Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новгородский государственный университет имени Ярослава Мудрого»
Центр дополнительного образования детей
«Дом научной коллаборации имени С.В. Ковалевской»

УТВЕРЖДАЮ
Проректор по ОД
_____ Ю.В.Данейкин
«_22_» _сентября_ 20_23_ г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА

Программирование на Python

Лицензия Серия 90Л01 №0009115 (Рег. № 2078) от 13.04.2016,
Выданная Рособрнадзором на срок - бессрочно

СОГЛАСОВАНО:

Директор ЦДО

_____ Е.И. Белова
«_20_» _сентября_ 2023 г.

Начальник ОРК

_____ Н.Н. Ермишкина
«_20_» _сентября_ 2023 г.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 5cd5e7f80e776f2b786f828090289cd2
Владелец: Данейкин Юрий Викторович
Действителен: с 02.06.2023 до 25.08.2024
НОВГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ЯРОСЛАВА МУДРОГО

РАЗРАБОТАЛ:

Педагог дополнительного образования
ОДОП «Детский университет»
ЦДОД «ДНК им. С.В. Ковалевской»

_____ Е.М.Феденко
«_18_» _сентября_ 2023 г.

Директор ЦДОД «Дом научной коллаборации им. С.В. Ковалевской»

_____ А.В. Нестерчук
«_18_» _сентября_ 2023 г.

Великий Новгород – 2023

Раздел 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1. Актуальность программы

Актуальность дополнительной общеобразовательной программы "Программирование на Python" обосновывается неотложной потребностью в развитии информационной грамотности и компьютерной грамотности среди обучающихся, а также соответствием приоритетным направлениям современного образования. Программа направлена на удовлетворение растущего спроса общества на профессиональных программистов, разработчиков и инженеров информационных технологий. Она отвечает потребностям обучающихся и их родителей в овладении навыками программирования, а также социальному заказу, связанному с цифровой трансформацией.

Программа предоставляет педагогически целесообразное обучение, учитывая отличительные особенности Python как языка программирования, его уровень обучения. Она доступна на нескольких уровнях сложности, позволяя начать с основ и достичь углубленного уровня владения языком.

Цель:

Цель дополнительной общеобразовательной программы "Программирование на Python" заключается в следующем:

1. Удовлетворение потребностей обучающихся в развитии интеллектуальных и навыков программирования на Python.
2. Способствование развитию творческих способностей обучающихся в сфере информационных технологий.
3. Формирование культуры здорового и безопасного использования информационных технологий.
4. Подготовка обучающихся к приобретению, совершенствованию и освоению знаний и навыков в области программирования на Python в соответствии с направленностью, уровнем и содержанием программы.

Задачи:

Обучающие задачи:

1. Познакомить обучающихся с основами языка программирования Python.

2. Обучить обучающихся разрабатывать программы на Python, включая основные концепции, синтаксис и структуры данных.
3. Расширить знания обучающихся о возможностях использования Python в создании приложений и решении задач.
4. Способствовать развитию аналитического и алгоритмического мышления при решении задач программирования.

Воспитательные задачи:

1. Сформировать у обучающихся уважение к логическому и системному мышлению.
2. Поддержать развитие самостоятельности, творческого мышления и критического анализа.
3. Содействовать формированию ответственного и этичного использования информационных технологий.

Развивающие задачи:

1. Способствовать развитию навыков решения практических задач и применения информационных технологий в реальных ситуациях.
2. Поддерживать развитие коммуникативных навыков и умения работать в команде.
3. Содействовать развитию технической грамотности и общей компьютерной грамотности обучающихся.

Планируемые результаты:

В результате обучения обучающиеся будут:

Знать:

1. Основы синтаксиса языка программирования Python.
2. Основные структуры данных и алгоритмы программирования.
3. Основные библиотеки и инструменты для разработки на Python.
4. Правила оформления и структурирования программного кода.

Уметь:

1. Создавать программы на Python для решения разнообразных задач.
2. Анализировать и оптимизировать программный код.

3. Работать с внешними библиотеками и ресурсами.
4. Решать практические задачи с использованием языка Python.

Владеть:

1. Навыками самостоятельной разработки и отладки программ.
2. Умением работать с командой и коллективно решать задачи.
3. Навыками эффективного поиска информации и решения проблем.

Категория обучающихся:

Дополнительная общеобразовательная программа "Программирование на Python" предназначена для обучающихся среднего и старшего школьного возраста (12-18 лет), которые интересуются программированием и информационными технологиями.

Форма обучения:

Очная

Режим занятий:

1 занятие в неделю по 2 ак.час.

Трудоемкость программы:

Трудоемкость программы составляет 20 часов

Раздел 2. СОДЕРЖАНИЕ ПРОГРАММЫ

2.1. Учебный план

№ п/п	Наименование разделов (модулей) и тем	Аудиторные учебные занятия, учебные работы (АЧ)			Внеаудиторная самостоятельная работа	Формы контроля	Трудоемкость
		всего	теоретические занятия	практические занятия			
		20	**10**	**10**			
1	Основы Python, переменные, операторы	2	1	1		Тест	
2	Структуры данных: списки, кортежи, словари	4	2	2		Тест	

3	Управляющие конструкции: условные операторы, циклы	2	1	1		Тест	
4	Функции и модули в Python	2	1	1		Тест	
5	Работа с файлами	2	1	1		Тест	
6	Дополнительные практические задачи	2	1	1		Задачи	
7	Объектно-ориентированное программирование (ООП) в Python	4	2	2		Тест	
8	Наследование и полиморфизм	2	1	1		Тест	
	Итого	20					20

2.3. (или 2.2.) Учебная программа

Содержание программы отражается через краткое описание тем программы с указанием теоретических и практических видов занятий, объема часов, в соответствии с учебным планом. Содержание реализуемой дополнительной общеобразовательной программы должно быть направлено на достижение целей программы, планируемых результатов ее освоения.

№ п/п	**Виды учебных занятий, учебных работ, объем в часах**	**Содержание**
Раздел 1		
Тема 1.1	Лекция 1 час	Введение в основы Python: синтаксис, переменные, операторы
	Практическое занятия 1 час	Практические упражнения и задачи по основам Python
Тема 1.2	Лекция 2 часа	Структуры данных: списки, кортежи, словари
	Практическое занятия 2 часа	Практические упражнения и задачи по работе со структурами данных
Тема 1.3	Лекция 1 час	Управляющие конструкции: условные операторы, циклы
	Практическое занятия 1 час	Практические задачи на условные операторы и циклы
Тема 1.4	Лекция 1 час	Функции и модули в Python

	Практическое занятия 1 час	Практические задачи по созданию функций и использованию модулей
Тема 1.5	Лекция 1 час	Работа с файлами
	Практическое занятия 1 час	Практические задачи по работе с файлами
Тема 1.6	Лекция 1 час	Дополнительные практические задачи
	Практическое занятия 1 час	Дополнительные упражнения и задачи для практики
Тема 1.7	Лекция 2 часа	Объектно-ориентированное программирование (ООП) в Python
	Практическое занятия 2 часа	Практические задачи по ООП
Тема 1.8	Лекция 1 час	Наследование и полиморфизм
	Практическое занятия 1 час	Практические задачи по наследованию и полиморфизму

Раздел 3. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

3.1. Литература

Основная:

Буйначев С. К., Боклаг Н. Ю. "Основы программирования на языке Python". Издательство Урал. ун-та, 2014 год

Дополнительная:

Простой Python. Современный стиль программирования. 2-е изд. — СПб.: Питер, 2021. — 592 с.: ил. — (Серия «Бестселлеры O'Reilly»)

Интернет-ресурсы:

Задания для начинающих изучать Python pythoninfo.ru.

3.2. Материально-технические условия реализации программы

Занятия проводятся на базе ФГБОУ ВО «Новгородского государственного университета имени Ярослава Мудрого» в учебном компьютерном классе ЦДОД «ДНК им. С.В. Ковалевской»

Материалы, необходимые для осуществления образовательного процесса:

аудитория для лекций и практических занятий на 20-30 мест, компьютер и проектор преподавателю для подготовки учебно-методических материалов. Компьютеры для учеников, обеспечивающие доступ к электронным ресурсам и выполнение заданий.

Педагогические условия:

1. Требования к участникам программы:

Базовые навыки работы с компьютером.

Желание изучать новые концепции и умения.

2. Уровень слушателей:

Программа предназначена для начинающих, не имеющих опыта в программировании или настройке ПО.

Рекомендуется базовое понимание работы компьютера и его основных функций.

3. Другие условия:

Наличие персонального компьютера с установленным программным обеспечением для обучения Python.

Желательно наличие доступа к интернету для дополнительных материалов и обучающих ресурсов.

4. Ожидаемые результаты:

Освоение основ программирования на Python.

Способность создавать простые программы для решения задач в повседневной жизни.

Понимание базовых концепций алгоритмов и структур данных.

5. Формат обучения:

Интерактивные лекции, объясняющие основы Python и программирования.

Практические упражнения для закрепления знаний.

Мини-проекты для применения изученных концепций на практике.

6. Продолжительность программы:

8-10 недель, включая финальный проект.

Особенности освоения программы инвалидами и лицами с ограниченными возможностями здоровья:

Для инвалидов и лиц с ограниченными возможностями учебный процесс осуществляется в соответствии с Положением «Об организации сопровождения инвалидов и лиц с ограниченными возможностями здоровья в федеральном государственном бюджетном образовательным учреждением высшего образования «Новгородский государственный университет имени Ярослава Мудрого» от 30.03.2021 г.

Раздел 4. ФОРМЫ АТТЕСТАЦИИ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Формы текущего контроля:

Оценка активности на уроках и участие в дискуссиях.

Практические задания и упражнения во время занятий.

Формы промежуточной аттестации:

Промежуточные тесты по каждой из основных тем программы.

Практические проекты, оцениваемые в соответствии с критериями, изученными в ходе обучения.

Итоговая аттестация:

Завершающий тест, охватывающий ключевые аспекты изученного материала.

Самооценка итоговых достижений в рамках обучения.

5. СОСТАВИТЕЛИ ПРОГРАММЫ

Феденко Ефим Михайлович, Педагог дополнительного образования ОДОП «Детский университет» ЦДОД «ДНК им. С.В. Ковалевской»

ФОРМЫ АТТЕСТАЦИИ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Формы текущего контроля:

Основы Python, переменные, операторы:

1. Напишите программу, которая запрашивает у пользователя его имя и возраст, а затем выводит приветственное сообщение с этими данными.
2. Напишите программу, которая запрашивает у пользователя длину и ширину прямоугольника и выводит его площадь.
3. Напишите программу, которая просит пользователя ввести радиус круга и выводит его площадь.
4. Напишите программу, которая проверяет, является ли введенный пользователем год високосным.
5. Напишите программу, которая запрашивает у пользователя два числа и выводит их произведение.

Структуры данных: списки, кортежи, словари:

1. Создайте список чисел и напишите программу для вычисления их суммы.
2. Создайте кортеж с набором цветов радуги и напишите программу, которая выводит каждый цвет на новой строке.
3. Создайте словарь, представляющий информацию о вашем любимом фильме, и напишите программу для вывода этой информации.
4. Создайте список из имен ваших друзей и добавьте новое имя в этот список.
5. Создайте список чисел и напишите программу для нахождения наибольшего и наименьшего числа в этом списке.

Управляющие конструкции: условные операторы, циклы:

1. Напишите программу, которая определяет, является ли введенное пользователем число положительным, отрицательным или нулем.
2. Напишите программу, которая запрашивает у пользователя число и проверяет, принадлежит ли оно к диапазону от 50 до 100.
3. Напишите программу, которая выводит все числа от 1 до 10, кроме числа 5.
4. Напишите программу, которая запрашивает у пользователя число и выводит таблицу умножения для этого числа.

5. Напишите программу, которая запрашивает у пользователя строку и выводит ее в обратном порядке.

Функции и модули в Python:

1. Напишите функцию для вычисления факториала числа.
2. Напишите программу, которая импортирует модуль math и использует его функцию для вычисления косинуса угла.
3. Создайте функцию, которая принимает список чисел и возвращает только четные числа из этого списка.
4. Напишите программу, которая использует встроенную функцию map для умножения каждого элемента списка на 2.
5. Создайте функцию, которая принимает строку в качестве аргумента и возвращает количество гласных букв в этой строке.

Работа с файлами:

1. Напишите программу, которая создает новый файл и записывает в него текст.
2. Напишите программу, которая открывает существующий файл и выводит его содержимое.
3. Напишите программу, которая копирует содержимое одного файла в другой файл.
4. Напишите программу, которая удаляет существующий файл.
5. Напишите программу, которая переименовывает существующий файл.

Дополнительные практические задачи:

1. Напишите программу для нахождения среднего арифметического списка чисел.
2. Напишите программу для сортировки списка чисел в порядке возрастания.
3. Напишите программу, которая находит все простые числа в заданном диапазоне.
4. Напишите программу, которая находит сумму всех чисел Фибоначчи до заданного предела.
5. Напишите программу для вычисления наименьшего общего кратного двух чисел.

Объектно-ориентированное программирование (ООП) в Python:

1. Создайте класс "Книга" с атрибутами, такими как название, автор и год издания, и методом для вывода информации о книге.
2. Реализуйте класс "Автомобиль" с атрибутами, такими как марка, модель и цвет, и методами для изменения цвета автомобиля.
3. Создайте класс "Фрукт" с атрибутами, такими как тип и цена, и методом для вывода информации о фрукте.
4. Реализуйте класс "Сотрудник" с атрибутами, такими как имя, возраст и зарплата, и методами для изменения и вывода информации о сотруднике.
5. Создайте класс "Животное" с атрибутами, такими как вид и возраст, и методами для изменения и вывода информации о животном.

Наследование и полиморфизм:

1. Создайте класс "Фигура" с методом для вычисления площади и периметра, а затем создайте подклассы, такие как "Круг" и "Прямоугольник", с переопределенными методами.
2. Реализуйте класс "Транспортное средство" с методами для движения и остановки, а затем создайте подклассы, такие как "Автомобиль" и "Велосипед", с собственными методами движения.
3. Создайте класс "Фигура" с методом для отображения фигуры на экране, а затем создайте подклассы, такие как "Треугольник" и "Квадрат", с собственными методами отображения.
4. Реализуйте класс "Фрукт" с методами для определения вкуса и цвета, а затем создайте подклассы, такие как "Яблоко" и "Апельсин", с собственными методами для определения специфических характеристик.
5. Создайте класс "Животное" с методами для издания звуков и движения, а затем создайте подклассы, такие как "Кошка" и "Собака", с собственными методами для специфического поведения.

Формы промежуточной аттестации:

Тест по теме "Основы Python, переменные, операторы":

1. Что выведет следующий код: print(3 + 2 * 2)?

a) 7

b) 10

c) 8

d) 12

2. Какие типы данных поддерживает Python?

a) int, float, string

b) int, float, complex

c) float, complex, list

d) list, tuple, dict

3. Что делает оператор % в Python?

a) Возвращает остаток от деления

b) Возвращает целую часть от деления

c) Выполняет степенную операцию

d) Преобразует число в отрицательное

4. Какая из следующих переменных будет являться корректной переменной Python?

a) my-var

b) 2var

c) my_var

d) my var

5. Какой из перечисленных операторов Python используется для сравнения значений?

a) ==

b) %

c) +=

d) **

Тест по теме "Структуры данных: списки, кортежи, словари":

1. Как добавить элемент в конец списка в Python?

a) list.add(element)

b) list.push(element)

c) list.append(element)

d) list.insert(element)

2. Что произойдет, если вы попытаетесь изменить элемент кортежа в Python?

a) Вызовется ошибка

b) Элемент будет успешно изменен

c) Кортеж будет конвертирован в список

d) Элемент будет удален из кортежа

3. Какие методы используются для удаления элемента из списка в Python?

a) remove() и delete()

b) erase() и discard()

c) pop() и clear()

d) remove() и pop()

4. Что такое ключ в словаре Python?

a) Значение, связанное с элементом

b) Позиция элемента в словаре

c) Индекс элемента в словаре

d) Уникальный идентификатор для доступа к элементу

5. Какие из перечисленных структур данных в Python являются изменяемыми?

a) Списки

b) Кортежи

c) Словари

d) Списки и кортежи

Тест по теме "Управляющие конструкции: условные операторы, циклы":

1. Какая конструкция используется для проверки условия в Python?

a) if-else

b) for

c) while

d) do-while

2. Что делает оператор in в Python?

a) Проверяет принадлежность элемента к списку

b) Выполняет математическую операцию

c) Соединяет строки

d) Создает новую переменную

3. Какой цикл будет выполняться, пока заданное условие истинно?

a) for

b) while

c) do-while

d) repeat-until

4. Какие операторы используются для логических операций в Python?

a) && и ||

b) and и or

c) AND и OR d) & и |

5. Как прервать выполнение цикла и перейти к следующей итерации в Python?

a) break

b) skip

c) continue

d) pass

Тест по теме "Функции и модули в Python":

1. Что такое функция в Python?

a) Набор инструкций, выполняющих определенную задачу и возвращающих значение

b) Переменная, хранящая значение

c) Условие, определяющее действие

d) Ключевое слово для создания классов

2. Что такое модуль в Python?

a) Файл, содержащий набор функций и инструкций

b) Переменная, хранящая список значений

c) Массив, содержащий элементы

d) Ключевое слово для создания циклов

3. Какой оператор используется для возврата значения из функции в Python?

a) break

b) continue

c) return

d) yield

4. Какой метод используется для импорта модуля в Python?

a) use

b) require

c) import

d) include

5. Какие из перечисленных функций являются встроенными функциями Python?

a) print() и input()

b) read() и write()

c) load() и save()

d) add() и remove()

Тест по теме "Работа с файлами":

1. Какой режим открытия файла в Python позволяет только чтение?

a) r

b) w

c) x

d) a

2. Какой метод используется для чтения содержимого файла в Python?

a) read()

b) load()

c) fetch()

d) open()

3. Какой метод используется для записи в файл в Python?

a) write()

b) save()

c) store()

d) add()

4. Что делает метод close() при работе с файлами в Python?

a) Закрывает файл

b) Удаляет файл

c) Переименовывает файл

d) Копирует файл

5. Какой оператор используется для создания нового файла в Python?

a) new

b) create

c) make

d) open

Тест по теме "Объектно-ориентированное программирование (ООП) в Python":

1. Что такое класс в объектно-ориентированном программировании?

a) Шаблон, определяющий структуру и поведение объекта

b) Объект, созданный на основе класса

c) Метод, выполняющий действие

d) Свойство, хранящее значение

2. Что такое объект в объектно-ориентированном программировании?

a) Экземпляр класса

b) Функция, выполняющая определенное действие

c) Список, содержащий элементы

d) Переменная, хранящая значение

3. Что такое метод в объектно-ориентированном программировании?

a) Функция, принадлежащая классу

b) Переменная, хранящая значение

c) Условие, определяющее действие

d) Свойство, хранящее значение

Тест по теме " Наследование и полиморфизм в Python":

1. Что такое наследование в объектно-ориентированном программировании?

a) Процесс получения свойств и методов от родительского класса

b) Сокрытие свойств и методов в подклассе

c) Создание нового класса на основе существующего

d) Удаление свойств и методов из родительского класса

2. Что такое полиморфизм в объектно-ориентированном программировании?

a) Возможность использования разных типов данных в одной операции

b) Возможность создания нескольких объектов одного класса

c) Возможность изменения типа данных переменной

d) Возможность изменения структуры класса после создания объекта.

Итоговый тест:

1. Что такое метод в объектно-ориентированном программировании?

a) Функция, принадлежащая классу

b) Переменная, хранящая значение

c) Условие, определяющее действие

d) Свойство, хранящее значение

2. Какой режим открытия файла в Python позволяет только чтение?

a) r

b) w

c) x

d) a

3. Что такое полиморфизм в объектно-ориентированном программировании?

a) Возможность использования разных типов данных в одной операции

b) Возможность создания нескольких объектов одного класса

c) Возможность изменения типа данных переменной

d) Возможность изменения структуры класса после создания объекта.

4. Какой метод используется для импорта модуля в Python?

a) use

b) require

c) import

d) include

5. Какая конструкция используется для проверки условия в Python?

a) if-else

b) for

c) while

d) do-while

6. Как добавить элемент в конец списка в Python?

a) list.add(element)

b) list.push(element)

c) list.append(element)

d) list.insert(element)

7. Какие типы данных поддерживает Python?

a) int, float, string

b) int, float, complex

c) float, complex, list

d) list, tuple, dict

Критерии оценки теста:

- За каждый правильный ответ выставляется 1 балл.
- За неправильный ответ или пропуск вопроса выставляется 0 баллов.