

Лабораторная работа №1

Задачи.

Ссылка на Replite: <https://replit.com/@letablohina2017/Practice>

Выполнила Блохина Валерия студентка 2 курса ИВТ.

Задание 1

```
C:\Users\79112>pip

Usage:
  pip <command> [options]

Commands:
  install          Install packages.
  download         Download packages.
  uninstall        Uninstall packages.
  freeze           Output installed packages in requirements format.
  inspect          Inspect the python environment.
  list             List installed packages.
  show            Show information about installed packages.
  check            Verify installed packages have compatible dependencies.
  config           Manage local and global configuration.
  search           Search PyPI for packages.
  cache            Inspect and manage pip's wheel cache.
  index            Inspect information available from package indexes.
  wheel            Build wheels from your requirements.
  hash            Compute hashes of package archives.
  completion       A helper command used for command completion.
  debug            Show information useful for debugging.
  help            Show help for commands.

General Options:
  -h, --help       Show help.
  --debug          Let unhandled exceptions propagate outside the main subroutine, instead of logging them
```

Для одновременного выполнения программ в Replite были сделаны 2 файла и далее они импортированы в главный файл. Поэтому при отсутствии ошибок в работе калькулятора (1 задание), программа выводит сразу требования ввести число для 2 программы (2 задание)

Задание 3.1

```
main.py x task1.py task2.py +
main.py
1  # main.py
2  import task1
3  import task2
4
```

```
main.py task1.py task2.py +
task1.py > ...
1 """Calculates the result of the calculator
2
3 Arguments:
4 a (int, float): The first number.
5 b (int, float): The second number.
6 operation (str): The operation to be performed.
7     - 'add' for addition,
8     - 'sub' for subtraction,
9     - 'div' for division,
10    - 'mult' for multiplication.
11
12 Returns:
13 int, float: The result of the operation.
14
15 Exceptions:
16 ValueError: If the division operation and the second number is zero.
17 """
18 def calculate(a, b, operation):
19     if operation == 'add':
20         return a + b
21     if operation == 'sub':
22         return a - b
23     if operation == 'div':
24         if b == 0:
25             raise ValueError("Division is not possible")
26         return a / b
27     if operation == 'mult':
28         return a * b
29
30 def test_add():
31     assert calculate(1, 2, 'add') == 3
32 def test_sub():
33     assert calculate(8, 6, 'sub') == 2
34 def test_div():
35     assert calculate(64, 8, 'div') == 8
36 def test_mult():
37     assert calculate(6, 9, 'mult') == 54
38
39 test_add()
40 test_sub()
41 test_mult()
42 test_div()
```

Задание 3.2

```
main.py task1.py task2.py +
task2.py > ...
1 """The function takes a number and a range (low and high values) and tries to guess a number using
2 binary search.
3 Returns the guessed number and the number of attempts, which were required for the search. If the
4 number is not found in the range, returns None and the number of attempts.
5 Guessing arguments:
6 number (int): The number that needs to be guessed.
7 low (int): The lower limit of the search range.
8 high (int): The upper limit of the search range.
9 The program displays the desired number and the number of attempts
10 """
11 def guessing(number, low, high):
12     attempts = 0
13     while low <= high:
14         attempts += 1
15         middle = (low + high) // 2
16         if middle < number:
17             low = middle + 1
18         elif middle > number:
19             high = middle
20         else:
21             return middle, attempts
22     return None, attempts
23
24 target_number = int(input("Guess the number from 1 to 100: "))
25
26 while target_number < 1 or target_number > 100:
27     print("Guess the number from 1 to 100.")
28     target_number = int(input("Guess the number from 1 to 100: "))
29
30 guessed_number, num_attempts = guessing(target_number, 1, 100)
31
32 if guessed_number is not None:
33     print(f"I guessed your number: {guessed_number} in {num_attempts} attempts!")
34 else:
35     print("Couldn't guess the number.")
```

Run

Guess the number from 1 to 100: 45
I guessed your number: 45 in 7 attempts!

