Міністерство освіти і науки України
Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації і управління

Звіт

з лабораторної роботи №6 з дисципліни
“Програмування 2. Структури даних та алгоритми”

“ДОСЛІДЖЕННЯ МЕХАНІЗМУ ОБРОБКИ ВИНЯТКОВИХ СИТУАЦІЙ”

Варіант №13

Виконав студент Флорчук Назарій Петрович

Перевірив викладач Проскура Світлана Леонідівна

Мета роботи:

1. Дослідити механізм обробки ситуацій та причини його застосування мовами програмування C++ та C#.

Завдання:

1. Вивчити причини застосування обробки виняткових ситуацій.
2. Дослідити механізм обробки виняткових ситуацій мовами програмування C++ та C#.
3. Написати програми мовами C++ та C#, у яких створити клас що представляє собою вираз. Членами даними класу будуть операнди виразу, а операціями – методи встановлення значень виразу та його обчислення, згідно варіанту. При обчисленні арифметичного виразу та введенні інформації передбачити обробку виняткових ситуацій.
4. Доробити вихідний код лабораторної роботи з перевантаження операцій та операторів, включивши обробку виняткових ситуацій там, де це потрібно.

Арифметичний вираз	Опис класу - виразу	Завдання
(8 * lg(b - 1) - c) / (a * 2 + b / c)	Змінні - операнди виразу; конструктори; методи встановлення значень об’єкта, обчислення виразу, отримання значення об’єкта.	Створити масив об’єктів; обчислити вираз для кожного об’єкта, вивести результат.

Код програми (C++):

```
#include <iostream>
#include <time.h>
#include <math.h>

/**
 * Division by zero custom exception
 */
class DivideByZeroException : public std::exception {
   private:
      std::string message;

   public:
      DivideByZeroException(std::string message) : message(message)
      {
         //
      }

      std::string what()
      {
         return message;
      }
};

class Expression
{
   private:
      int _m_a;
      int _m_b;
      int _m_c;

   public:
      Expression(int a = 0, int b = 0, int c = 0)
      {
         this->_m_a = a;
         this->_m_b = b;
         this->_m_c = c;
      }

      /**
       * Get expression result
       */
      float Get()
      {
         if (this->_m_c == 0)
         {
            throw DivideByZeroException(
               "(8 * log(b - 1) - c) / (a * 2 + b / c) expression. Division by zero (c = 0). (a = "
```

```
                + std::to_string(this->_m_a) + "; b = " + std::to_string(this->_m_b) + "; c = "
                + std::to_string(this->_m_c) + ")."
            );
        }

        float divider = (this->_m_a * 2 + this->_m_b / this->_m_c);

        if (divider == 0)
        {
            throw DivideByZeroException(
                "(8 * log(b - 1) - c) / (a * 2 + b / c) expression. Division by zero. ((a * 2 + b / c) = 0). (a = "
                + std::to_string(this->_m_a) + "; b = " + std::to_string(this->_m_b) + "; c = " +
std::to_string(this->_m_c) + ")."
            );
        }

        return (float) (8 * log(this->_m_b - 1) - this->_m_c) / divider;
    }

    /**
     * Get object a property value
     */
    int GetA()
    {
        return this->_m_a;
    }

    /**
     * Get object b property value
     */
    int GetB()
    {
        return this->_m_b;
    }

    /**
     * Get object c property value
     */
    int GetC()
    {
        return this->_m_c;
    }

    /**
     * Set value for object a property
     */
    void SetA(int a)
    {
        this->_m_a = a;
```

```
    }

    /**
     * Set value for object b property
     */
    void SetB(int b)
    {
        this->_m_b = b;
    }

    /**
     * Set value for object c property
     */
    void SetC(int c)
    {
        this->_m_c = c;
    }
};

int main()
{
    const unsigned int LENGTH = 15;

    std::srand((unsigned) time(NULL));

    Expression expressions[LENGTH];

    for (int i = 0; i < LENGTH; i++)
    {
        int a = (int) rand();
        int b = (int) rand();
        int c = (int) rand();

        if (i == 0) {
            // Check first exception
            c = 0;
        } else if (i == 1) {
            // Check second exception
            a = 0;
            b = 0;
        }

        expressions[i] = Expression(a, b, c);

        try {
            float result = expressions[i].Get();

            std::cout << "(8 * log(b - 1) - c) / (a * 2 + b / c) = " << result << ". (a = " << a << "; b = " << b
                    << "; c = " << c << ")." << std::endl;
```

```
        } catch (DivideByZeroException e) {
            std::cout << e.what() << std::endl;
        } catch (...) {
            std::cout << "Something goes wrong." << std::endl;
        }
    }

    exit(0);
}
```

```
[redacted]@debian:~/Documents/kpi/basics_of_programming_2/lab_6$ g++ lab_6.cpp -o lab_6
[redacted]@debian:~/Documents/kpi/basics_of_programming_2/lab_6$ ./lab_6
(8 * log(b - 1) - c) / (a * 2 + b / c) expression. Division by zero (c = 0). (a = 729640712; b = 519882401; c = 0).
(8 * log(b - 1) - c) / (a * 2 + b / c) expression. Division by zero. ((a * 2 + b / c) = 0). (a = 0; b = 0; c = 173002199).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -1.65291. (a = 416264853; b = 2015226782; c = 1376100255).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 0.355303. (a = 1560009180; b = 1066786855; c = 417463151).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 0.387143. (a = 1210479295; b = 2121055174; c = 725510469).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -10.3437. (a = 54299629; b = 1229258141; c = 1123323352).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -4.47853. (a = 81957389; b = 710544391; c = 734097611).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -0.342722. (a = 960835475; b = 557760972; c = 658598284).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -0.761092. (a = 641584496; b = 1863116597; c = 976610369).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -1.68791. (a = 446213955; b = 1650880401; c = 1506334509).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 1.30104. (a = 1368793592; b = 233037466; c = 2026216911).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 0.692062. (a = 1522210276; b = 827907315; c = 865455803).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 0.810574. (a = 1695212475; b = 1244172168; c = 733198937).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -0.974198. (a = 923829082; b = 656697700; c = 1799985792).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 1.09996. (a = 1341292233; b = 1867176995; c = 1773557318).
```

Код програми (C#):

```
class Expression
{
    private int _m_a;
    private int _m_b;
    private int _m_c;

    public Expression(int a = 0, int b = 0, int c = 0)
        {
            this._m_a = a;
            this._m_b = b;
            this._m_c = c;
        }

        /**
         * Get expression result
         */
        public float Get()
        {
            if (this._m_c == 0)
            {
                throw new System.DivideByZeroException(
                    "(8 * log(b - 1) - c) / (a * 2 + b / c) expression. Division by zero (c = 0). (a = "
                    + this._m_a + "; b = " + this._m_b + "; c = " + this._m_c + ")."
                );
            }

            float divider = (this._m_a * 2 + this._m_b / this._m_c);

            if (divider == 0)
            {
                throw new System.DivideByZeroException(
                    "(8 * log(b - 1) - c) / (a * 2 + b / c) expression. Division by zero ((a * 2 + b / c) = 0). (a = "
                    + this._m_a + "; b = " + this._m_b + "; c = " + this._m_c + ")."
                );
            }

            return (float) (8 * System.Math.Log(this._m_b - 1) - this._m_c) / divider;
        }

        /**
         * Get object a property value
         */
        public int GetA()
        {
            return this._m_a;
        }
```

```
        /**
         * Get object b property value
         */
        public int GetB()
        {
            return this._m_b;
        }

        /**
         * Get object c property value
         */
        public int GetC()
        {
            return this._m_c;
        }

        /**
         * Set value for object a property
         */
        public void SetA(int a)
        {
            this._m_a = a;
        }

        /**
         * Set value for object b property
         */
        public void SetB(int b)
        {
            this._m_b = b;
        }

        /**
         * Set value for object c property
         */
        public void SetC(int c)
        {
            this._m_c = c;
        }
};

class Application
{
    static void Main(string[] args)
    {
        const int LENGTH = 15;

        System.Random rand = new System.Random();
```

```
        Expression[] expressions = new Expression[LENGTH];

        for (int i = 0; i < LENGTH; i++)
        {
            int a = rand.Next();
            int b = rand.Next();
            int c = rand.Next();

            if (i == 0) {
                // Check first exception
                c = 0;
            } else if (i == 1) {
                // Check second exception
                a = 0;
                b = 0;
            }

            expressions[i] = new Expression(a, b, c);

            try {
                float result = expressions[i].Get();

                System.Console.WriteLine(
                    "(8 * log(b - 1) - c) / (a * 2 + b / c) = " + result
                    + ". (a = " + a + "; b = " + b + "; c = " + c + ")."
                );
            } catch (System.DivideByZeroException e) {
                System.Console.WriteLine(e.Message);
            } catch {
                System.Console.WriteLine("Something goes wrong.");
            }
        }
    }
}
```

```
[redacted]@debian:~/Documents/kpi/basics_of_programming_2/lab_6$ mcs lab_6.cs
[redacted]@debian:~/Documents/kpi/basics_of_programming_2/lab_6$ mono lab_6.exe
(8 * log(b - 1) - c) / (a * 2 + b / c) expression. Division by zero (c = 0). (a = 1365701956; b = 255616635; c = 0).
(8 * log(b - 1) - c) / (a * 2 + b / c) expression. Division by zero ((a * 2 + b / c) = 0). (a = 0; b = 0; c = 1491059816).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -0.5605144. (a = 1010270026; b = 2100624153; c = 1132541944).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 1.776852. (a = 1939876885; b = 453751307; c = 737773128).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 0.2400233. (a = 1222844393; b = 194418817; c = 443870155).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 0.5883434. (a = 1710506527; b = 1437297046; c = 514185413).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 1.170945. (a = 1319308599; b = 240119234; c = 1939494821).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 0.009787622. (a = 1276447151; b = 2122521558; c = 17050922).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -0.781441. (a = 1031701678; b = 993379257; c = 1612428048).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -1.158733. (a = 826880534; b = 1504516594; c = 1916267527).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -0.1613353. (a = 400653817; b = 304051623; c = 129279388).
(8 * log(b - 1) - c) / (a * 2 + b / c) = -1.007517. (a = 288630357; b = 1399688868; c = 581600276).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 0.936874. (a = 1189868584; b = 566417457; c = 1794329464).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 0.5951869. (a = 1380619978; b = 172231438; c = 912854647).
(8 * log(b - 1) - c) / (a * 2 + b / c) = 16.88436. (a = 2120008419; b = 1668196169; c = 927803323).
```

Висновки / Відповіді на контрольні запитання:

1. Дайте визначення виняткової ситуації.
   Винятковою ситуацією є подія, яка перериває поточне нормальне виконання програми та потребує обробки.
2. У яких випадках необхідно передбачати обробку виняткових ситуацій?
   Коли необхідно продовжити виконання програми, навіть після виникнення виняткової ситуації.
3. У чому полягає механізм обробки виняткової ситуації?
   Механізм обробки виняткової ситуації полягає у використанні логічних блоків *try* та *catch*.
4. Поясніть, як користувач може згенерувати виняткову ситуацію.
   Для генерування виняткової ситуації використовується ключове слово *throw*.
5. Поясніть призначення та устрій блоку *try*.
   В середині логічного блоку *try*, описується код програми, в якому потенційно можливе виникнення виняткової ситуації.
6. Поясніть призначення та устрій обробників виняткових ситуацій.
   - *catch* — логічний блок, у якому виконується обробка виняткової ситуації. Може використовуватись як для перехоплення заданого типу виключення, так і всіх потенційних виключень;
   - *finally* (лише у C#) — логічний блок, який містить інструкції, які виконуються в не залежності від того, виникла виняткова ситуація, чи ні.
7. Поясніть призначення та устрій блоку *filally*.
   Логічний блок *filally*, виконується завжди після логічних блоків *try* та *catch*.
   Використовується, якщо потрібно виконати операцію, після тих же логічних блоків *try* та *catch*.
8. Поясніть сутність використання конструкції using при обробці виключень.
   —
9. Поясніть сутність використання вкладених блоків *try*.
   Логічні блоки *try*, можуть бути вкладеними, тобто використовуватись в середині один одного.
   - C++:
     ```
     try {
        try {
           //
        } catch(...) {
           //
        }
     } catch(...) {
        //
     }
     ```
   - C#:
     ```
     try {
        try {
           //
        } catch {
           //
        } finally {
           //
     ```

```
    }
} catch {
   //
} finally {
      //
}
```

10. Поясніть, у якій послідовності розташовуються блоки *catch*.
    Логічні блок *catch*, розміщуються одразу після логічного блоку *try*.
11. Поясніть призначення та наведіть приклад опису та використання об’єкта виключення у обробнику виключення.
    Приклад наведено у коді програми.
12. Поясніть, як обирається тип для опису виключення в C++ та C#.
    Обирається той тип виключення, який найкраще підходить у заданій ситуації.
13. Поясніть принцип створення користувацького типу виключення в C#.
    - для C++, створюється похідний клас, користувацького типу виключення, від стандартного базового класу std::exception;
    - для C#, створюється похідний клас, користувацького типу виключення, від стандартного базового класу Exception.
14. Наведіть приклади загально відомих типів виключень.
    - C++:
      - std::exception — базовий клас усіх стандартних виключень у C++;
      - std::range_error — викликається, при спробі зберегти значення, яке виходить за межі діапазону;
      - std::invalid_argument — виключення викликається під передачі недійсного аргументу функції, методу etc.;
      - ...
    - C#:
      - System.Exception — базовий клас усіх стандартних виключень у C#;
      - System.IndexOutOfRangeException — викликається, коли є місце посиланню на індекс масиву поза діапазоном;
      - System.DivideByZeroException — викликається, у результаті ділення на нуль;
      - System.IO.FileNotFoundException — викликається, при спробі отримати доступ до не існуючого файлу;
      - System.ArgumentException — виключення викликається під передачі недійсного аргументу функції, методу etc.;
      - ...
15. Наведіть основні члени типу Exception в C#.
    —
16. Поясніть принцип дії та наведіть форми операторів checked та unchecked в C#.
    —