

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

|  |  |
| --- | --- |
| **Институт  информационных технологий** | **Кафедра информационных систем** |

|  |  |  |
| --- | --- | --- |
| **Основная образовательная программа 09.03.02 «Информационные системы и технологии»**  **Отчет по дисциплине «Объектно-ориентированное программирование на языке высокого уровня»** | | |
| **по лабораторной работе №1** | | |
|  | | |
|  | | |
| **Проверил**  **ассистент** |  | **Михайлуца Н.А.** |
|  |  |  |
| **Выполнил**  **студент группы ИДБ-22-06** |  | **Мустафаева П.М.** |

ОГЛАВЛЕНИЕ

[ЗАДАНИЕ 1 3](#_Toc147596591)

[1.1 ОПИСАНИЕ 3](#_Toc147596592)

[1.2 ТАБЛИЦА 1 ИСПОЛЬЗОВАННЫЕ ПЕРЕМЕННЫЕ 3](#_Toc147596593)

[1.3 ПРОГРАММА 3](#_Toc147596594)

[ЗАДАНИЕ 2 6](#_Toc147596595)

[2.1 ОПИСАНИЕ 6](#_Toc147596596)

[2.2 ТАБЛИЦА 2 ИСПОЛЬЗОВАННЫЕ ПЕРЕМЕННЫЕ 6](#_Toc147596597)

[2.3 ПРОГРАММА 6](#_Toc147596598)

[2.4 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ 8](#_Toc147596599)

[ЗАКЛЮЧЕНИЕ 9](#_Toc147596600)

# **ЗАДАНИЕ 1**

## **1.1 ОПИСАНИЕ**

Вручную создать текстовый файл со следующей информацией: в две колонки, разделяя их пробелом, марка машины — модель. Например: Мерседес 320. (всего примерно 10 строк в файле, марка и модель могут повторяться).

В программе загрузить файл, распределяя данные по объектам, тип которых соответствует марке автомобиля, а модель — член данных в базовом классе. Храниться объекты должны совместно, используя контейнер vector. Вывести данные о модели и марке каждого автомобиля при выходе из программы. (Подсказка: в деструкторах).

## **1.2 ТАБЛИЦА 1 ИСПОЛЬЗОВАННЫЕ ПЕРЕМЕННЫЕ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя переменной** | **Класс** | **Тип** | **Смысл** |
| info | Локальный | string | Информация о модели и марке автомобиля |
| model | Локальный | string | Модель автомобиля |
| i | Локальный | int | Итерации цикла |

## **1.3 ПРОГРАММА**

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

using namespace std;

class Car {

public:

Car() {}

virtual ~Car() {}

};

class Mersedes : public Car {

string model;

public:

Mersedes(string car) {

model = car;

}

~Mersedes() {

cout << "Mersedes " << model << endl;

}

};

class BMW : public Car {

string model;

public:

BMW(string car) {

model = car;

}

~BMW() {

cout << "BMW " << model << endl;

}

};

class Audi : public Car {

string model;

public:

Audi(string car) {

model = car;

}

~Audi() {

cout << "Audi " << model << endl;

}

};

int main() {

setlocale(LC\_ALL, "Russian");

vector <Car\*> c;

ifstream myFile("car.txt");

string info;

cout << "Модель и марка автомобилей: " << endl;

while (!myFile.eof()) {

myFile >> info;

if (info == "Mersedes") {

myFile >> info;

c.push\_back(new Mersedes(info));

}

else if (info == "BMW") {

myFile >> info;

c.push\_back(new BMW(info));

}

else if (info == "Audi") {

myFile >> info;

c.push\_back(new Audi(info));

}

}

for (int i = 0; i < c.size(); i++) {

delete c[i];

}

\_CrtDumpMemoryLeaks();

}

**1.4 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ**

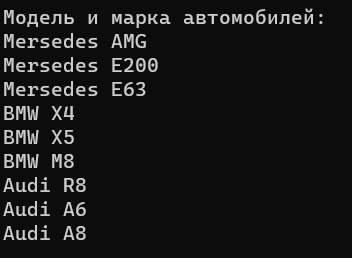


Рис. 1. Результат выполнения программы 1

# **ЗАДАНИЕ 2**

## **2.1 ОПИСАНИЕ**

Создать проект консольное приложение.

Задать два не абстрактных класса, связанных в полиморфную иерархию: Class1 (базовый) и Class2. В главной функции программы создать несколько автоматических объектов этих типов. Затем посредством вызова функции my\_copy скопировать эти объекты в базу данных (использовать vector), т. е. в базе данных должны оказаться копии созданных автоматических объектов.

Далее, вывести на консоль информацию о типе хранимых в БД объектах, используя typeid.

Исключить утечку памяти.

## **2.2 ТАБЛИЦА 2 ИСПОЛЬЗОВАННЫЕ ПЕРЕМЕННЫЕ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя переменной** | **Класс** | **Тип** | **Смысл** |
| a | Локальный | int | Инициализация объекта класса Class1 |
| b | Локальный | double | Инициализация объекта класса Class2 |
| i | Локальный | int | Итерации цикла |

## **2.3 ПРОГРАММА**

#include <iostream>

#include <vector>

#include <typeinfo>

using namespace std;

class Class1 {

int a;

public:

Class1() {

a = 0;

}

Class1(int a1) {

a = a1;

}

Class1(const Class1& obj) {

a = obj.a;

}

virtual Class1\* my\_copy() {

Class1\* new\_obj = new Class1(\*this);

return new\_obj;

}

virtual~Class1() {}

};

class Class2 : public Class1

{

double b;

public:

Class2()

{

b = 0.0;

}

Class2(double b1) {

b = b1;

}

Class2(const Class2& obj) : Class1(obj) {

b = obj.b;

}

Class2\* my\_copy() {

Class2\* new\_obj = new Class2(\*this);

return new\_obj;

}

~Class2() {}

};

int main() {

vector <Class1\*> tap;

int a = 4; double b = 6.8;

Class1\* a1 = new Class1(a);

Class2\* b1 = new Class2(b);

tap.push\_back(a1->my\_copy());

tap.push\_back(b1->my\_copy());

for (int i = 0; i < tap.size(); i++) {

cout << typeid(tap[i]).name() << endl;

}

for (int i = 0; i < tap.size(); i++) {

delete tap[i];

}

delete a1;

delete b1;

return 0;

}

## **2.4 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ**

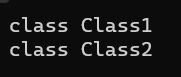


Рис. 2. Результат выполнения кода 2

# **ЗАКЛЮЧЕНИЕ**

В данной лабораторной работе была проведена работа со считыванием файлов и выведением данных из файла в консоль. Так же работа с базами данных и создание консольного приложения.