Практическое занятие № 17 «Обработка ошибок, исключительные ситуации»

Учебные цели:

- получение умений и навыков обработки ошибок, и использования исключительных ситуаций в C++.

Воспитательные цели:

- формировать диалектико-материалистическое мировоззрение;
- формировать навыки самостоятельности и дисциплинированности;
- стимулировать активную познавательную деятельность обучаемых, способствовать формированию у них творческого мышления.

Категория слушателей: 2,3 курс.
Время: 90 мин.
Место проведения: компьютерный класс.
Материально-техническое обеспечение:
1) персональный компьютер *IBM PC* с операционной системой Windows XP; 2) среда разработки приложений *Visual C++.NET.*

ПЛАН ПРАКТИЧЕСКОГО ЗАНЯТИЯ

Учебные вопросы	Время, мин
Вступительная часть	5
1. Механизм исключительных ситуаций	15
2. Выполнение индивидуального задания	65
Заключительная часть	5

Рассмотрим примеры использования исключительных ситуаций.

Пример 1. Написать программу, которая находит результат деления одного целого числа на другое. Если втрое число равно нулю, вызвать соответствующее исключение.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
class DivideByZeroException { };
int main()
{
      int m,n;
      double res;
      try {
cout<<"m=";
      cin>>m;
      cout<<"n=";
      cin>>n;
      if (n==0)
```

```
throw (DivideByZeroException());
      res=m/double(n);
      cout<<"res="<<res<<endl;
      }
      catch (DivideByZeroException)
      {
            cout<<"Error n=0!"<<endl;
      }
      return 0;
}
```

Пример 2. Следующая программа MELTDOWN.CPP иллюстрирует использование функции nuke_meltdown. Эта программа использует оператор try для разрешения обнаружения исключительной ситуации. Далее программа вызывает функцию add_u232 с параметром amount. Если значение этого параметра меньше 255, функция выполняется успешно. Если же значение параметра превышает 255, функция генерирует исключительную ситуацию nuke_meltdown:

```
 #include <iostream.h>
 class nuke_meltdown
 {
 public:
    nuke_meltdown(void)  { cout << "\a\a\aРаботаю! Работаю!
Работаю!"  << endl; }
 };
 void add_u232(int amount)
 {
    if (amount < 255) cout << "Параметр add_u232 в порядке"  <<
endl;
    else throw nuke_meltdown();
 }
 void main(void)
 {
    try
    {
       add_u232(255);
    }
    catch (nuke_meltdown)
    {
       cout << "Программа устойчива"  << endl;
    }
 }
```

Контрольные вопросы

1. Механизм исключительных ситуаций.
2. Возбуждение и обработка ситуаций.
3. Свертка стека, исполнение конструкторов и деструкторов. Поддержка иерархии классов.
4. Стандартные классы исключительных ситуаций.
5. Примеры простых программ с использованием исключительных ситуаций.

Порядок выполнения лабораторной работы

1. Напишите программу согласно Вашему варианту задания.
2. Показать результат работы программы преподавателю.
3. Защитить лабораторную работу.

Требования к отчету

Отчет должен содержать:

1. конспект теоретической части;
2. лабораторное задание;
3. схему алгоритма;
4. порядок выполнения лабораторной работы;
5. результаты выполнения программ.

Практическое задание

Задание 1. Напишите и отладьте программу для вычисления значения функции f(x), с использованием механизма исключительных ситуаций, предусматривающий вывод на экран различных сообщений о некорректном вводе значения переменной x. Например, «При введенном значении x не существует значения квадратного корня» или «При введенном значении x не существует значения логарифма», или «При введенном значении x происходит деление на ноль», а также сообщения о том, что введённое значение *x* не принадлежит области допустимых значений функции.

Варианты задания 1.

1. $f(x) = arctg(\frac{(\sin(x)+\cos(x))}{\sqrt{2x}})$
2. $f(x) = e^{\frac{\sqrt{2}}{\sin(x)}} + 3x^2 - 1$
3. $f(x) = \ln(-\sqrt{3}\cos(3x)) + x^3 + 2x$
4. $f(x) = \sqrt[3]{(x+1)(x^2 + 2\sqrt{x} - 2)}$
5. $f(x) = \sqrt[3]{(\frac{1}{x} - 1)^2} - \sqrt[3]{(x-2)^2}$
6. $f(x) = -\sqrt[3]{(\sqrt{x+1} + 3)(x^3 + 6x + 6)}$
7. $f(x) = \ln(\frac{x^2 - 5x + 4}{x}) + 2x$
8. $f(x) = \frac{e^{2(x+2)}}{-(2x+3)} - 3x^4 + x$
9. $f(x) = \frac{-x^2 - 4x + 13}{4x + 3} - \ln(x+1)$
10. $f(x) = \frac{9 - 10x^2}{\sqrt{4x^2 - 1}} + \cos(\frac{3x}{x^2 - 1})$

11. $f(x)=x^2-2\sqrt{x}+\frac{16}{x-1}-13$
12. $f(x)=\frac{4}{x^2}-8x-15\sin(3\sqrt{x})$
13. $f(x)=\ln(\frac{1+2\sqrt{-x-x^2}}{2x+1})+3x$
14. $f(x)=\frac{17-x^2}{4x-5}+\sin(x+2)$
15. $f(x)=\ln(\frac{x}{x+5})-x^3+4x+13\sqrt{x-1}$

Пример выполнения задания 1 для функции f(x)=$1+\frac{2\sqrt{x}+3}{x-1}$

```
#include "stdafx.h"
#include <iostream>
#include "math.h"
class DivideByZeroException { };
class Koren { };
int main()
{
     using namespace std;
 float y,x;
  try {
       cout << "Vvedite x=";
 cin >> x;
  if (x<0)
       throw (Koren());
       if (x-1==0) throw (DivideByZeroException());
      y=1+2*sqrt(x)/(x-1);
            cout <<"Pri x="<< x <<" y="<< y <<endl;
           }
  catch (DivideByZeroException)
     {
          cout<<"Error=delenie na 0!"<<endl;
     }
  catch (Koren)
     {
          cout<<"Error=kvadratny koren!"<<endl;
     }

return 0;
}
```

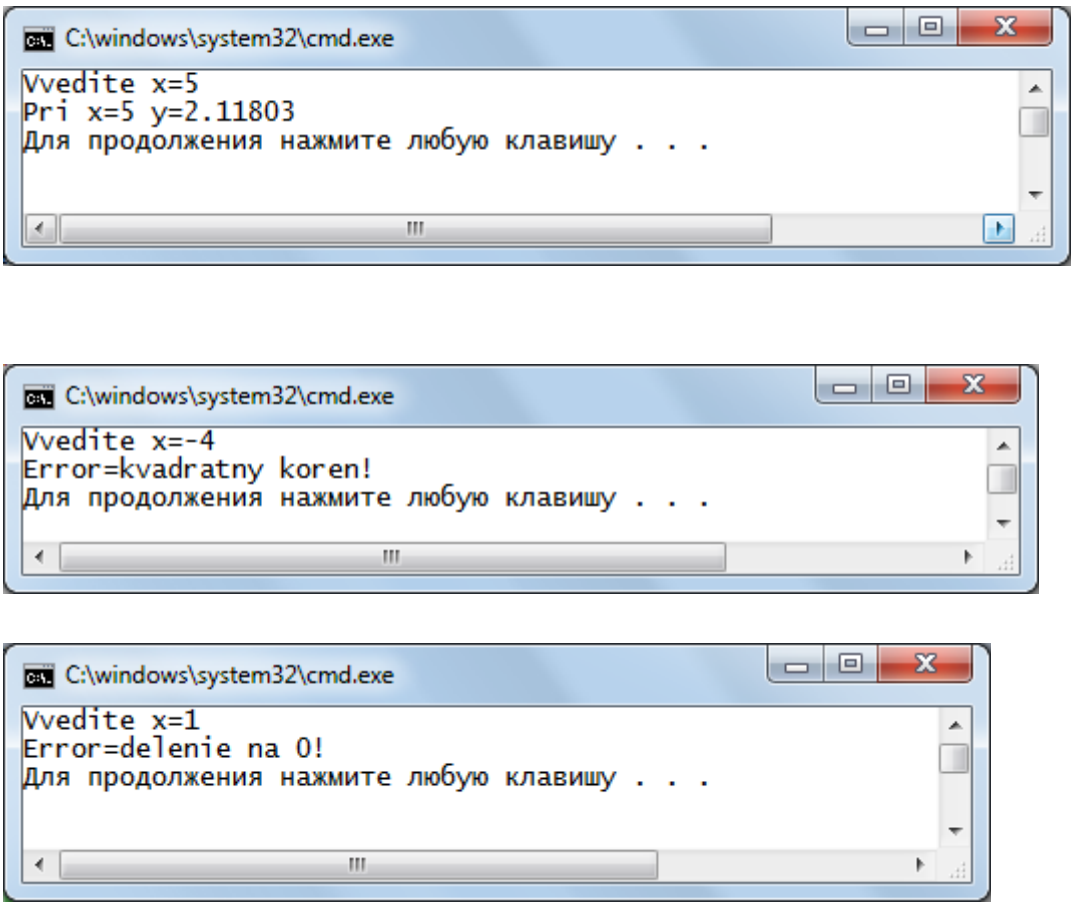

Рис. 1. Результаты работы программы примера 1