Министерство образование Республики Беларусь

Учреждение образования «Полоцкий государственный университет»

Информационно-технологический факультет

Кафедра технологий программирования

Лабораторная работа №7 по теме «ОБРАБОТКА ИСКЛЮЧЕНИЙ»

По дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Выполнил: Ролевич М.А.

Группа 19-ИТ-1.

Проверил: Васильева Д. М.

Новополоцк

2021 г.

**Цель работы: о**своить приемы обработки исключений (аварийных ситуаций) в программах на языке С++.

**Порядок выполнения работы**

1. Написать программу, в которой перехватываются исключения типа int,

char \* и Exception. Сгенерировать исключительную ситуацию.

2. Добавить к программе перехват любой исключительной ситуации.

3. Создайте тип ошибка - error. Добавить к программе перехват

исключительной ситуации типа, созданного Вами.

4. Создайте типы ошибок памяти и с файлами, наследуемые от error

**Задание 1.** Перехват исключений типа int, char, exception:

1. class Exeption
2. {
3. public:
4. string m\_text;
5. Exeption(string str)
6. {
7. m\_text = str;
8. }
9. void show()
10. {
11. cout << m\_text << endl;
12. }
13. };
14. double moreThan(double a)
15. {
16. if (a > 10) {
17. throw Exeption("Number couldn't be greaten that 10");
18. }
19. try
20. {
21. throw "Value a < 10";
22. }
23. catch (const char\* e)
24. {
25. cerr << "Catch in function: " << e << endl;
26. }
27. return a;
28. }
29. double mySqrt(double a)
30. {
31. if (a < 0.0) {
32. throw "Can not take sqrt of negative number";
33. }
34. if (a == false) {
35. throw "Value is not number";
36. }
37. try
38. {
39. throw "It's OK";
40. }
41. catch (const char\* e)
42. {
43. cerr << "Catch in function: " << e << endl;
44. }
45. return sqrt(a);
46. }
47. int main()
48. {
49. cout << "Enter a number: ";
50. double a;
51. double b = 5;
52. cin >> a;
53. ifstream file;
54. file.open("emp.txt", ios::in);
55. try
56. {
57. if (!file) {
58. throw "File not found";
59. }
60. double d = mySqrt(a);
61. cout << "The sqrt of " << a << " is " << d << '\n';
62. moreThan(a);
63. throw error(a);
64. double\* a = new double[1000];
65. }
66. catch (Error & e)
67. {
68. e.isOpen();
69. }
70. catch (const char\* e)
71. {
72. cerr << "Catch in main. Error: " << e << endl;
73. }
74. catch (Exeption & e)
75. {
76. e.show();
77. }
78. catch (const error & e)
79. {
80. cerr << "ERROR!!! " << e.m\_code << endl;
81. }
82. catch (bad\_alloc & e)
83. {
84. cerr << e.what() << endl;
85. }
86. }

**Задание 2.** Тип ошибка - error.:

1. class error
2. {
3. public:
4. error(int code)
5. : m\_code(code)
6. {}
7. int m\_code;
8. };

**Задание 3.** Создайте типы ошибок с файлами, наследуемые от error:

1. class Error : public error
2. {
3. private:
4. ifstream file;
5. public:
6. void isOpen() {
7. file.open("Temp.txt", ios::in);
8. if (!file) {
9. throw "File not found";
10. }
11. }
12. };