Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

ОТЧЕТ

по лабораторной работе №6

ИСПОЛЬЗОВАНИЕ ПОТОКОВ ВВОДА – ВЫВОДА.

ПОТОКИ (IOSTREAM). ФАЙЛЫ – ПОТОКИ (FSTREAM).

ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ.

Выполнил: ст. гр. ИТб-21

Куркчи А.Э.

Проверила:

Тимофеев И.С.

Севастополь

2015

# 1. Цель

Изучить механизм потокового ввода/вывода, обеспечивающий гибкий и эффективный с гарантией типа метод обработки символьного ввода целых чисел, чисел с плавающей точкой и символьных строк, а также простую модель его расширения для обработки типов, определяемых пользователем. Приобрести практические навыки при написании объектно-ориентированных программ с использованием обработки исключительных ситуаций.

2. Постановка задач

Вариант №1

1. Написать программу – возведение числа m в степень числа n. Входные данные поступают с клавиатуры. Результат выводится в файл в 15 позициях, точность составляет 3 знака, пробелы необходимо заменить символом «@».
2. Создать класс массив букв. Описать перехват ошибок, связанных с недопустимым индексом массива.

3. Текст программы

#include <iostream>

#include <fstream>

#include <iomanip>

#include <stdexcept>

#include <math.h>

**using** **namespace** std**;**

class marray **{**

private**:**

int length**;**

char**\*** array**;**

public**:**

marray**(**const int len**)** **{**

length **=** len**;**

array **=** **new** char**[**len**];**

**}**

void put**(**int i**,**char c**)** **{**

**if(**i **<** 0 **||** i **>=** length**)** **{**

**throw** out\_of\_range**(**"MArray put index"**);**

**}**

array**[**i**]** **=** c**;**

**}**

char get**(**int i**)** **{**

**if(**i **<** 0 **||** i **>=** length**)** **{**

**throw** out\_of\_range**(**"MArray get index"**);**

**}**

**return** array**[**i**];**

**}**

**};**

int main**(**int argc**,**char**\*\*** argv**)** **{**

cout **<<** "SevGU semestr.03 OOP lab.05" **<<** endl**;**

double m**,**n**;**

int len**;**

cout **<<** "Input m and n:" **<<** endl**;**

cin **>>** m **>>** n**;**

cout **<<** "Input char array length:" **<<** endl**;**

cin **>>** len**;**

marray ma**(**len**);**

cout **<<** "Input " **<<** len **<<** " char array elements:" **<<** endl**;**

char c**;**

**for(**int i**=**0**;**i**<**len**;**i**++)** **{**

cin **>>** c**;**

ma**.**put**(**i**,**c**);**

**}**

cout **<<** "Demonstration start" **<<** endl**;**

// cout << "Creating fout object" << endl;

cout **<<** "> ofstream fout(\"result.txt\",ios::append);" **<<** endl**;**

ofstream fout**(**"result.txt"**,**ios**::**app**);**

// cout << "Printing the m^n in fout" << endl;

cout **<<** "> fout << setfill('@') << setw(15) << setprecision(3) << pow(m,n) << endl;" **<<** endl**;**

fout **<<** setfill**(**'@'**)** **<<** setw**(**15**)** **<<** setprecision**(**3**)** **<<** pow**(**m**,**n**)** **<<** endl**;**

// cout << "Closing fout object" << endl;

cout **<<** "> fout.close();" **<<** endl**;**

fout**.**close**();**

// cout << "Printing last char array element" << endl;

cout **<<** "> cout << ma.get(len-1) << endl;" **<<** endl**;**

cout **<<** ma**.**get**(**len**-**1**)** **<<** endl**;**

// cout << "Printing non-existing last+1 char array element" << endl;

cout **<<** "> cout << ma.get(len) << endl;" **<<** endl**;**

**try** **{**

cout **<<** ma**.**get**(**len**)** **<<** endl**;**

**}** **catch(**out\_of\_range**&** e**)** **{**

cout **<<** "Out of range error: " **<<** e**.**what**()** **<<** endl**;**

**}**

// cout << "Putting char to non-existing last+1 char array element" << endl;

cout **<<** "> ma.put(len,'x');" **<<** endl**;**

**try** **{**

ma**.**put**(**len**,**'x'**);**

**}** **catch(**out\_of\_range**&** e**)** **{**

cout **<<** "Out of range error: " **<<** e**.**what**()** **<<** endl**;**

**}**

cout **<<** "Demonstration end" **<<** endl**;**

**return** 0**;**

**}**

4. Тестовые примеры

На рисунке 1 представлен вывод программы

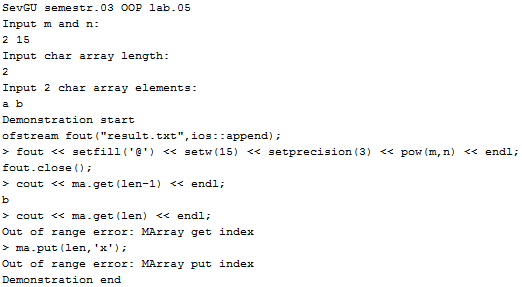


Рисунок 1 – Вывод программы

После ввода m, n, размера массива и соответствующего массива char начинается демонстрация корректности программы. В строках, начинающихся с «>», печатается выполняемая в программном коде строка, в остальных строках – вывод программы. Функция «get» объекта ma возвращает в результате i-й элемент массива, «put» соответственно помещает в i-ю позицию элемент x.

Вывод

В ходе лабораторной работы был изучен механизм потокового ввода/вывода, обеспечивающий гибкий и эффективный с гарантией типа метод обработки символьного ввода целых чисел, чисел с плавающей точкой и символьных строк, а также простую модель его расширения для обработки типов, определяемых пользователем. Приобретены практические навыки при написании объектно-ориентированных программ с использованием обработки исключительных ситуаций.