 2.4, 2.5	2
	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 работы		

	Составление программ линейной структуры		40
	Составление программ разветвляющейся структуры		
	Цикл с предусловием		
	Цикл с постусловием		
	Цикл с параметром.		
	Обработка одномерных массивов		
	Обработка двумерных массивов		
	Работа со строками		
	Работа с данными типа множество		
	Файлы последовательного доступа		
	Типизированные файлы		
	Нетипизированные файлы		
	Решение задач		
Раздел 3. Подпрограммы			
Тема 3.1. Процедуры и функции **Модульное программирование**	**Содержание учебного материала**	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5, ПК 2.4, 2.5	2
	Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций. Рекурсия. Программирование рекурсивных алгоритмов Модульное программирование. Понятие модуля. Структура модуля. Компоновка программы. Стандартные модули		
	Лабораторные работы		
	Организация процедур		24
	Использование процедур		
	Организация функций		
	Применение рекурсивных функций		
	Программирование модуля		
Раздел 4. Программирование в объектно-ориентированной среде			
Тема 4.1 Основные принципы объектно-ориентированного программирования (ООП). **Интегрированная среда разработчика** **Визуальное событийно-**	**Содержание учебного материала**	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5,	2
	История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Классы объектов. Компоненты и их свойства. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. Требования к аппаратным и программным средствам интегрированной среды разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.		

управляемое программирование	Компоненты и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.	ПК 2.4, 2.5	
	Практические занятия		
	Изучение интегрированной среды разработчика		
	Решение задач. Создание простого проекта		24
	Создание проекта с использованием различных компонентов		
Тема 4.2 Разработка оконного приложения	**Практические занятия**		
	1. Разработка функционального интерфейса приложения. Создание интерфейса приложения	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5, ПК 2.4, 2.5	36
	2. Разработка функциональной схемы работы приложения		
	3. Разработка игрового приложения		
	Разработка оконного приложения с несколькими формами		
	Разработка игрового приложения		
		Итого	**152**

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация программы дисциплины требует наличия учебного кабинета «Программирование»
Оборудование учебного кабинета:
- Автоматизированные рабочие места на 15 обучающихся (Процессор не ниже Core i3, оперативная память объемом не менее 4 Гб;)
- Автоматизированное рабочее место преподавателя (Процессор не ниже Core i3, оперативная память объемом не менее 4 Гб;)

Технические средства обучения:
- Проектор и экран;
- Маркерная доска;
- Программное обеспечение общего и профессионального назначения, в т.ч. интегрированная среда разработчика IDE Python 3.

3.2. Информационное обеспечение обучения

Основные источники:

1. Федоров, Д. Ю. Программирование на языке высокого уровня python: учебное пособие для среднего профессионального образования / Д. Ю. Федоров. Москва: Издательство Юрайт, 2019. 126 с.
2. Семакин И.Г., Шестаков А.П. Основы алгоритмизации и программирования. Практикум. М.: ОИЦ «Академия», 2017. 144 с.

Дополнительные источники:

Профильные периодические издания:

Информатика

Информатика и образование

Интернет-источники: https://dfedorov.spb.ru/python3/ https://github.com/dm-fedorov/python3book

http://www.intuit.ru http://habrahabr.ru/blogs/programming

http://pythontutor.ru/

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий и лабораторных работ, тестирования, а также выполнения студентами индивидуальных заданий, проектов, исследований.

Результаты обучения	Критерии оценки	Формы и методы оценки
Перечень умений, осваиваемых в рамках дисциплины: • 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.	«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко. «Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками. «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 выполненных заданий содержат ошибки. «Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.	**Примеры форм и методов контроля и оценки** • Компьютерное тестирование на знание терминологии по теме; • Тестирование • Контрольная работа • Самостоятельная работа • Защита реферата • Семинар • Защита курсовой работы (проекта) • Выполнение проекта; • Наблюдение за выполнением практического задания. (деятельностью студента) • Оценка выполнения практического задания(работы) • Подготовка и выступление с докладом, сообщением, презентацией • Решение ситуационной задачи
Перечень знаний, осваиваемых в рамках дисциплины: • 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.		