Отчёт  
по практической работе №7  
по дисциплине «Инструментальные средства разработки программного обеспечения»  
Тема: «Организация обработки исключений»

|  |  |  |
| --- | --- | --- |
| Студент группы 22 ИТ |  | Липинский К.С. |
| Преподаватель |  | Быковский З.С. |

## Цель работы

Изучить операторы, используемые при обработке исключительных ситуаций, возникающих во время выполнения вычислительных процессов, получить практические навыки в составлении программ.

## Выполнение работы

Код проекта (рис. 1)



Рисунок 1

Тест проекта (рис. 2, рис. 3, рис. 4)

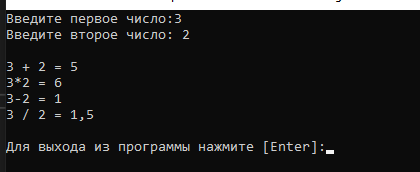


Рисунок 2

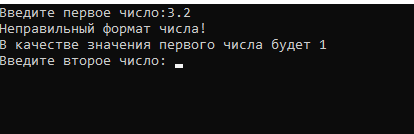


Рисунок 3

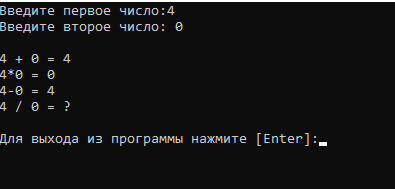


Рисунок 4

Код второго задания (рис. 5)

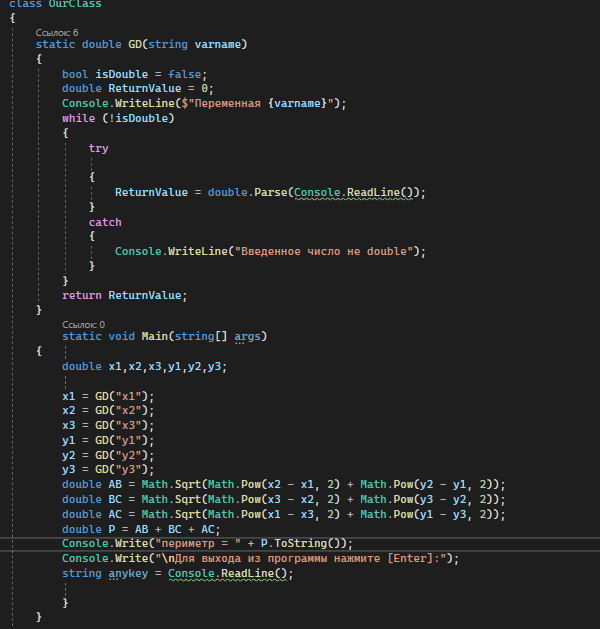


Рисунок 5

Тест (рис. 6 и рис. 7)

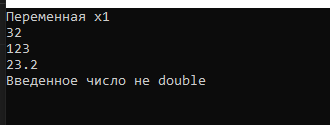


Рисунок 6

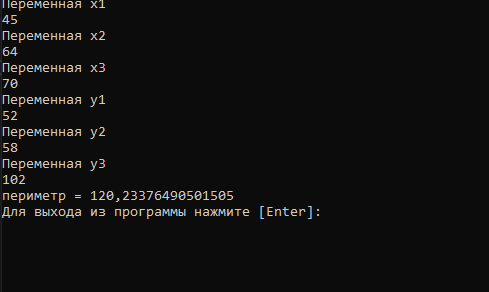


Рисунок 7

## Контрольные вопросы

**Создание защищенного блока кода**: В языках программирования, таких как Python, Java или C#, защищенный блок кода создается с использованием конструкции **try**. Код, который может вызвать исключение, помещается внутрь блока **try**, после него следуют блоки **catch** (в Java и C#) или **except** (в Python) для обработки исключений.

**Процедура обработки исключения**: Чтобы обработать конкретное исключение, в блоке **except** (в Python) или **catch** (в Java/C#) указывается тип исключения.

**Генерация исключения**: Исключение можно сгенерировать с помощью ключевого слова **raise** (в Python) или **throw** (в Java/C#).

**Ограничение списка исключений, которые могут генерироваться в функции**: Чтобы ограничить исключения, генерируемые в функции, можно использовать конструкцию **throws** в Java, указывая типы исключений, которые может "бросать" метод. В Python такого явного ограничения нет, но можно документировать возможные исключения в комментариях или использовать аннотации.