Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

ОТЧЕТ

по лабораторной работе №5

ШАБЛОНЫ.

Выполнил: ст. гр. ИТб-21

Куркчи А.Э.

Проверила:

Тимофеев И.С.

Севастополь

2015

# 1. Цель

Приобретение практических навыков при написании объектно-ориентированных программ с использованием шаблонов функций и классов. Освоение особенностей отладки объектно-ориентированных программ.

2. Постановка задач

Написать функцию-шаблон, заданную по варианту. Проиллюстрировать её корректную работу на различных по типу наборах данных.

Вариант №1

Написать функцию-шаблон, вычисляющую транспонирование матрицы.

3. Текст программы

#include <iostream>

#include <math.h>

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

template **<**class T**>**

T**\*\*** transpose**(**T**\*\*** m**,**const int h**,**const int w**)** **{**

T**\*\*** r **=** **new** T**\*[**w**];**

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

r**[**i**]** **=** **new** T**[**h**];**

**}**

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

**for(**int j**=**0**;**j**<**w**;**j**++)** **{**

r**[**j**][**i**]** **=** m**[**i**][**j**];**

**}**

**}**

**return** r**;**

**}**

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

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

int n\_int**,**n\_char**,**n\_double**;**

int m\_int**,**m\_char**,**m\_double**;**

cout **<<** "Input int matrix size NxM:" **<<** endl**;**

cin **>>** n\_int **>>** m\_int**;**

cout **<<** "Input int matrix " **<<** n\_int **<<** "x" **<<** m\_int **<<** endl**;**

int**\*\*** ma\_int **=** **new** int**\*[**n\_int**];**

**for(**int i**=**0**;**i**<**n\_int**;**i**++)** **{**

ma\_int**[**i**]** **=** **new** int**[**m\_int**];**

**for(**int j**=**0**;**j**<**m\_int**;**j**++)** **{**

cin **>>** ma\_int**[**i**][**j**];**

**}**

**}**

cout **<<** "Input char matrix size NxM:" **<<** endl**;**

cin **>>** n\_char **>>** m\_char**;**

cout **<<** "Input char matrix " **<<** n\_char **<<** "x" **<<** m\_char **<<** endl**;**

char**\*\*** ma\_char **=** **new** char**\*[**n\_char**];**

**for(**int i**=**0**;**i**<**n\_char**;**i**++)** **{**

ma\_char**[**i**]** **=** **new** char**[**m\_int**];**

**for(**int j**=**0**;**j**<**m\_char**;**j**++)** **{**

cin **>>** ma\_char**[**i**][**j**];**

**}**

**}**

cout **<<** "Input double matrix size NxM:" **<<** endl**;**

cin **>>** n\_double **>>** m\_double**;**

cout **<<** "Input double matrix " **<<** n\_double **<<** "x" **<<** m\_double **<<** endl**;**

double**\*\*** ma\_double **=** **new** double**\*[**n\_double**];**

**for(**int i**=**0**;**i**<**n\_double**;**i**++)** **{**

ma\_double**[**i**]** **=** **new** double**[**m\_double**];**

**for(**int j**=**0**;**j**<**m\_double**;**j**++)** **{**

cin **>>** ma\_double**[**i**][**j**];**

**}**

**}**

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

// cout << "Transposing int matrix ma\_int" << endl;

cout **<<** "> a\_int = transpose(ma\_int,n\_int,m\_int);" **<<** endl**;**

ma\_int **=** transpose**(**ma\_int**,**n\_int**,**m\_int**);**

**for(**int i**=**0**;**i**<**m\_int**;**i**++)** **{**

**for(**int j**=**0**;**j**<**n\_int**;**j**++)** **{**

cout **<<** ma\_int**[**i**][**j**]** **<<** " "**;**

**}**

cout **<<** endl**;**

**}**

// cout << "Transposing int matrix ma\_int" << endl;

cout **<<** "> ma\_char = transpose(ma\_char,n\_char,m\_char);" **<<** endl**;**

ma\_char **=** transpose**(**ma\_char**,**n\_char**,**m\_char**);**

**for(**int i**=**0**;**i**<**m\_char**;**i**++)** **{**

**for(**int j**=**0**;**j**<**n\_char**;**j**++)** **{**

cout **<<** ma\_char**[**i**][**j**]** **<<** " "**;**

**}**

cout **<<** endl**;**

**}**

// cout << "Transposing int matrix ma\_int" << endl;

cout **<<** "> ma\_double = transpose(ma\_double,n\_double,m\_double);" **<<** endl**;**

ma\_double **=** transpose**(**ma\_double**,**n\_double**,**m\_double**);**

**for(**int i**=**0**;**i**<**m\_double**;**i**++)** **{**

**for(**int j**=**0**;**j**<**n\_double**;**j**++)** **{**

cout **<<** ma\_double**[**i**][**j**]** **<<** " "**;**

**}**

cout **<<** endl**;**

**}**

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

**return** 0**;**

**}**

4. Тестовые примеры

На рисунке 1 представлен вывод программы

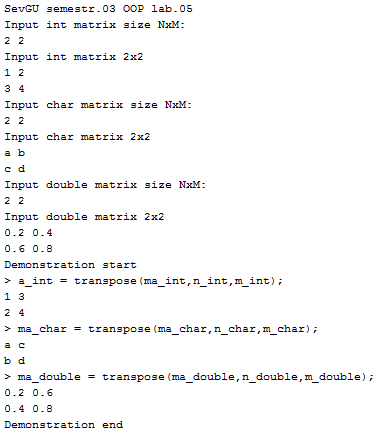


Рисунок 1 – Вывод программы

После ввода 3х размеров и соответствующих матриц для типов int, chat и double начинается демонстрация корректности программы. В строках, начинающихся с «>», печатается выполняемая в программном коде строка, в остальных строках – вывод программы. Функция-шаблон transpose возвращает в результате транспонированную матрицу.

Вывод

В ходе лабораторной работы были приобретены практические навыки написания объектно-ориентированных программ с использованием шаблонов функций и классов. Освоены особенности отладки объектно-ориентированных программ.