ГУАП  
КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент |  |  |  | А.В. Аграновский |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| УСЛОВНЫЕ КОНСТРУКЦИИ |
| по курсу: ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | М023 |  |  |  | Д.А. Трегуб |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

ПОСТАНОВКА ЗАДАЧИ

Задание № 9.10.  
Вычислить значение функции fрез(x) при условии, что функция вычисляется по формуле:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № вар | a1 | f1(x) | a2 | f2(x) | a3 | f3(x) |
| 10 | 7 |  | -7 |  | -21 |  |

Требования к программе:  
Цель программы: вывод значения функции в заданной точке или сообщения об ошибке. На вход подаётся аргумент x, по которому нужно вычислить функцию. Выходными данными является число или сообщения по ошибке.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

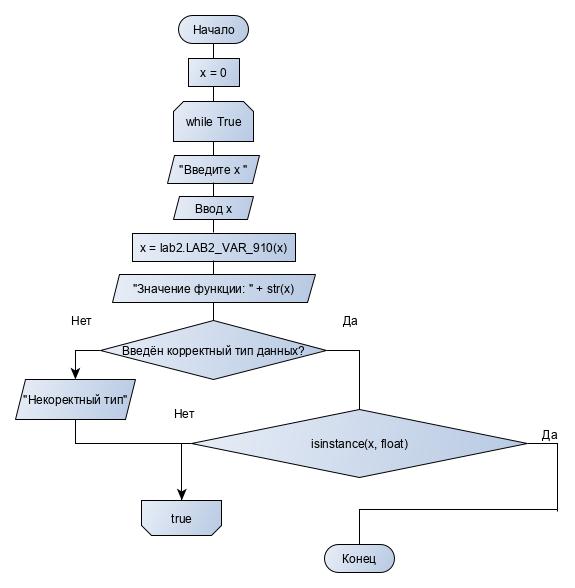
Преобразование формулы:

ОПИСАНИЕ РАЗРАБОТАННОЙ ПРОГРАММЫ

Таблица 1 Описание переменных

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип переменной | Назначение |
| x | Int/Float | Запись начальных данных/Хранение результата вычислений |

На вход подаётся 1 число x.

Рисунок 1 – функция main

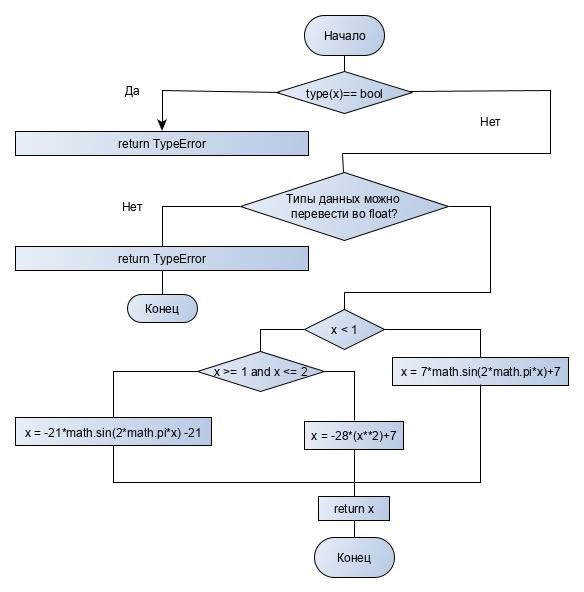


Рисунок 2 - функция LAB2\_VAR\_910

ОПИСАНИЕ ТЕСТОВОГО НАБОРА

Таблица 2 Тестовый набор

|  |  |  |  |
| --- | --- | --- | --- |
| Описание тестового случая | Входные данные | Ожидаемый результат | Результат теста |
| Передано целое число | x = 0 | 7 | OK |
| Передано дробное число | x = 1.2 | -33.32 | OK |
| Введен строковый тип данных | x = “var” | TypeError | OK |
| Введен логический тип данных | True | TypeError | OK |
| Введен массив | [1] | TypeError | OK |

В случае непрохождения теста выводится результат работы программы после сообщения о непрохождении.

ПРИМЕР РАБОТЫ ПРОГРАММЫ

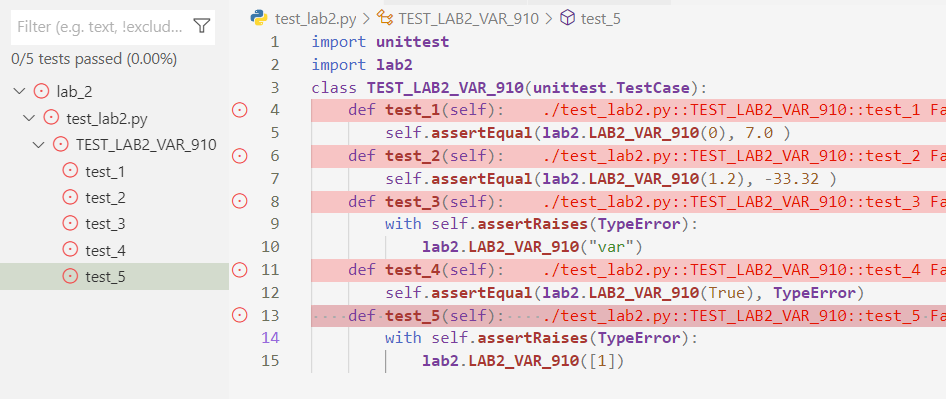


Рисунок 3 - Результат работы тестов до написания программы

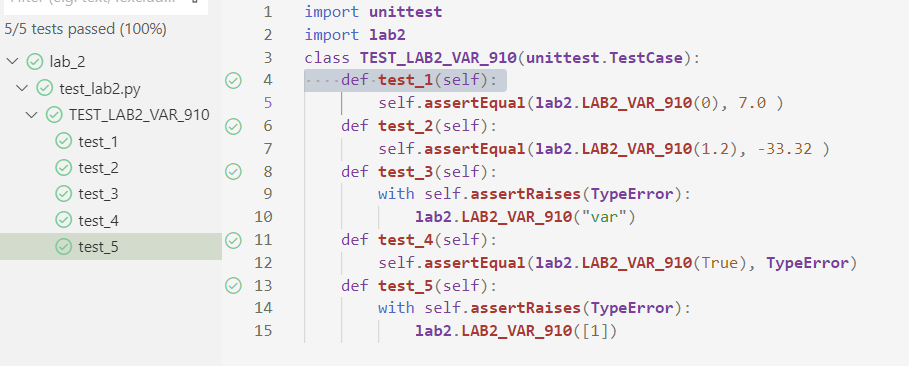


Рисунок 4 - Результат работы тестов после написания программы

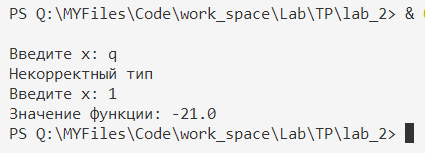


Рисунок 5 - Результат работы программы

ВЫВОД

Были улучшены навыки работы с python, изучены условные конструкции, была выполнена поставленная задача.

ПРИЛОЖЕНИЯ

Приложение 1

Код программы

Код тестов:

import **unittest**

import **lab2**

class **TEST\_LAB2\_VAR\_910**(**unittest**.**TestCase**):

    def **test\_1**(self):

        self.**assertEqual**(**lab2**.**LAB2\_VAR\_910**(0), 7.0 )

    def **test\_2**(self):

        self.**assertEqual**(**lab2**.**LAB2\_VAR\_910**(1.2), -33.32 )

    def **test\_3**(self):

        with self.**assertRaises**(**TypeError**):

**lab2**.**LAB2\_VAR\_910**("var")

    def **test\_4**(self):

        self.**assertEqual**(**lab2**.**LAB2\_VAR\_910**(True), **TypeError**)

    def **test\_5**(self):

        with self.**assertRaises**(**TypeError**):

**lab2**.**LAB2\_VAR\_910**([1])

Код функции LAB2\_VAR\_910:

import **math**

def **LAB2\_VAR\_910**(x):

*#Проверка, что введённые данные не являются логическим значением*

    if **type**(x) == **bool**:

        return **TypeError**

*#Попытка привести введённые данные к типу float*

    try:

        x = **float**(x)

*#Вывод ошибки TypeError в случае неудачи*

    except:

        raise **TypeError**

*#При x меньше 1 вычислить*

    if x < 1:

       x = 7\***math**.**sin**(2\***math**.pi\*x)+7

*#Если x больше или равен 1, или меньше или равен 2, вычислить*

    elif x >= 1 and x <= 2:

        x = -28\*(x\*\*2)+7

*#Иначе вычислить*

    else:

        x = -21\***math**.**sin**(2\***math**.pi\*x) -21

*#Вернуть x*

    return x;

Код функции main:

import lab2

*#Инициализация переменной*

x=0

*#Начало бесконечного цикла*

while True:

*#Начало блока обработки исключений*

    try:

        x = **input**("Введите x: ") *#Ввод х*

        x = lab2.LAB2\_VAR\_910(x)*#Вызов функции и запись результата её работы в переменную*

**print**("Значение функции: " + str(x))*#Вывод в консоль результата*

*#При вызове исключения*

    except TypeError:

**print**("Некорректный тип")*#Вывод в консоль сообщения об ошибке*

*#Выход в случае успешного выполнения*

    if **isinstance**(x, float):

        break