

Лабораторная работа 1

Программа 1 — [калькулятор](#):

```
1  """this program calculates operations based on the user input"""
2
3
4  def calculate(val1, val2, operation):
5      """
6      main calculating function
7      expects first variable, second variable, and operation in order
8      supports only +,-,*,/
9      """
10     # checking the operation type
11     match operation:
12         case '+':
13             return val1 + val2
14         case '-':
15             return val1 - val2
16         case '*':
17             return val1 * val2
18         case '/':
19             return val1 / val2
20         case _:
21             # if user inputs unsuported operation throw an error
22             return 'Invalid Operation'
23
24
25     # tests
26     def test_calculate_summation(val1, val2):
27         assert calculate(val1, val2, '+') == val1 + val2, 'addition'
28
29
30     def test_calculate_subscription(val1, val2):
31         assert calculate(val1, val2, '-') == val1 - val2, 'subtraction'
32
33
34     def test_calculate_multiplication(val1, val2):
35         assert calculate(val1, val2, '*') == val1 * val2, 'multiplication'
36
37
38     def test_calculate_division(val1, val2):
39         assert calculate(val1, val2, '/') == val1 / val2, 'division'
40
41
42     # thirst variable
43     a = input()
44     # second variable
45     b = input()
46
47     # testing
48     test_calculate_summation(a,b)
49     test_calculate_subscription(a,b)
50     test_calculate_multiplication(a,b)
51     test_calculate_division(a,b)
```

Программа 2 — [угадыватель числа из диапазона](#):

```
1  """This program searches for a number in a given diapason, by eliminating half of the variants
2  and then trying all others at random, storing made guesses in an array"""
3  import random
4
5  # array of tried numbers
6  tries = []
7  # number of tries it took to guess the number
8  number_of_tries = 0
9  # the correct guess
10 the_correct_guess = 0
11
12
13 # main function of the program
14 def guess(min, max, number):
15     """
16     Main function expects three arguments:
17     range in two int variables
18     and the number to guess also in int
19     """
20     # importing the tries array, the number of tries and the correct guess
21     global tries
22     global number_of_tries
23     global the_correct_guess
24     # eliminating half of the variants
25     # by checking if the number is
26     # bigger or smaller than the middle of min and max
27     if number < max - ((max - min) / 2):
28         # creating a current guess by getting a random variable in the new range
29         current_guess = random.randint(min, (max - int((max - min) / 2)))
30         # if the guess is correct print the number of tries,
31         # the guessed number and all the tried numbers
32         if current_guess == number:
33             print(
34                 f"The number was {current_guess} and it took {number_of_tries} tries to guess it."
35             )
36             print(f"The tries were {tries}")
37             the_correct_guess = current_guess
38
39     # if not check if the guess was already made and
40     # if not add it to the tries array and check again
41     else:
42         if current_guess in tries:
43             guess(min, max, number)
44         else:
45             tries.append(current_guess)
46             number_of_tries += 1
47             guess(min, max, number)
48     # same as previous steps but if the range is in the latter half
49     else:
50         current_guess = random.randint(max - int((max - min) / 2), max)
51         if current_guess == number:
52             print(
53                 f"The number was {current_guess} and it took {number_of_tries} tries to guess it."
54             )
55             print(f"The tries were {tries}")
56             the_correct_guess = current_guess
57         else:
58             if current_guess in tries:
59                 guess(min, max, number)
60             else:
61                 tries.append(current_guess)
62                 number_of_tries += 1
63                 guess(min, max, number)
64
65 # test
66 def test_guess():
67     global the_correct_guess
68     guess(1, 10, 5)
69     return the_correct_guess == 5
70 print(test_guess())
71
```