

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №4
дисциплины «Программирование на Python»

Выполнил:
Горбунов Данила Евгеньевич
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

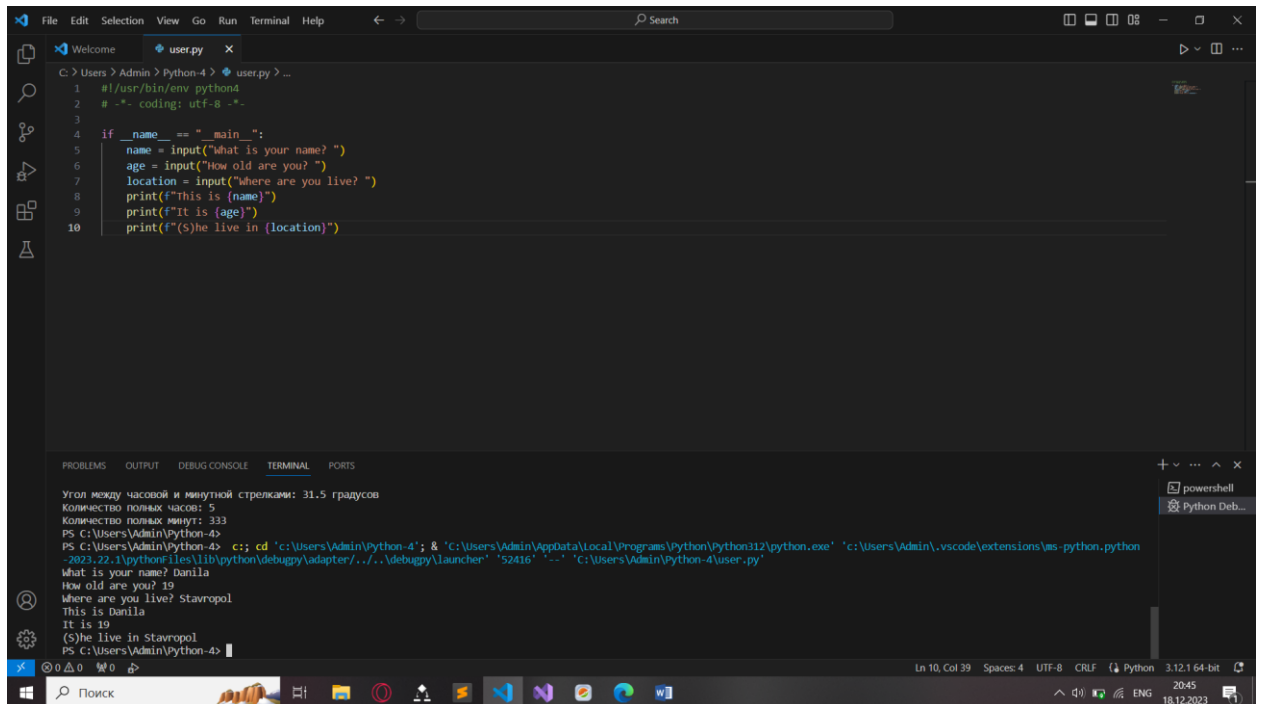
Ставрополь, 2023 г.

Тема: Основы языка Python.

Цель работы – исследование процесса установки и базовых возможностей языка Python версии 3.x.

Ход работы

1. Написал программу, которая запрашивает у пользователя его имя, возраст и место жительства, а после этого выводит данные, введенные пользователем.



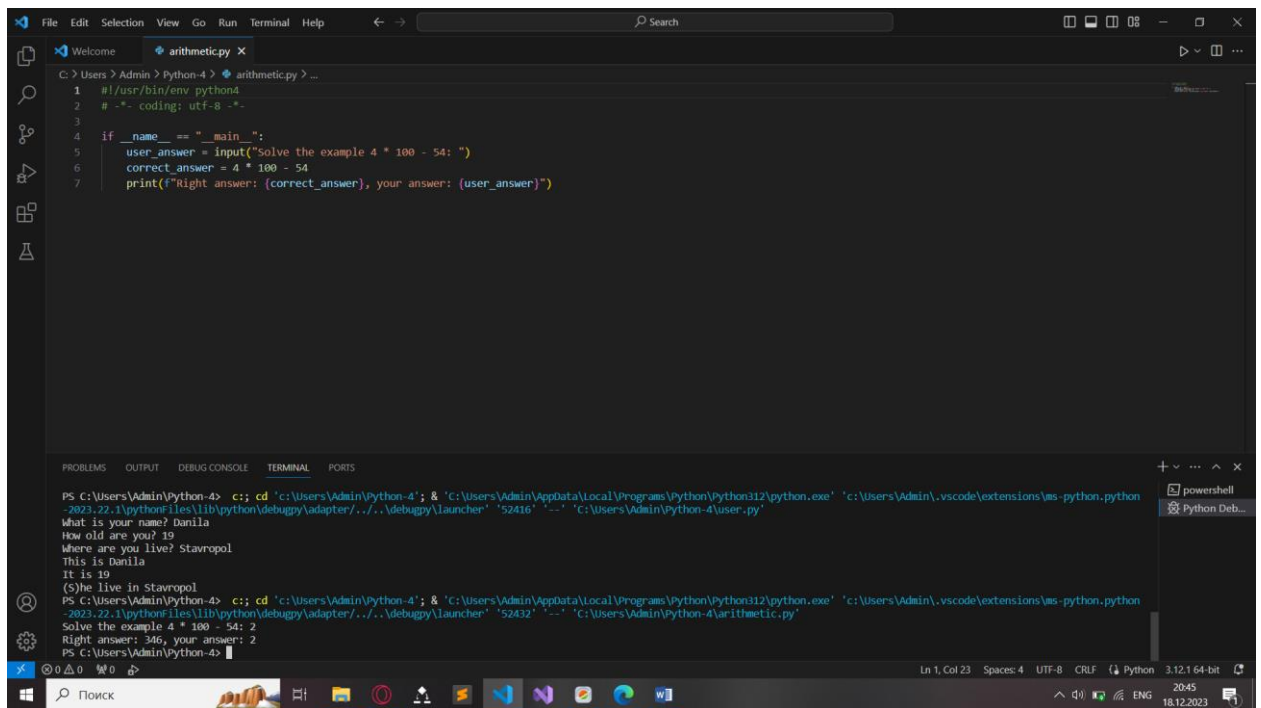
The screenshot shows a Visual Studio Code editor window with a Python file named 'user.py'. The code in the file is as follows:

```
1 #!/usr/bin/env python4
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     name = input("What is your name? ")
6     age = input("How old are you? ")
7     location = input("Where are you live? ")
8     print(f"This is {name}")
9     print(f"It is {age}")
10    print(f"(S)he live in {location}")
```

Below the editor, the 'TERMINAL' panel shows the execution of the script. The output is as follows:

```
Угол между часовой и минутной стрелками: 31.5 градусов
Количество полных часов: 5
Количество полных минут: 333
PS C:\Users\Admin\Python-4>
PS C:\Users\Admin\Python-4> cd "c:\Users\Admin\AppData\Local\Programs\Python\Python312\python.exe" "c:\Users\Admin\.vscode\extensions\ms-python.python-2023.22.1\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher" "52416" "--" "C:\Users\Admin\Python-4\user.py"
What is your name? Danila
How old are you? 19
Where are you live? Stavropol
This is Danila
It is 19
(S)he live in Stavropol
PS C:\Users\Admin\Python-4>
```

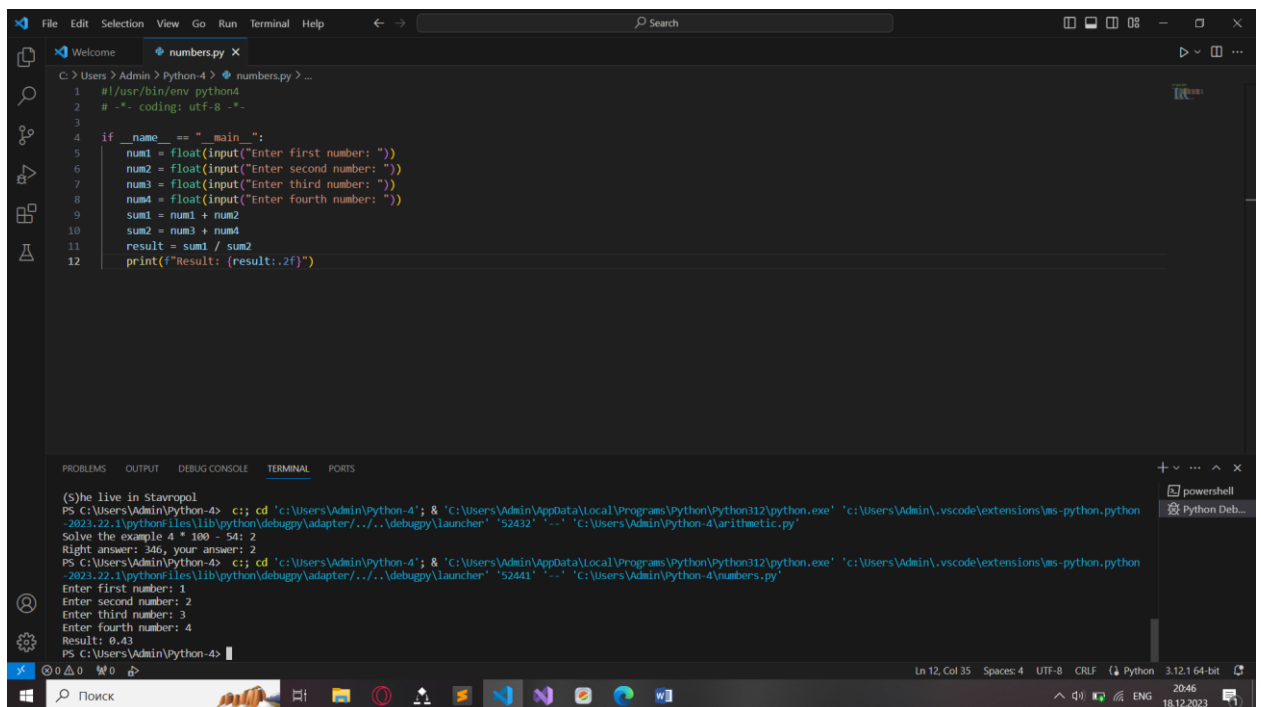
2. Написал программу, которая предлагает пользователю решить пример $4 \cdot 100 - 54$. Потом выводит на экран правильный ответ и ответ пользователя.



```
1 #!/usr/bin/env python4
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     user_answer = input("Solve the example 4 * 100 - 54: ")
6     correct_answer = 4 * 100 - 54
7     print(f"Right answer: {correct_answer}, your answer: {user_answer}")
```

```
PS C:\Users\Admin\Python-4> c:\cd 'c:\Users\Admin\Python-4'; & 'c:\Users\Admin\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\Admin\...
What is your name? Danila
How old are you? 19
Where are you live? Stavropol
This is Danila
It is 19
(S)he live in Stavropol
PS C:\Users\Admin\Python-4> c:\cd 'c:\Users\Admin\Python-4'; & 'c:\Users\Admin\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\Admin\...
Solve the example 4 * 100 - 54: 2
Right answer: 346, your answer: 2
PS C:\Users\Admin\Python-4>
```

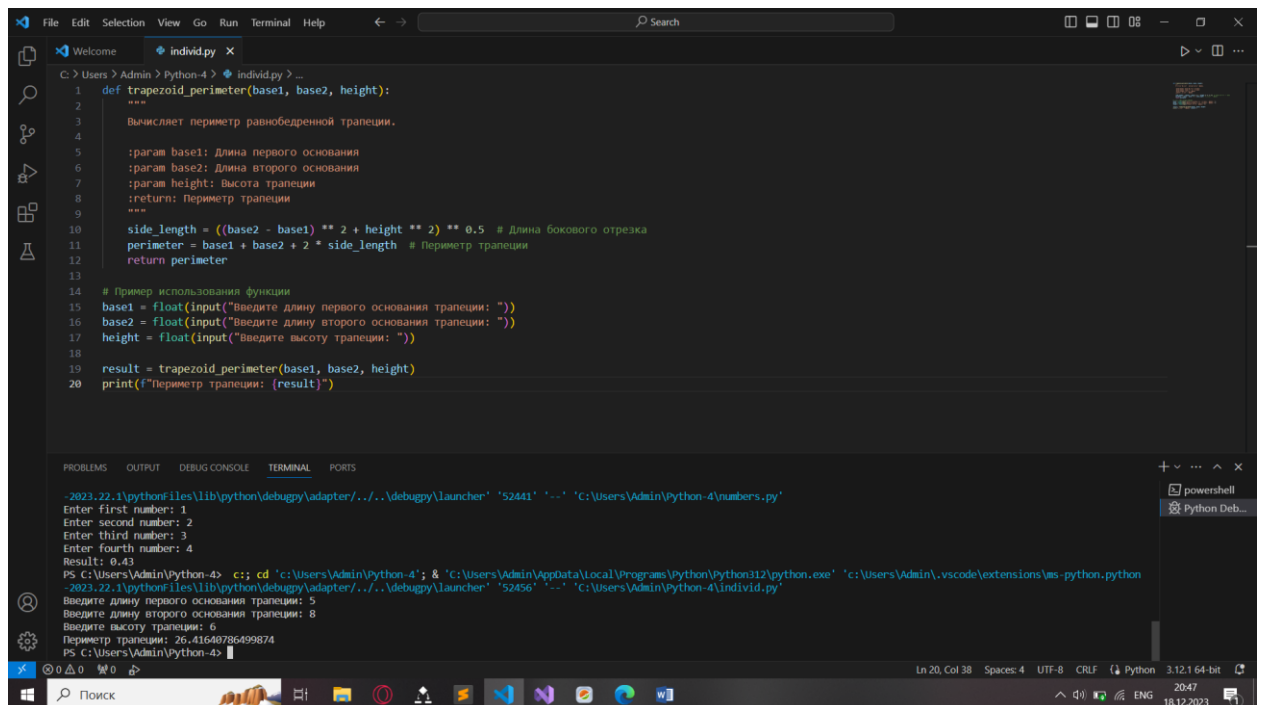
3. Написал программу, которая запрашивает у пользователя четыре числа (файл numbers.py), отдельно складывает первые два и отдельно вторые два, далее делит первую сумму на вторую и выводит результат на экран так, чтобы ответ содержал две цифры после запятой.



```
1 #!/usr/bin/env python4
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     num1 = float(input("Enter first number: "))
6     num2 = float(input("Enter second number: "))
7     num3 = float(input("Enter third number: "))
8     num4 = float(input("Enter fourth number: "))
9     sum1 = num1 + num2
10    sum2 = num3 + num4
11    result = sum1 / sum2
12    print(f"Result: {result:.2f}")
```

```
(S)he live in Stavropol
PS C:\Users\Admin\Python-4> c:\cd 'c:\Users\Admin\Python-4'; & 'c:\Users\Admin\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\Admin\...
Enter first number: 1
Enter second number: 2
Enter third number: 3
Enter fourth number: 4
Result: 0.43
PS C:\Users\Admin\Python-4>
```

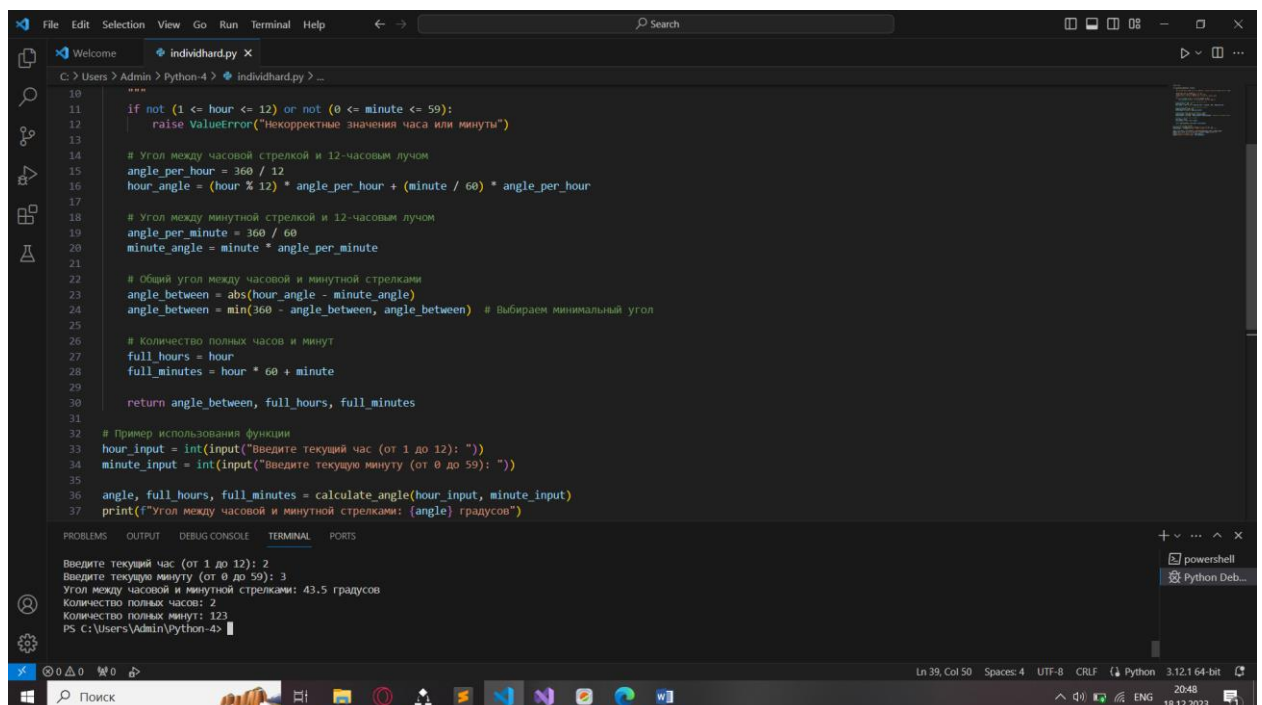
4. Написал программу для решения индивидуального задания. Вариант 7: даны основания и высота равнобедренной трапеции. Найти периметр трапеции.



```
def trapezoid_perimeter(base1, base2, height):  
    """  
    Вычисляет периметр равнобедренной трапеции.  
    :param base1: Длина первого основания  
    :param base2: Длина второго основания  
    :param height: Высота трапеции  
    :return: Периметр трапеции  
    """  
    side_length = ((base2 - base1) ** 2 + height ** 2) ** 0.5 # Длина бокового отрезка  
    perimeter = base1 + base2 + 2 * side_length # Периметр трапеции  
    return perimeter  
  
# Пример использования функции  
base1 = float(input("Введите длину первого основания трапеции: "))  
base2 = float(input("Введите длину второго основания трапеции: "))  
height = float(input("Введите высоту трапеции: "))  
result = trapezoid_perimeter(base1, base2, height)  
print(f"Периметр трапеции: {result}")
```

Terminal output:
-2023.22.1\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher '52441' '-...' 'C:\Users\Admin\Python-4\numbers.py'
Enter first number: 1
Enter second number: 2
Enter third number: 3
Enter fourth number: 4
Result: 0.43
PS C:\Users\Admin\Python-4> c:\c:\Users\Admin\Python-4\; & 'c:\Users\Admin\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\Admin\.vscode\extensions\ms-python.python-2023.22.1\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '52456' '-...' 'C:\Users\Admin\Python-4\indiv.py'
Введите длину первого основания трапеции: 5
Введите длину второго основания трапеции: 8
Введите высоту трапеции: 6
Периметр трапеции: 26.41640786499874
PS C:\Users\Admin\Python-4>

1. Написал программу для решения индивидуального задания повышенной сложности. Вариант 7: часовая стрелка образует угол α с лучом, проходящим через центр и через точку, соответствующую 12 часам на циферблате, $0 < \alpha < 2\pi$. Определить значение угла для минутной стрелки, а также количество полных часов и полных минут.



```
if not (1 <= hour <= 12) or not (0 <= minute <= 59):  
    raise ValueError("Некорректные значения часа или минуты")  
  
# Угол между часовой стрелкой и 12-часовым лучом  
angle_per_hour = 360 / 12  
hour_angle = (hour % 12) * angle_per_hour + (minute / 60) * angle_per_hour  
  
# Угол между минутной стрелкой и 12-часовым лучом  
angle_per_minute = 360 / 60  
minute_angle = minute * angle_per_minute  
  
# Общий угол между часовой и минутной стрелками  
angle_between = abs(hour_angle - minute_angle)  
angle_between = min(360 - angle_between, angle_between) # Выбираем минимальный угол  
  
# Количество полных часов и минут  
full_hours = hour  
full_minutes = hour * 60 + minute  
  
return angle_between, full_hours, full_minutes  
  
# Пример использования функции  
hour_input = int(input("Введите текущий час (от 1 до 12): "))  
minute_input = int(input("Введите текущую минуту (от 0 до 59): "))  
  
angle, full_hours, full_minutes = calculate_angle(hour_input, minute_input)  
print(f"Угол между часовой и минутной стрелками: {angle} градусов")
```

Terminal output:
Введите текущий час (от 1 до 12): 2
Введите текущую минуту (от 0 до 59): 3
Угол между часовой и минутной стрелками: 43.5 градусов
Количество полных часов: 2
Количество полных минут: 123
PS C:\Users\Admin\Python-4>

Контрольные вопросы

1. Опишите основные этапы установки Python в Windows и Linux.

В Windows: Скачать установочный файл Python с официального сайта (python.org). Запустить установщик и следовать инструкциям. Выбрать опцию "Add Python to PATH" (добавить Python в переменную среды PATH) для удобства использования Python из командной строки. Завершить процесс установки.

В Linux: Многие дистрибутивы Linux уже имеют Python предустановленным. В противном случае, можно установить его с помощью пакетного менеджера вашего дистрибутива, например, в Ubuntu: `sudo apt-get install python3`. После установки можно проверить версию с помощью команды `python3 --version`.

2. В чем отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта?

Anaconda - это дистрибуция Python, предназначенная для научных вычислений и анализа данных. Основное отличие заключается в том, что Anaconda включает в себя множество предустановленных библиотек и инструментов, таких как NumPy, Pandas, Matplotlib, Jupyter и многие другие, что делает ее идеальным выбором для работы в области анализа данных и машинного обучения. Стандартный пакет Python с официального сайта включает только базовые библиотеки.

3. Как осуществить проверку работоспособности пакета Anaconda?

Для проверки работоспособности Anaconda можно запустить интерактивную оболочку IPython или Jupyter Notebook. Также можно создать новое окружение и установить несколько библиотек для проверки.

4. Как задать используемый интерпретатор языка Python в IDE PyCharm?

В PyCharm можно задать интерпретатор Python в настройках проекта или в глобальных настройках IDE. Для этого перейдите в "File" -> "Settings" (или "Preferences" на macOS) -> "Project: [имя проекта]" -> "Python Interpreter" и выберите нужный интерпретатор Python.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Для запуска программы в PyCharm можно нажать кнопку "Run" (Запустить) или "Debug" (Отладка) в верхней панели. Также можно использовать комбинации клавиш, например, Shift + F10 для запуска.

6. В чем суть интерактивного и пакетного режимов работы Python?

Интерактивный режим позволяет вводить команды Python построчно и немедленно видеть результат. Пакетный режим используется для выполнения скриптов и программ, которые выполняются целиком.

7. Почему язык программирования Python называется языком динамической типизации?

Язык Python называется динамическим из-за того, что типы данных переменных определяются автоматически во время выполнения программы, а не во время компиляции.

8. Какие существуют основные типы в языке программирования Python?

Основные типы данных в Python включают числа (int, float), строки (str), списки (list), кортежи (tuple), множества (set), словари (dict), булевы значения (bool), и многое другое.

9. Как создаются объекты в памяти? Каково их устройство? В чем заключается процесс объявления новых переменных и работа операции присваивания?

Переменные - это ссылки на объекты. Процесс объявления переменных заключается в присвоении им значений, и Python автоматически выделяет память для хранения объектов.

10. Как получить список ключевых слов в Python?

Получение списка ключевых слов в Python можно сделать с помощью модуля keyword. Используйте import keyword и keyword.kwlist для получения списка ключевых слов.

11. Каково назначение функций id() и type()?

Функция id() возвращает уникальный идентификатор объекта в памяти, а функция type() возвращает тип объекта.

12. Что такое изменяемые и неизменяемые типы в Python.

Изменяемые типы данных могут быть изменены после создания, например, списки. Неизменяемые типы данных, такие как кортежи и строки, не могут быть изменены после создания.

13. Чем отличаются операции деления и целочисленного деления?

Операция деления (/) возвращает результат в виде числа с плавающей точкой, а операция целочисленного деления (//) возвращает результат в виде целого числа.

14. Какие имеются средства в языке Python для работы с комплексными числами?

Для работы с комплексными числами в Python используется встроенный тип `complex`.

15. Каково назначение и основные функции библиотеки (модуля) `math`? По аналогии с модулем `math` изучите самостоятельно назначение и основные функции модуля `cmath`.

Библиотека (модуль) `math` в Python предоставляет функции и константы для выполнения математических операций и вычислений. Основное назначение и функции библиотеки `math` включают в себя:

Вычисления математических функций: `math` предоставляет функции для выполнения различных математических операций, таких как корень, логарифмы, тригонометрические функции и др.

Константы: `math` содержит константы, такие как число π (π) и экспонента (e), которые можно использовать в вычислениях.

Округление и модуль: `math` предоставляет функции для округления чисел, нахождения модуля числа, а также другие функции для работы с числами.

Тригонометрические функции: `math` включает в себя тригонометрические функции, такие как синус, косинус, тангенс и другие, которые позволяют выполнять вычисления связанные с углами.

Экспоненциальные и логарифмические функции: `math` предоставляет функции для работы с экспоненциальными и логарифмическими вычислениями, такие как возведение в степень, натуральный логарифм и другие.

Библиотека `math` является полезным инструментом для выполнения разнообразных математических операций в Python, и она широко используется при решении математических задач, научных вычислений и инженерных задач.

Модуль `cmath` предоставляет аналогичные функции для комплексных чисел.

16. Каково назначение именных параметров `sep` и `end` в функции `print()`?

Параметры `sep` и `end` в функции `print()` используются для настройки разделителей между значениями и окончания вывода.

17. Каково назначение метода `format()`? Какие еще существуют средства для форматирования строк в Python? Примечание: в дополнение к рассмотренным средствам изучите самостоятельно работу с f-строками в Python.

Метод `format()` используется для форматирования строк, позволяя вставлять значения в строку. Также в Python есть f-строки для удобного форматирования строк с использованием выражений.

18. Каким образом осуществить ввод с консоли значения целочисленной и вещественной переменной в языке Python?

Для ввода значений с консоли в Python используются функции `input()` для строк, `int(input())` для целых чисел и `float(input())` для вещественных чисел.

Вывод: В ходе выполнения лабораторной работы был успешно установлен и изучен на базовом уровне Python 3.x. Были приобретены базовые навыки программирования на этом языке, что дало возможность создавать и решать задачи с использованием Python. Кроме того была освоена модель `git-flow`.