**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | 09.02.07 Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по лабораторной работе № 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Анализ и обеспечение обработки исключительных ситуаций. | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Тестирование информационных систем |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Назаров Никита Максимович |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-311 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев Иван Валерьевич |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024 г**

**Лабораторная работа №4.** **«Анализ и обеспечение обработки исключительных ситуаций.»**

**Задания:**

1. **Задание №1**:
   * Составить программу, которая вычисляет возраст по вводимому с клавиатуры году рождения;
   * За исключения принять значения года рождения меньше нуля и год рождения, больше текущего;
   * При помощи try – throw – catch (или аналогичных операторов обработки исключений на других языках) составить программу так, чтобы при исключении каждого вида выводилось соответствующее сообщение;

**Листинг составленной программы на языке питон:**

import datetime

def err(y):

    try:

        int(y)

        return False

    except ValueError:

        return True

def calculate\_age(birth\_year):

    current\_year = datetime.datetime.now().year

    if birth\_year < 0:

        raise ValueError("Год рождения не может быть меньше нуля.")

    elif birth\_year > current\_year:

        raise ValueError("Год рождения не может быть больше текущего года.")

    return current\_year - birth\_year

def main():

    try:

        year\_of\_birth = input("Введите год рождения: ")

        if err(year\_of\_birth):

            print("Ошибка ввода")

        else:

            age = calculate\_age(int(year\_of\_birth))

            print(f"Ваш возраст: {age} лет.")

    except ValueError as e:

        print(f"Ошибка: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

* + Составить отчёт о работе программы, прикрепив скриншоты её работы при правильном результате и при каждом из видов исключений.

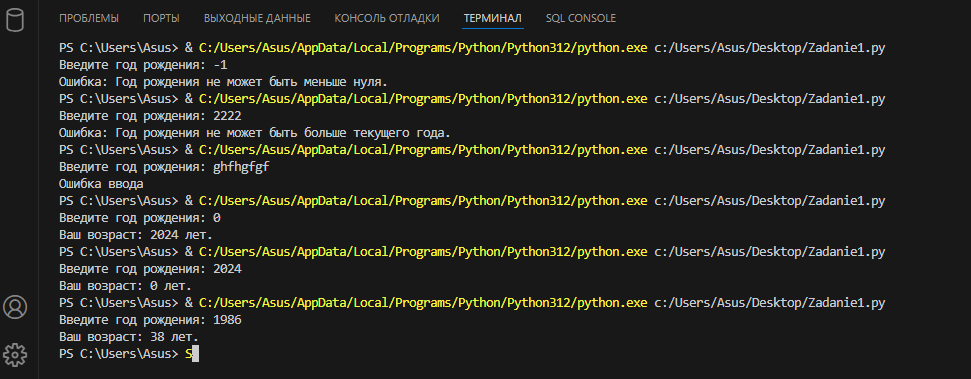
Скриншот терминала с примерами работы программы

Рисунок 1 Скриншот терминала с примерами работы программы

|  |  |  |
| --- | --- | --- |
| **Значение** | **Ожидаемый результат** | **Фактический результат** |
| -1 | Год рождения не может быть меньше нуля. | Год рождения не может быть меньше нуля. |
| 2025 | Год рождения не может быть больше текущего года. | Год рождения не может быть больше текущего года. |
| ghfhgfgf | Ошибка ввода | Ошибка ввода |
| 0 | 2024 | 2024 |
| 2024 | 0 | 0 |
| 1986 | 38 | 38 |

1. **Задание №2**:
   * Составить программу которая по вводимым с клавиатуры А, В и С вычисляла бы acos(A)+sqrt(sin(PI\*B))/(tan(PI\*C)+1); Примерная структура программы дана на следующем слайде, можно также использовать её костяк для первой программы.
   * Понять, какие значения А, В и С дают какие исключения;

**A может принимать значения: -1 <= А <= 1 (арккосинус)**

**B должно быть таким чтобы под корнем было не отрицательное значение**

**C должно быть таким, чтобы не было деления на 0, а так же не должно быть кратным 1/2, условие определенности тангенса.**

* + Аналогично **Заданию №1:** при помощи try – throw – catch (или аналогичных операторов обработки исключений на других языках) составить программу так, чтобы при исключении каждого вида выводилось соответствующее сообщение;

**Листинг составленной программы на языке питон:**

import math

PI = math.pi

def calculate(A, B, C):

    if A < -1.0 or A > 1.0:

        raise ValueError("Ошибка: A должно быть в диапазоне [-1, 1].")

    if math.sin(PI \* B) < 0:

        raise ValueError("Ошибка: Выражение под корнем принимает отрицательные значения.")

if C % 2 == 0.75:

        raise ValueError("Ошибка: tan(PI \* C) + 1 не должно быть равно нулю.")

    if (C % 0.5 == 0.0) and (C != 0):

        raise ValueError("Ошибка: Тангенс не определен для Пи/2 + Пи\*n (n=0,1,2...).")

    return math.acos(A) + math.sqrt(math.sin(PI \* B)) / (math.tan(PI \* C) + 1)

def main():

    try:

        A = float(input("Введите значение A (-1 <= A <= 1): "))

        B = float(input("Введите значение B (sin(PI \* B) не должно принимать отрицательные значения) : "))

        C = float(input("Введите значение C (не кратно 1/2 а так же не должно приводить к tan(PI \* C) + 1 = 0: "))

        result = calculate(A, B, C)

        print(f"Результат: {result}")

    except ValueError as e:

        print(e)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

* + Аналогично **Заданию №1:** cоставить отчёт о работе программы, прикрепив скриншоты её работы при правильном результате и при каждом из видов исключений.

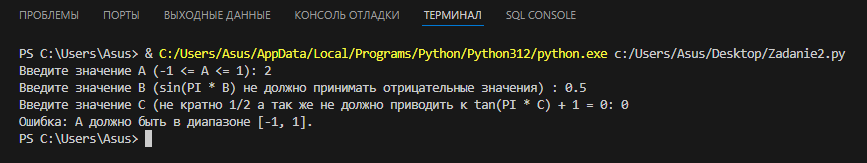


Рисунок 2 Исключение, когда А в не допустимом диапазоне

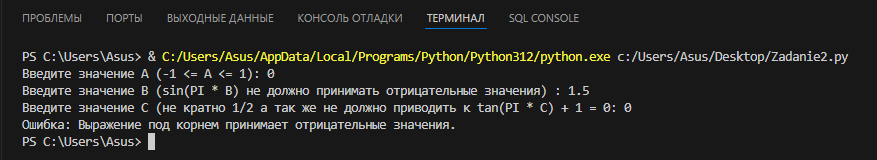


Рисунок 3 Исключение, когда B приводит к отрицательному значению под корнем

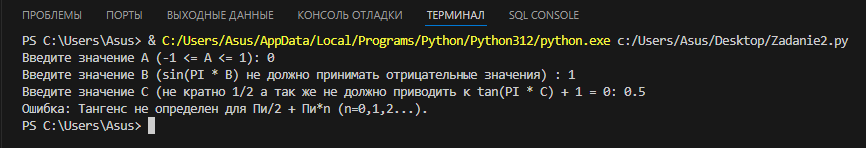


Рисунок 4 Исключение, когда значение С таково, что тангенс не определен

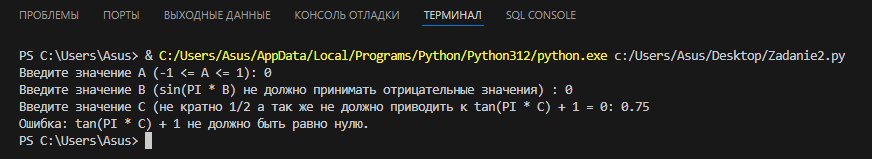


Рисунок 5 Исключение, когда С таково, что тангенс равен -1, и возникает деление на 0.

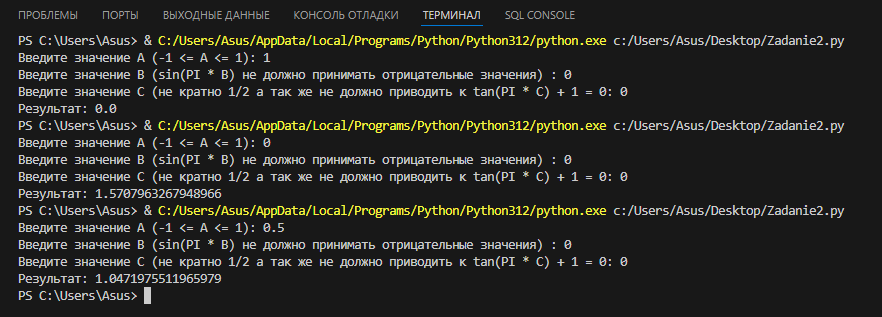


Рисунок 6 Работа кода для корректных значений А,В,С