МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ

КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

ордена Трудового Красного Знамени федеральное государственное

бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информатика»

Практические работы

по дисциплине

«Проектный практикум»

|  |  |
| --- | --- |
|  | Выполнил: студент гр. БЭИ2202  Кулешов А. С.  Вариант 14  Проверил: к.п.н., доцент кафедры «Информатика» Гуриков С. Р. |

Москва, 2024 г.

**Практическая работа №1**

**«МЕТОД ПОШАГОВОЙ ДЕТАЛИЗАЦИИ»**

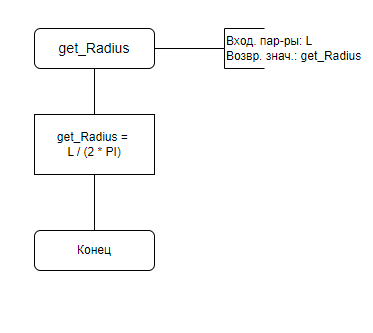
1. Индивидуальное задание на разработку проекта

Вычислить значение площади каждого из трёх кругов, ограниченных тремя окружностями, длины которых L1, L2, L3 известны. Используя формулы для окружностей:

**Назначение функции**: Вычислить площадь круга , предварительно вычислив R по формуле

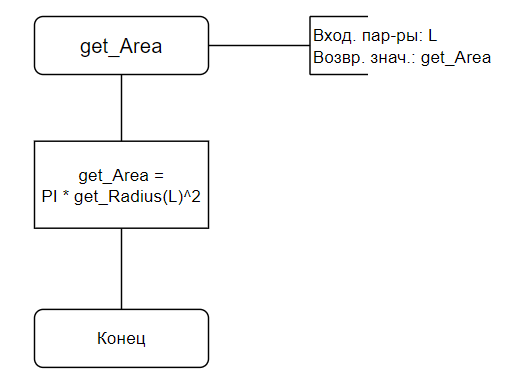
1. Схемы алгоритмов

Для функции, решающей поставленную задачу, была написана вспомогательная функция нахождения радиуса окружности get\_Radius(), алгоритм которой представлен на рисунке 1.



* + 1. Схема Функция вычисления радиуса

Cхема функции, которая решает задачу показана на рисунке 2.



* + 1. Cхема функции, которая решает задачу. Функция вычисления площади круга

Также был реализован алгоритм, который позволяет работать 3 тремя полями одновременно, см. на рисунке 3.

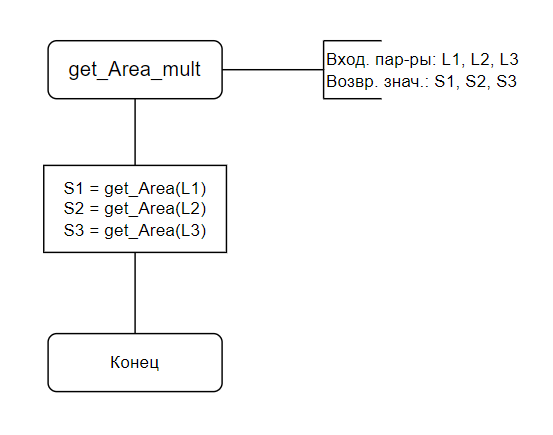


Рисунок 3 – Функция вычисления площади круга для нескольких кругов

Схема функции основной программы показана на рисунке 4.

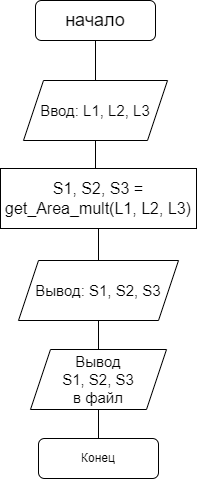


Рисунок 4 – Схема функции основной программы

1. Программный код

import math  
  
def get\_Radius(L):  
 return L / 2 \* math.pi  
  
def get\_Area(L):  
 return math.pi \* get\_Radius(L)\*\*2  
  
def get\_Area\_mult(L1, L2, L3):  
 return get\_Area(L1), get\_Area(L2), get\_Area(L3)  
  
L1, L2, L3 = 0, 0, 0  
  
try:  
 L1 = float(input("Введите длину окружности L1: "))  
 L2 = float(input("Введите длину окружности L2: "))  
 L3 = float(input("Введите длину окружности L3: "))  
 if (L1 < 0 or L2 < 0 or L3 < 0):  
 print("Введённые значения должны быть не меньше нуля")  
 quit()  
except ValueError:  
 print("Введённое значение должно быть числом")  
 quit()  
  
S1, S2, S3 = get\_Area\_mult(L1, L2, L3)  
  
print(f"Выходные данные:\nS1: {S1}, S2: {S2}, S3:{S3}")  
file = open("result.txt", "w", encoding="utf-8")  
file.write(f'Исходные данные:\n L1: {L1}, L2: {L2}, L3:{L3} \n')  
file.write(f'Выходные данные:\n S1: {S1}, S2: {S2}, S3: {S3}')  
file.close()

1. Результаты работы программы

На рисунке 5 представлен результат работы программы при выдуманных данных для проверки значения, представленного программой. Как видно, программа работает корректно.

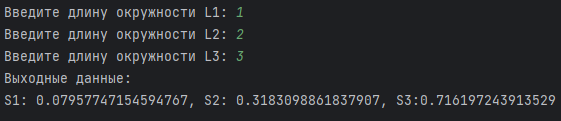


Рисунок 5 – Результаты работы программы

Результат записи в файл показан на рисунке 6.

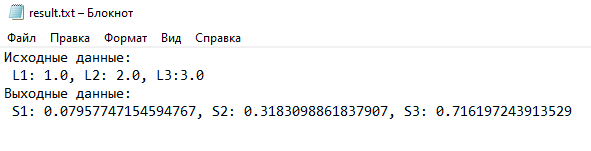


Рисунок 6 – Результаты записи в файл

Обработка исключений показана на рисунках 7-8.

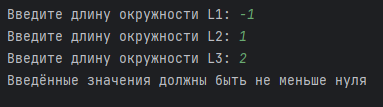


Рисунок 7 – Обработка исключений. Отрицательные числа

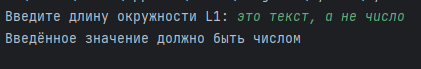


Рисунок 8 – Обработка исключений. Текст

1. Сравнительный анализ работы двух программ

Обе программы были созданы с одной целью - получение значения функции и обеспечение взаимодействия с пользователем.

Программа, написанная на языке C++, использует графический интерфейс пользователя, который основан на Windows Forms Application. В программе на CLI/C++ пользователь вводил данные в текстовые поля и получал ответ в другие текстовые поля, рисунок 9.

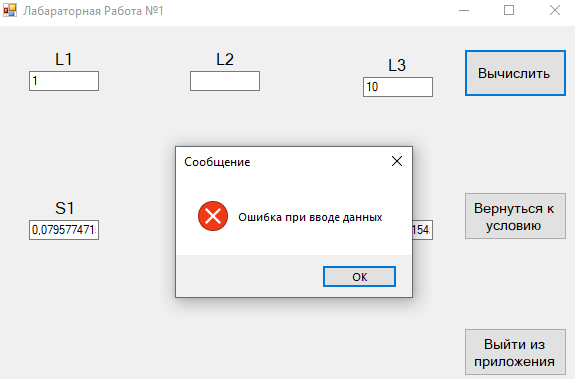


Рисунок 8 – Ввод и вывод данных в C++. Сообщение об ошибке

Программа на языке Python предлагает интерфейс в виде командной строки. В ней пользователь вводит значение переменной в консоль, и результат выводится также в консоль. Рисунок 9.

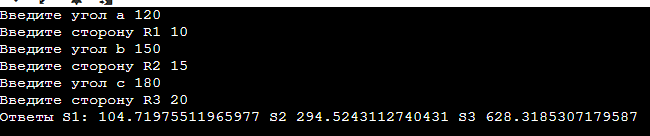


Рисунок 9 – Ввод и вывод данных в Python

В обеих программах предусмотрена обработка возможных ошибок при вводе данных. В программе на C++, если пользователь вводит некорректное значение, ему выводится сообщение об ошибке в диалоговом окне, рисунок 9. В программе на Python при вводе некорректного значения также выводится сообщение об ошибке, но в консоль.

Суммируя все выше перечисленное, разница обеих программ в интерфейсе, среде разработки, способах ввода и вывода данных и в языке программирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Гуриков С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. — Москва : ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст : электронный. - URL: https://znanium.com/catalog/product/1356003 (дата обращения: 26.02.2024). – Режим доступа: по подписке.

2 ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) — М., Стандартинформ, 2017.

**Практическая работа №2**

**«программирование алгоритмов разветвляющихся структур»**

1. Индивидуальное задание на разработку проекта

Запрограммировать систему кусочно-ломанной функции согласно индивидуальному варианту №14:

1. Программный код

Программный код основного файла main.py продемонстрирован ниже

from module import \*  
import logging  
logging.basicConfig(filename='log.txt', level=logging.INFO, encoding='utf-8')  
  
  
  
  
x, a, z, b, c = 0, 0, 0, 0, 0  
  
try:  
 x = float(input("Введите x: "))  
 a = float(input("Введите a: "))  
 z = float(input("Введите z: "))  
 b = float(input("Введите b: "))  
 c = float(input("Введите c: "))  
except ValueError as e:  
 print("Введённые значения должны быть числового типа")  
 logging.error(str(e))  
 quit()  
  
result = 0  
  
try:  
 result = func(x,a,z,b,c)  
except Exception as e:  
 print("Произошла ошибка при вычислениях")  
 logging.error(str(e))  
 quit()  
print()  
print(f"Выходные данные:")  
print(f"func(три знака после запятой): {'%.3f' % result}")  
print(f"func(ceil): {math.ceil(result)}")  
print(f"func(floor): {math.floor(result)}")  
print(f"func(round): {round(result)}")  
print(f"func(trunc): {math.trunc(result)}")  
  
file = open("result.txt", "w", encoding="utf-8")  
file.write(f'Исходные данные:\n x: {x}, a: {a}, z:{z}, b:{b}, c:{c} \n')  
  
file.write(f"Выходные данные:\n")  
file.write(f"func(три знака после запятой): {'%.3f' % result}\n")  
file.write(f"func(ceil): {math.ceil(result)}\n")  
file.write(f"func(floor): {math.floor(result)}\n")  
file.write(f"func(round): {round(result)}\n")  
file.write(f"func(trunc): {math.trunc(result)}\n")  
  
logging.info(f"Значение результирующей функции: {'%.3f' % result}")  
  
  
file.close()

Программный код файла модуля module.py продемонстрирован ниже

import math  
  
def func(x, a, z, b, c):  
  
 if x < 0 and z < 0:  
 m = x  
 if a > m:  
 m = a  
 m2 = z  
 if x \* a \*\* m < m2:  
 m2 = x \* a \*\* m  
 return math.log(x \* a + m2)  
 else:  
 if 0 <= x and x < 4 and 0 <= z and z < 3:  
 m = z  
 if (math.sqrt(x) > m):  
 m = math.sqrt(x)  
 if (abs(math.sqrt(x)) > m):  
 m = abs(math.sqrt(x))  
 return math.log(b \*\* 2) + m  
 else:  
 return b + c \* x

1. Результаты работы программы

На рисунках 1-3 представлены результаты работы программы при выдуманных данных для проверки значений, представленных программой. Каждый тест был подготовлен на соответствующую ветку решения. Как видно, программа работает корректно.

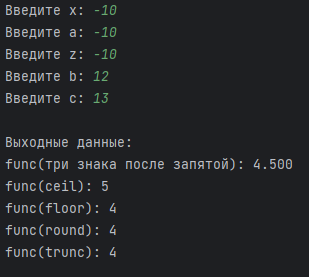


Рисунок 1 – Результаты работы программы для первого ветвления

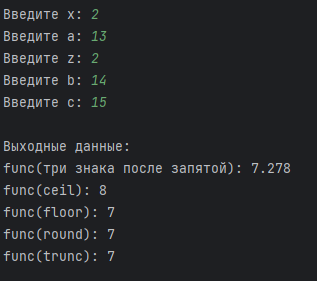


Рисунок 2 – Результаты работы программы для второго ветвления

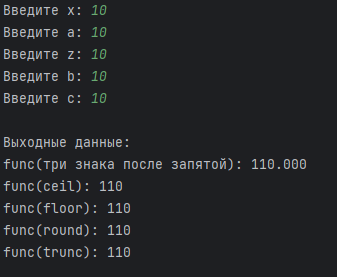


Рисунок 3 – Результаты работы программы для третьего ветвления

Результат записи в файл показан на рисунке 4.

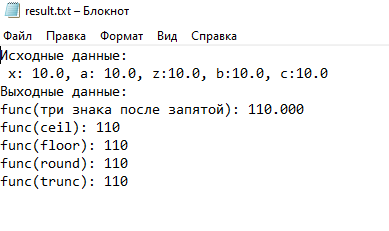


Рисунок 4 – Результаты записи в файл

Обработка исключений показана на рисунке 5.

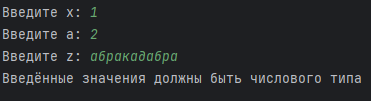


Рисунок 5 – Обработка исключений. Текст

Запись логов показана на рисунке 6.

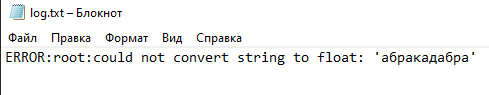


Рисунок 6 – Запись логов

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Гуриков С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. — Москва : ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст : электронный. - URL: https://znanium.com/catalog/product/1356003 (дата обращения: 26.02.2024). – Режим доступа: по подписке.

2 ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) — М., Стандартинформ, 2017.