

Ссылка на Replit: <https://replit.com/@melnik3570/LR1?v=1>

Комплект 3: Задачи для самостоятельной работы.

В Replit модуль main.py создан для вызова двух программ одновременно из других модулей, т.к. в других случаях модуль task2.py не запускался по неизвестной ошибке.

main.py

```
1  '''Модуль для вызова функций из task1.py и task2.py'''
2
3  print('Задача 3.1')
4  import task1
5  print('Задание 3.2')
6  import task2
7  |
```

```
Задача 3.1
Введите первое число: 5
Введите второе число: 45
Введите оператор: *
225.0
test passed
test passed
test passed
test passed
Задание 3.2
Введите границы диапазона
Введите начало диапазона: 3
Введите конец диапазона: 89
Введите загаданное число в пределах диапазона: 13
Загаданное число: 13, Число сравнений: 3
```

3.1: Создайте простую программу калькулятор, которая позволяет из функции `main()` ввести два числа и тип арифметической операции, а потом вычисляет результат. Свой код опубликуйте на <https://replit.com/~> и предоставьте ссылку в ответах на лабораторную работу в Moodle в документе-отчёте. Реализацию арифметических действий и вычисление результата с его возвратом сделайте в отдельной функции `calculate(...)`. Протестируйте свой калькулятор с помощью вызова нескольких своих простых функций `test_*`() с ключевым словом `assert` внутри. Обязательно напишите хорошую документацию к своему коду.

```
1 def calculate(a: float, b: float, operator: str):
2     ''' Выполняет определенную арифметическую операцию над двумя числами.
3     Если заданный операнд равен предложенным, то выполняется соответствующая операция и возвращается
4     результат вычисления.
5     Аргументы:
6     a - первое число.
7     b - второе число.
8     operator - оператор, который задается и используется для выполнения операции.
9     return - возвращает результат выполнения арифметической операции.
10    '''
11    if operator == '+':
12        return a + b
13    elif operator == '-':
14        return a - b
15    elif operator == '*':
16        return a * b
17    elif operator == '/':
18        if b == 0:
19            return 'На ноль делить нельзя!'
20        return a / b
21    else:
22        return 'Недопустимый оператор'
23
24 def test_add():
25     '''Проверяет функцию calculate() на сложения чисел через тестирование.'''
26     assert calculate(1, 2, '+') == 3
27     return 'test passed'
28
29 def test_sub():
30     '''Проверяет функцию calculate() на вычитание чисел через тестирование.'''
31     assert calculate(4, 65, '-') == -61
32     return 'test passed'
```

```

32
33 def test_mult():
34     '''Проверяет функцию calculate() на умножение чисел через тестирование.'''
35     assert calculate(5, 5, '*') == 25
36     return 'test passed'
37
38 def test_div():
39     '''Проверяет функцию calculate() на деление чисел через тестирование.'''
40     assert calculate(10, 2, '/')
41     return 'test passed'
42
43 def main():
44     '''Основная функция.
45     Аргументы:
46     number1 - первое число.
47     number2 - второе число.
48     operator - оператор, который задается и используется для выполнения операции.
49     '''
50     number1 = float(input('Введите первое число: '))
51     number2 = float(input('Введите второе число: '))
52     operator = str(input('Введите оператор: '))
53
54     print(calculate(number1, number2, operator))
55     print(test_add())
56     print(test_sub())
57     print(test_div())
58     print(test_mult())
59
60 main()

```

Задача 3.1

```

Введите первое число: 5
Введите второе число: 45
Введите оператор: *
225.0
test passed
test passed
test passed
test passed

```

3.2: Реализуйте программно классическую простую игру "угадай число" (guess number) с помощью алгоритма медленного перебора (инкремента) по одному числу, либо с помощью алгоритма бинарного поиска. Алгоритм принимает на вход само число, которое он должен угадать, интервал значений в котором оно загадано и в цикле делает угадывания тем или иным выбранным вами способом. После угадывания из функции алгоритма возвращается угаданное число и число угадываний/сравнений, которые пришлось проделать. Обязательно напишите хорошую документацию к своему коду.

```
1 def binary_search():
2     '''Выполняет бинарный поиск по заданным числам.
3     Если число найдено, то возвращает загаданное число и количество итераций.
4     Если не найдено, то приходит функция сообщает об этом.
5     Аргументы:
6     number1 - первое число, начало диапазона.
7     number2 - второе число, конец диапазона.
8     number3 - третье число, загаданное для поиска.
9     counter - параметр цикла, который считает количество итераций.
10    '''
11    print('Введите границы диапазона')
12    number1 = int(input('Введите начало диапазона: '))
13    number2 = int(input('Введите конец диапазона: '))
14    number3 = int(input('Введите загаданное число в пределах диапазона: '))
15    counter = 0
16
17    while number1 <= number2:
18        center = (number1 + number2) // 2
19        counter += 1
20        if number3 == center:
21            print(f'Загаданное число: {center}, Число сравнений: {counter}')
22            return
23        elif number3 > center:
24            number1 = center + 1
25        else:
26            number2 = center - 1
27    print('Загаданное число не входит в диапазон')
28
29    print(f'Загаданное число: {number3}, Число сравнений: {counter}')
30
31    binary_search()
```

Задание 3.2

Введите границы диапазона

Введите начало диапазона: 3

Введите конец диапазона: 89

Введите загаданное число в пределах диапазона: 13

Загаданное число: 13, Число сравнений: 3