

**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 №6** | | |
|  | | |
|  | | |
| **Проверил**  **ассистент** |  | **Михайлуца Н.А.** |
|  |  |  |
| **Выполнил**  **студент группы ИДБ-22-06** |  | **Мустафаева П.М.** |

ОГЛАВЛЕНИЕ

[ЗАДАНИЕ 1 2](#_Toc152692935)

[1.1 ОПИСАНИЕ 2](#_Toc152692936)

[1.2 ПРОГРАММА 3](#_Toc152692937)

[1.3 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ 7](#_Toc152692938)

[ЗАКЛЮЧЕНИЕ 7](#_Toc152692939)

# **ЗАДАНИЕ 1**

## **1.1 ОПИСАНИЕ**

Самостоятельно, используя notepad, создать файл со следующей информацией внутри (указан вид товара и его наличное количество через пробел в магазине «Спорт»):

футболка 258

бейсболка 512

мяч 120

рюкзак 89

кеды 120

велосипед 0

мяч 70

мяч 130

Задать полиморфную иерархию из нужного числа классов, наследующих общему (не чистому) абстрактному классу (придумайте ему название), где каждый класс соответствует определенному виду товара. В классе товара определить член-данных для хранения количества экземпляров этого товара на складе. Кроме того, задать глобальное хранилище, используя контейнер vector, для сохранения внутри него объектов типов созданной иерархии.

Создать также хранилище типа multimap для хранения пар: (количество товара, вид товара).

Затем в главной функции программы прочитать созданный файл с информацией о наличии товаров на складе магазина, по ходу создавая объект соответствующего типа и помещая его в хранилище.

Затем заполнить второе хранилище, взяв нужные данные из первого. Далее вывести по порядку накопленные в multimap данные на консоль. Также вывести только наименьшее количество товаров на складе, если имеются повторения товаров. Убедиться, что информация выводится в порядке возрастания количества товара на складе.

В конце освободить ресурсы.

## **1.2 ПРОГРАММА**

#include <iostream>

#include <fstream>

#include <vector>

#include <map>

#include <string>

using namespace std;

class Product {

public:

int count;

string name;

Product() {}

virtual ~Product() {}

};

class Tshirt : public Product {

public:

Tshirt(int a) {

count = a;

name = "Футболка";

}

~Tshirt() {}

};

class Cap : public Product {

public:

Cap(int a) {

count = a;

name = "Бейсболка";

}

~Cap() {}

};

class Ball : public Product {

public:

Ball(int a) {

count = a;

name = "Мяч";

}

~Ball() {}

};

class Backpack : public Product {

public:

Backpack(int a) {

count = a;

name = "Рюкзак";

}

~Backpack() {}

};

class Sneakers : public Product {

public:

Sneakers(int a) {

count = a;

name = "Кеды";

}

~Sneakers() {}

};

class Bicycle : public Product {

public:

Bicycle(int a) {

count = a;

name = "Велосипед";

}

~Bicycle() {}

};

int main() {

setlocale(LC\_ALL, "rus");

ifstream myFile("товары.txt");

if (!myFile) {

cout << "Файл не найден" << endl;

return 1;

}

vector<Product\*> sklad;

multimap<int, string> count\_product;

string view;

int count;

bool outBall = false;

bool outTshirt = false;

bool outCap = false;

bool outBackpack = false;

bool outSneakers = false;

bool outBicycle = false;

while (myFile >> view >> count) {

if (view == "футболка") {

sklad.push\_back(new Tshirt(count));

}

else if (view == "бейсболка") {

sklad.push\_back(new Cap(count));

}

else if (view == "мяч") {

sklad.push\_back(new Ball(count));

}

else if (view == "рюкзак") {

sklad.push\_back(new Backpack(count));

}

else if (view == "кеды") {

sklad.push\_back(new Sneakers(count));

}

else if (view == "велосипед") {

sklad.push\_back(new Bicycle(count));

}

}

myFile.close();

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

count\_product.insert(make\_pair(sklad[i]->count, sklad[i]->name));

}

for (auto it = count\_product.begin(); it != count\_product.end(); ++it) {

if (it->second == "Мяч" && !outBall) {

cout << it->second << " " << it->first << endl;

outBall = true;

}

if (it->second == "Футболка" && !outTshirt) {

cout << it->second << " " << it->first << endl;

outTshirt = true;

}

if (it->second == "Бейсболка" && !outCap) {

cout << it->second << " " << it->first << endl;

outCap = true;

}

if (it->second == "Рюкзак" && !outBackpack) {

cout << it->second << " " << it->first << endl;

outBackpack = true;

}

if (it->second == "Кеды" && !outSneakers) {

cout << it->second << " " << it->first << endl;

outSneakers = true;

}

if (it->second == "Велосипед" && !outBicycle) {

cout << it->second << " " << it->first << endl;

outBicycle = true;

}

}

return 0;

}

## **1.3 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ**

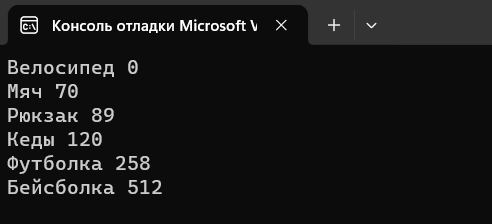


Рис. 1. Результат выполнения программы

# **ЗАКЛЮЧЕНИЕ**

В ходе лабораторной работы была реализована работа с открытием и чтением информации из файла, а так же с multimap.