**Тема: ИИ и программировании: основы программирования на Python.**

**Иван:** на прошлом занятии я узнал, что такое переменные и типы данных в Python. То, что искусственный интеллект способен за меня написать программу на языке Python, это я понял. Но я хочу попробовать кодить сам. Я же могу, например, написать программу, которая будет мне решать задачи по математике?

Например, программу, которая будет выводить таблицу умножения до пяти.

**Нейросеточка:** Для этой программы тебе нужно использовать вложенные циклы.

Вложенный цикл — это цикл, который находится внутри других циклов.

ПАУЗА

Один цикл - внешний, будет перебирать значения первого числа, пусть это будет i, а внутренний, будет перебирать второй множитель, например переменная j.

**Иван**: Понял, тогда ввожу так:

*Иван Открывает VSC вводит код программы.*

*for i in range(1, 6):*

*for j in range(1, 6):*

*print(i \* j)*

*запускает программу, она выводит в столбик значения.*

*1*

*2*

*3*

*4*

*5*

*2*

*4*

*6*

*8*

*10*

*3*

*6*

*9*

*12*

*15*

*4*

*8*

*12*

*16*

*20*

*5*

*10*

*15*

*20*

*25*

**Иван:** но я хочу, чтобы выводились оба множителя и их произведение на экран.

**Нейросеточка:** тогда тебе нужно немного изменить код. Написать сначала произведение переменных i , j а затем уже равно и результат.

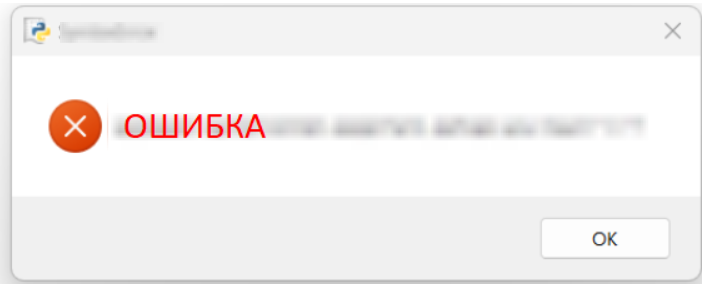
**Иван на экране выводит**

*for i in range(1, 6):*

*for j in range(1, 6):*

*print(i \* j =i \* j )*

*Запускает программу. На экране окно ошибки*

**

**Иван:** Не работает. И как исправить?

**Нейросеточка.** Попроси искусственный интеллект тебе помочь. Опиши условие задачи, для которой пишешь программу, затем составь запрос на исправление ошибки.

**Рассказчик:** Промпт: Я пишу программу, которая выводит таблицу умножения до 5. Хочу чтобы на экран выводились два множителя, а затем их произведение. Написал программу

for i in range(1, 6):

for j in range(1, 6):

print(i \* j =i \* j )

Но в ней ошибка. Исправь ошибку и объясни, почему мой код не работает

Результат:

В вашем коде есть ошибка в операторе print. Вы не указали разделители между множителями и их произведением, а также забыли поставить пробел после знака равенства. Правильный код будет выглядеть так:



for i in range(1, 6):

for j in range(1, 6):

print(i, "\*", j, "=", i \* j)

В этом случае программа будет выводить на экран два множителя, знак умножения и их произведение.

Иван: Скопирую программу и запущу!

*На экране он в VSC вставляет код.*

*for i in range(1, 6):*

*for j in range(1, 6):*

*print(i, "\*", j, "=", i \* j)*

*В выводе*

*1 \* 1 = 1*

*1 \* 2 = 2*

*1 \* 3 = 3*

*1 \* 4 = 4*

*1 \* 5 = 5*

*2 \* 1 = 2*

*2 \* 2 = 4*

*2 \* 3 = 6*

*2 \* 4 = 8*

*2 \* 5 = 10*

*3 \* 1 = 3*

*3 \* 2 = 6*

*3 \* 3 = 9*

*3 \* 4 = 12*

*3 \* 5 = 15*

*4 \* 1 = 4*

*4 \* 2 = 8*

*4 \* 3 = 12*

*4 \* 4 = 16*

*4 \* 5 = 20*

*5 \* 1 = 5*

*5 \* 2 = 10*

*5 \* 3 = 15*

*5 \* 4 = 20*

*5 \* 5 = 25*

**Иван:** Работает!

**Нейросеточка:** Да, как видишь, Искусственный интеллект умеет находить и исправлять ошибки в коде.

Рассказчик.

Задание: Создайте промпт для ЯндекcGPT, который поможет составить таблицу степеней числа 2.

Пауза.

**Иван:** на прошлом занятии я составил план изучения Python. Давай продолжим? Что у нас там дальше по списку?

*На экране появляется список, из прошлого урока:*

* *Основы Python:*
  + *Установка Python и среды разработки (IDE).*
  + *Типы данных, переменные, операторы.*
  + *Условные операторы и циклы.*
  + *Функции и модули.*

**Нейросеточка:** нужно познакомиться с функциями и модулями**.**

**Иван:** Отлично! Расскажешь, что такое функции?

**Нейросеточка: Функция** — это блок кода, который выполняет определенное действие или вычисление, имеет имя и может быть вызван из другого места программы для повторного использования**.**

ты можешь вызывать функцию в других частях кода, когда тебе нужно совершить это действие, ты просто пишешь название функции, и программа это действие выполняет.

Делать это можно столько раз, сколько необходимо.

Функции нужны для того, чтобы код был проще и короче, чтобы его можно было адаптировать для разных платформ и легко его исправлять

**Рассказчик:**

Тело функции — это набор инструкций, объединенных для общей задачи. В теле функции следует писать код, отвечающий за всю работу функции.

Аргумент (параметр) функции —это переменная, которой присваивается конкретное значение, передаваемое в функцию при ее вызове.Аргументы пишутся в скобках после названия функции.

В Python есть два типа функций, которые нужны для разного рода задач — встроенные и пользовательские.

Встроенных функций в языке Python очень много.

Встроенные функции — это заранее определенные функции, которые уже есть в Python и готовы к использованию. Они доступны сразу после запуска программы. Например,

print() - выводит текст на экран,

input() - считывает данные, которые вводит пользователь

int() - преобразует тип данных в целое число

float() - преобразует тип данных в вещественное число

Пользовательские функции — это блок кода, который выполняет определенную задачу и создается программистом для использования в программе.

def my\_function():

print("Привет, мир!")



Эта функция называется my\_function(), она ничего не принимает в качестве аргументов и не возвращает значение. При её вызове будет выведено сообщение «Привет, мир!».

Пример функции с аргументами и возвращаемым значением:

**def** **add**(x, y):

result = x + y

**return** result

Эта функция называется add() и принимает два аргумента x и y. Она складывает их значения и возвращает результат в виде числа. Когда мы вызываем эту функцию, например, add(5, 3), она вернёт результат 8.

Иван: интересно! И где мне это может пригодиться? Спрошу у яндекс GPT

Промпт: приведи пример ситуации из жизни, для которой я могу написать пользовательскую функцию на Python. Ответ предоставь в таком виде: сначала формулировка ситуации, а затем напиши программу, которая будет предназначена для этой ситуации.

Результат: вы хотите автоматизировать процесс определения стоимости покупки одного товара в магазине, учитывая скидку 10% на все товары дороже 500 рублей.

def calculate\_price(price):

if price > 500:

discount = price \* 0.1

return price - discount

else:

return price

# Ввод цены товара

price = float(input("Введите цену товара: "))

# Вызов функции и вывод результата

result = calculate\_price(price)

print("Стоимость товара с учётом скидки:", result)



Иван: Здорово получилось. А вот можно ли эту программу переписать без пользовательской функции?

Нейросеточка: Давай спросим об этом у ЯндексGPT.

Рассказчик: промпт:

Перепиши программу без пользовательской функции

def calculate\_price(price):

if price > 500:

discount = price \* 0.1

return price - discount

else:

return price

# Ввод цены товара

price = float(input("Введите цену товара: "))

# Вызов функции и вывод результата

result = calculate\_price(price)

print("Стоимость товара с учётом скидки:", result)

Результат: Конечно, вот программа на Python без пользовательской функции:

price = float(input("Введите цену товара: "))

if price > 500:

discount = price \* 0.1

result = price - discount

else:

result = price

print("Стоимость товара с учётом скидки:", result)

Иван: Понял, то есть программы будут работать и без функции.

Нейросеточка: конечно, но тогда код будет длиннее и менее читаемым.

Иван: Для чего нужны функции я понял. А вот модули? Это что такое? Думаю, об этом тоже может рассказать искусственный интеллект.

*На экране появляется список, из прошлого урока:*

* *Основы Python:*
  + *Установка Python и среды разработки (IDE).*
  + *Типы данных, переменные, операторы.*
  + *Условные операторы и циклы.*
  + *Функции и модули.*

**Рассказчик:**

Промпт: расскажи что такое модуль в Python. Привели примеры модулей, которые будут полезны школьнику.

Модуль в Python — это файл с кодом, который можно импортировать и использовать в других программах. Модули позволяют структурировать код, повторно использовать его в разных проектах и упрощают процесс разработки.

В Python есть стандартные модули, которые входят в состав интерпретатора, а также сторонние модули, доступные для установки через менеджер пакетов pip.

Вот несколько примеров модулей, которые могут быть полезны школьнику:

* math: содержит математические функции, такие как синус, косинус, логарифм, квадратный корень и т. д.
* random: позволяет генерировать случайные числа и выбирать элементы из списка случайным образом.
* datetime: предоставляет инструменты для работы с датами и временем.
* os: даёт доступ к функциям операционной системы, таким как создание файлов и каталогов, получение текущего каталога и т. п.

Чтобы использовать модуль в программе, нужно его импортировать с помощью команды import или from ... import. Например:

**import** math

x = math.sqrt(4) # вычисление квадратного корня из 4

print(x)

или

**from** math **import** sqrt

x = sqrt(4)

print(x)

**

Иван: Тааак… А я же могу попросить написать программу с помощью какого-нибудь модуля, например создание файл с помощью модуля os. Но я же не умею им пользоваться.

Нейросеточка: Есть еще один способ составления детального промпта.

***Рассказчик: Промпт:***

Тебе нужно принять роль Python начинающего программиста. Твоя главная цель - Написать оптимальный надежный код и объяснить его логику работы по следующему заданию. Используя модуль os создай файл с именем "Новый файл" в папке, где хранится сама программа.

Требования к твоему ответу:

- Пиши свой ответ по частям, и всегда указывай номер части;

- В первой части напиши краткое описание решения задачи, план. . На этом закончи свой первый ответ, и ожидай когда я одобрю твой план;

- Во второй части напиши структуру программы.

- В третьей части напиши реализацию кода. Если это часть будет очень большой, то ты можешь разделить реализацию когда на сколько угодно глав

- В последней части напиши вывод;

Требования к коду:

- Пиши комментарии в коде.



**Результат:**

*Для озвучки: Предлагаю не прочитывать полностью а только название частей.*

Часть 1. План решения задачи

Для создания файла с именем «Новый файл» в папке, где хранится программа, можно использовать модуль os. Вот план решения задачи:

* Импортировать модуль os.
* Получить путь к папке, в которой находится программа.
* Создать объект File для записи нового файла.
* Записать данные в новый файл.
* Закрыть файл.

Часть 2. Структура программы

Программа будет состоять из следующих блоков:

* импорт модуля os;
* получение пути к папке с программой;
* создание объекта File и запись данных в файл;
* закрытие файла.

Часть 3. Реализация кода

1. Импорт модуля os:



**import** os





2. Получение пути к папке с программой:



program\_folder = os.path.dirname(\_\_file\_\_)

Здесь file — это имя файла программы. Метод dirname() возвращает путь к папке.

3. Создание объекта File и запись данных в файл:



new\_file = open(os.path.join(program\_folder, 'Новый файл'), 'w')

new\_file.write('Данные для нового файла')

Метод join() объединяет путь к папке и имя файла. Параметр 'w' указывает на режим записи в файл.

4. Закрытие файла:



new\_file.close()

Часть 4. Вывод

После выполнения программы в папке с программой будет создан файл с именем «Новый файл». В файле будут записаны данные «Данные для нового файла».

Полный код программы:

# Часть 1. Импорт модуля os

**import** os

# Часть 2. Получение пути к папке с программой

program\_folder = os.path.dirname(\_\_file\_\_)

# Часть 3. Создание объекта File и запись данных в файл

new\_file = open(os.path.join(program\_folder, 'Новый файл'), 'w')

new\_file.write('Данные для нового файла')

# Часть 4. Закрытие файла

new\_file.close()



Иван: Классно! Скопирую и посмотрю, что получилось.

На экране Ваня запускает код

# Часть 1. Импорт модуля os

**import** os

# Часть 2. Получение пути к папке с программой

program\_folder = os.path.dirname(\_\_file\_\_)

# Часть 3. Создание объекта File и запись данных в файл

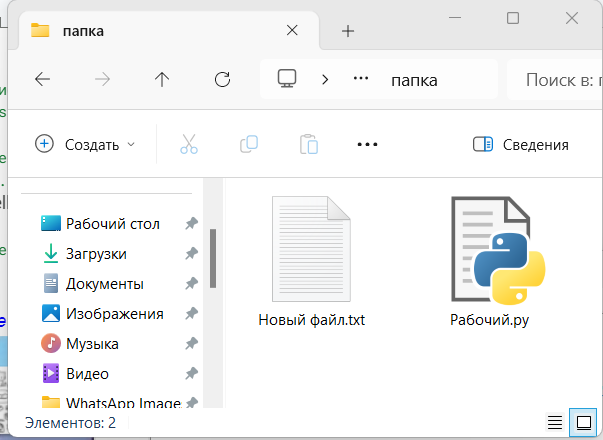
new\_file = open(os.path.join(program\_folder, 'Новый файл.txt'), 'w')

new\_file.write('Hello, world!')

# Часть 4. Закрытие файла

new\_file.close()

*После этого в окне папки появляется файлик который называется Новый файл*



Задание: Составьте промпт, в котором подробно попросите ЯндексGPT написать программу, которая использует модуль datetime для получения текущей даты и времени, а затем выводит их на экран в удобном формате.

Пауза.

**Иван:** Как интересно! Теперь я умею использовать искусственный интеллект для изучения программирования на Python.

**Рассказчик:** На этом занятии вы:

Узнали, что такое функции и модули в Python

Научились исправлять ошибку в программе с помощью искусственного интеллекта

Научились составлять детальный промпт, для написания качественного кода и описания к нему.

На сегодня всё! Увидимся на следующем занятии