Условие:

Задание №1

Создайте динамический массив (vector) размерностью 17 элементов и заполните его случайным образом числами от 1 до 99 в десятичном формате. Создайте новый массив vector2, разделив элементы массива на {1,0,5} соответственно, новый массив не должен содержать целых чисел. Выведите исходный и полученный массив на экран и в файл в несколько форматированных столбцов. Обработайте исключания при делении на 0 и |если файл не открывается.

Задание №2

Найти среднее квадратическое полученного вектора с использованием указателя (нули программа не суммирует). Данные вводить из файла. Осуществить контроль входных данных, используя обработку исключительных ситуаций.

Исходный код программы::  
  
  
#include **<iostream>**#include **<vector>**#include **<string>**#include **<fstream>**#include **<cmath>  
using namespace** std;  
  
**const int** n = 17;  
**class** VectorClass {  
  
**protected**:  
 vector<**int**> vector1;  
 vector<**double**> vector2;  
 FILE\* outfile;  
  
**public**:  
 **void** FileEmpty();  
 **void** initvector1();  
 **void** outvector1();  
 **void** middlevector();  
 **void** initvector2(**double** number);  
 **void** outvector2(**double** number);  
 **void** vector2checker(**double** number);  
 **void** FileOutVectors(**int** way\_id, **double** number);  
};  
  
**void** VectorClass::FileOutVectors(**int** way\_id, **double** number)  
{  
 **try** {  
 outfile = fopen(**"out.txt"**, **"a"**);  
  
 **if** (way\_id == 1) {  
 fprintf(outfile, **"Исходный вектор:\n["**);  
 **for** (**int** i = 0; i < vector1.size(); i++)  
 fprintf(outfile, **"%d "**, vector1[i]);  
  
 fprintf(outfile, **"]\n"**);  
 }  
  
 **if** (way\_id == 2) {  
 fprintf(outfile, **"\nПреобразованный вектор после разделения элементов на %f:\n["**, number);  
 **for** (**int** i = 0; i < vector2.size(); i++)  
 fprintf(outfile, **"%f "**, vector2[i]);  
  
 fprintf(outfile, **"]\n"**);  
 }  
  
 **if** (way\_id == 3) {  
 fprintf(outfile, **"Преобразованный вектор после удаления целых чисел: %f:\n["**, number);  
 **for** (**int** i = 0; i < vector2.size(); i++)  
 fprintf(outfile, **"%f "**, vector2[i]);  
  
 fprintf(outfile, **"]\n"**);  
 }  
  
 fclose(outfile);  
 }  
 **catch** (...) {  
 cout << **"Возникли проблемы с открытием файла"**;  
 }  
}  
  
**void** VectorClass::middlevector()  
{  
 FILE\* bufoutfile;  
 bufoutfile = fopen(**"buf.txt"**, **"w+"**);  
 vector<**double**> vector\_buf;  
  
 *//Выводим вектор в файл* **try** {  
 **for** (**int** i = 0; i < vector2.size(); i++)  
 fprintf(bufoutfile, **"%f "**, vector2[i]);  
 fclose(outfile);  
 }  
 **catch** (...) {  
 cout << **"Возникли проблемы с выводом вектора в файл"**;  
 }  
  
 *//Читаем вектор из файла* **try** {  
 **double** buf;  
 fstream fi;  
 fi.open(**"buf.txt"**);  
 **for** (**int** i = 0; i < vector2.size(); i++) {  
 fi >> buf;  
 vector\_buf.push\_back(buf);  
 }  
 fi.close();  
 }  
 **catch** (...) {  
 cout << **"Возникли проблемы с чтением вектора из файла"**;  
 }  
  
 **double** sum;  
 **for** (vector<**double**>::iterator it = vector\_buf.begin(); it != vector\_buf.end(); it++)  
 sum += pow(\*it, 2);  
 cout << **"Cреднее квадратическое вектора: "** << sqrt(sum / vector\_buf.size());  
}  
  
**void** VectorClass::outvector1()  
{  
  
 cout << **"\nИсходный вектор:\n["**;  
 **for** (**int** i = 0; i < vector1.size(); i++) {  
 cout << vector1[i] << **" "**;  
 }  
 cout << **"]\n"**;  
 FileOutVectors(1, 1);  
}  
  
**void** VectorClass::initvector1()  
{  
 **for** (**int** i = 0; i < n; i++)  
 vector1.push\_back(rand() % 99);  
}  
  
**void** VectorClass::outvector2(**double** number)  
{  
 cout << **"\nПреобразованный вектор после разделения элементов на "** << number << **"\n["**;  
 **for** (**int** i = 0; i < vector2.size(); i++) {  
 cout << vector2[i] << **" "**;  
 }  
 cout << **"]\n"**;  
 FileOutVectors(2, number);  
}  
  
**void** VectorClass::vector2checker(**double** number)  
{  
 **for** (**int** i = 0; i < vector2.size(); i++) {  
 **if** ((vector2[i] - **int**(vector2[i])) == 0)  
  
 vector2.erase(vector2.begin() + i);  
 }  
 cout << **"Преобразованный вектор после удаления целых чисел: \n["**;  
 **for** (**int** i = 0; i < vector2.size(); i++) {  
 cout << vector2[i] << **" "**;  
 }  
 cout << **"]\n"**;  
 middlevector();  
 FileOutVectors(3, number);  
}  
  
**void** VectorClass::initvector2(**double** number)  
{  
 **double** buf\_double;  
 vector2.clear();  
 **for** (**int** i = 0; i < n; i++)  
 vector2.push\_back(vector1[i]);  
  
 **for** (**int** i = 0; i < n; i++) {  
 **try** {  
 **if** (number == 0)  
 **throw** 123;  
 vector2[i] = vector1[i] / number;  
 }  
 **catch** (**int** e) {  
 vector2[i] = 0;  
 cout << **"\nДЕЛЕНИЕ НА 0"**;  
 }  
 }  
 outvector2(number);  
 vector2checker(number);  
}  
  
**void** VectorClass::FileEmpty()  
{  
 outfile = fopen(**"out.txt"**, **"w+"**);  
 fclose(outfile);  
}  
  
**int** main()  
{  
 srand(time(NULL));  
 VectorClass\* KOTobj;  
  
 KOTobj = **new** VectorClass;  
 KOTobj->FileEmpty();  
 KOTobj->initvector1();  
 KOTobj->outvector1();  
  
 KOTobj->initvector2(1.0);  
 KOTobj->initvector2(0.0);  
 KOTobj->initvector2(5.0);  
  
 cout << **"\n"**;  
 **return** 0;  
}

Скриншоты программы:

