ФАКУЛЬТЕТ «Информатика и системы управления»  
КАФЕДРА «Информационная безопасность»

ОТЧЕТ  
по лабораторной работе № 6  
по учебной дисциплине «Алгоритмические языки»  
на тему: «Исключения»

Вариант 5 (23)

Выполнил:   
Студент 1 курса, гр. ИУ8-24  
Спиридонов Олег

**Цель работы:**

Изучить исключения, производные классы от исключений и работу с ними.

**Условие задачи:**

Вариант № 5

Реализовать класс «Массив – последовательность Фибоначчи». Поля: указатель на массив и его размерность. Конструкторы: конструктор для инициализации полей (через параметры передается обычный массив). При необходимости добавить свои функции. При недопустимых параметрах объекта создается исключение (массив не является последовательностью Фибоначчи). Для исключения создать класс, производный от invalid\_argument. Дополнительно созданный класс для исключения кроме наследуемых полей (необходимо их также использовать) содержит поля класса «Массив – последовательность Фибоначчи» для недопустимого объекта и метод для печати всех данных. Продемонстрировать работу с объектами в случае возникновения исключения и его обработки и без возникновения исключения.

**Текст программы:**

#include <iostream>

using namespace std;

class Fibonacci\_sequence\_exception : public invalid\_argument {

int\* sequence;

int length;

public:

Fibonacci\_sequence\_exception(int array[]) : invalid\_argument("The array is not a Fibonacci sequence!") {

length = sizeof(array) - 1;

sequence = array;

};

void print() {

cout << "Length = " << length << endl << "Sequence: ";

for (int i = 0; i < length; i++) {

cout << sequence[i] << " ";

}

cout << endl;

}

};

class Fibonacci\_sequence {

int \*sequence;

int length;

public:

Fibonacci\_sequence(int array[]) {

length = sizeof(array) - 1;

sequence = array;

if (!test()) {

throw Fibonacci\_sequence\_exception(array);

}

}

bool test() {

if (sequence[0] == 0 and sequence[1] == 1) {

for (int i = 2; i < length; i++) {

if (not(sequence[i] == sequence[i - 1] + sequence[i - 2])) {

return false;

}

}

return true;

}

return false;

}

void print() {

cout << "Length = " << length << endl << "Sequence: ";

for (int i = 0; i < length; i++) {

cout << sequence[i] << " ";

}

cout << endl;

}

};

int main() {

int good\_array[] = {0, 1, 1, 2, 3, 5, 8};

try {

Fibonacci\_sequence good\_sequence(good\_array);

}

catch (Fibonacci\_sequence\_exception exception) {

cout << "Exception: " << exception.what() << endl;

exception.print();

}

int bad\_array[] = {0, 1, 1, 3, 3, 5, 8};

try {

Fibonacci\_sequence bad\_sequence(bad\_array);

}

catch (Fibonacci\_sequence\_exception exception) {

cout << "Exception: " << exception.what() << endl;

exception.print();

}

}

**Результат работы программы:**

Exception: The array is not a Fibonacci sequence!

Length = 7

Sequence: 0 1 1 3 3 5 8

**Вывод:**

В ходе выполнения лабораторной работы было успешно реализовано использование исключений в собственных классах, создание производного класса от исключения, использование блока try/catch. Это позволило познакомиться с применением исключений в практических задачах программирования.