:Министерство образования и науки Российской Федерации

Федеральное государственное образовательное учреждение   
высшего профессионального образования

«Алтайский Государственный Технический университет им. И. И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Ответ защищён с оценкой \_\_\_\_\_\_

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Троицкий В.С.

Отчет

по лабораторной работе №3

**«Классы С++»**

по дисциплине «Программирование»

Студент группы ПИ-91 Иванова Ольга

Преподаватель доцент, к.т.н. Троицкий В.С.

Барнаул 2021

**Задание**

Разработка программного модуля, описывающего работу книжного магазина с использованием классов языка C++.

Используемые структуры:

* Books – класс, наследник от bookshop

Поля(private):

* + - char title [] – название произведения, строка
    - int year – год издания произведения, целое число
    - double price – цена произведения, вещественное число

Методы(public):

* + - books() – конструктор
    - books(double profit, double loss, double packaging, char name[LENGHT], int year, double price) : bookshop(profit, loss, packaging) – второй конструктор
    - void input() – ввод данных полей класса
    - void output() – вывод данных полей класса
    - void add\_price(double sum) – сложение по полю цена
    - void packaging\_() – описывает процесс упаковки книг; при работе функции увеличиваются убытки магазина (loss)
    - void sale() – описывает процесс продажи книг, при котором увеличивается прибыль(profit) на их цену(book.price).
* Bookshop – родительский класс

Поля(private):

* + - double profit – прибыль магазина, вещественное число
    - double loss – убытки магазина, вещественное число
    - double packaging – цена на упаковку (фиксированное значение, предполагается использование одного вида упаковки), вещественное число.
    - Books book – вложенная структура, для передачи доступа к полям, данной структуры.

Методы(public):

* + - bookshop() – конструктор
    - bookshop(double profit, double loss, double packaging) – конструктор
    - void set\_packaging(double packaging) – сеттер
    - friends books – делает класс books дружественным

***Выполнение***

//Разработка программного модуля, описывающего работу книжного магазина с использованием классов языка C++.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <string.h>

#include<malloc.h>

#include <cstdlib>

#include <stdio.h>

#define LENGHT 100

#define standart "no\_name"

#define price\_packaging 10

using namespace std;

class books;

//родительский класс

class bookshop

{

public:

//конструктор

bookshop()

{

profit = 0;

loss = 0;

packaging = price\_packaging;

}

//конструктор

bookshop(double profit, double loss, double packaging)

{

this->profit = profit;

this->loss = loss;

this->packaging = packaging;

}

//сеттрер

void set\_packaging(double packaging)

{

this->packaging = packaging;

}

friend books;

private:

double profit;

double loss;

double packaging;

};

class books : public bookshop

{

public:

//конструктор

books()

{

strcpy\_s(title, standart);

year = 0;

price = 0;

}

//конструктор

books(double profit, double loss, double packaging, char name[LENGHT], int year, double price) : bookshop(profit, loss, packaging)

{

strcpy\_s(this->title, name);

this->year = year;

this->price = price;

}

//ввод данных полей класса

void input()

{

cout << "Input title: \t\t\t";

rewind(stdin);

gets\_s(title);

cout << "Input year of publication: \t";

cin >> year;

cout << "Input price of book: \t\t";

cin >> price;

}

//вывод данных полей класса

void output()

{

cout << "Store profit: \t\t\t" << profit << endl;

cout << "Store losses: \t\t\t" << loss << endl;

cout << "Package price: \t\t\t" << packaging << endl;

cout << "Book's title: \t\t\t" << title << endl;

cout << "Book's years of publication: \t" << year << endl;

cout << "Book's price: \t\t\t" << price << endl;

}

//сложение

double add\_price(double sum, double loss)

{

sum += price;

loss += packaging;

cout << "\nTotal price of books: \t\t" << sum;

cout << "\nTotal loss: \t\t\t"<< loss <<endl;

return sum;

}

//упаковка

void packaging\_()

{

loss = loss + packaging;

cout << "The book is packed" << endl;

}

//продажа

void sale()

{

profit += price;

loss -= packaging;

strcpy\_s(title, standart);

year = 0;

price = 0;

}

private:

char title[LENGHT];

int year;

double price;

};

int main()

{

//работа со статическими объектами

double sum = 0, sum1 = 0, sum2 = 0, sum3 =0, sum4 = 0, sum5 = 0;

double loss = 0, size = 0;

char st[LENGHT] = "no\_name";

books book1(0, 0, 10, st, 0, 0);

books book2(0, 0, 10, st, 0, 0);

cout << "Working with static class object:" << endl << endl;

cout << "Initialization" << endl;

book1.output();

cout << endl << "Input information about 1 book" << endl;

book1.input();

cout << endl << "Book packaging:" << endl;

book1.output();

book1.packaging\_();

cout << endl << "Book sales:" << endl;

book1.output();

cout << endl << "Input information about 2 book" << endl;

book2.input();

cout << endl << "Book packaging:" << endl;

book2.output();

book2.packaging\_();

cout << endl << "Book sales:" << endl;

book2.output();

sum = book1.add\_price(sum, loss);

sum1 = book2.add\_price(sum, loss);

cout << endl << endl << "After sale 1 book:" << endl;

book1.sale();

book1.output();

cout << endl << "After sale 2 book:" << endl;

book2.sale();

book2.output();

//работа с динамическими объектами

books\* book3 = new books(0, 0, 10, st, 0, 0);

books\* book4 = new books(0, 0, 10, st, 0, 0);

cout << endl << "Working with dinamic class object:" << endl << endl;

cout << "Initialization" << endl;

book3->output();

cout << endl << "Input information about 1 book" << endl;

book3->input();

cout << endl << "Book packaging:" << endl;

book3->output();

book3->packaging\_();

cout << endl << "Book sales:" << endl;

book3->output();

cout << endl << "Input information about 2 book" << endl;

book4->input();

cout << endl << "Book packaging:" << endl;

book4->output();

book4->packaging\_();

cout << endl << "Book sales:" << endl;

book4->output();

sum2 = book3->add\_price(sum2, loss);

sum3 = book4->add\_price(sum2, loss);

cout << endl << endl << "After sale 1 book:" << endl;

book3->sale();

book3->output();

cout << endl << "After sale 2 book:" << endl;

book4->sale();

book4->output();

delete book3;

delete book4;

//работа с массивом динамических объектов

cout << endl << "Working with array" << endl;

books\*\* booky = new books \* [LENGHT];

cout << endl << "Input numbers of books:\t\t";

cin >> size;

cout << endl << "Input information about books:" << endl;

for (int i = 0; i < size; i++)

{

booky[i] = new books(0, 0, 10, st, 0, 0);

booky[i]->input();

cout << endl;

}

cout << "Book packaging:" << endl;

for (int j = 0; j < size; j++)

{

booky[j]->output();

booky[j]->packaging\_();

cout << endl;

}

cout << "Book sales:" << endl;

for (int j = 0; j < size; j++)

{

booky[j]->output();

cout << endl;

}

cout << "Parts of lesses:";

for (int j = 0; j < size; j++)

{

sum4 = booky[j]->add\_price(sum4, loss);

}

cout << endl << "After sale books:" << endl;

for (int j = 0; j < size; j++)

{

booky[j]->sale();

booky[j]->output();

cout << endl;

}

delete[] booky;

//работа с диинамическим массивом

cout << endl << "Working with dinamic array" << endl;

cout << endl << "Input numbers of books:\t\t";

cin >> size;

books\* booky1 = (books\*)calloc(size,sizeof(books));

cout << endl << "Input information about books:" << endl;

for (int i = 0; i < size; i++)

{

booky1[i].input();

booky1[i].set\_packaging(price\_packaging);

cout << endl;

}

cout << "Book packaging:" << endl;

for (int j = 0; j < size; j++)

{

booky1[j].output();

booky1[j].packaging\_();

cout << endl;

}

cout << "Book sales:" << endl;

for (int j = 0; j < size; j++)

{

booky1[j].output();

cout << endl;

}

cout << "Parts of lesses:";

for (int j = 0; j < size; j++)

{

sum5 = booky1[j].add\_price(sum5, loss);

}

cout << endl << "After sale books:" << endl;

for (int j = 0; j < size; j++)

{

booky1[j].sale();

booky1[j].output();

cout << endl;

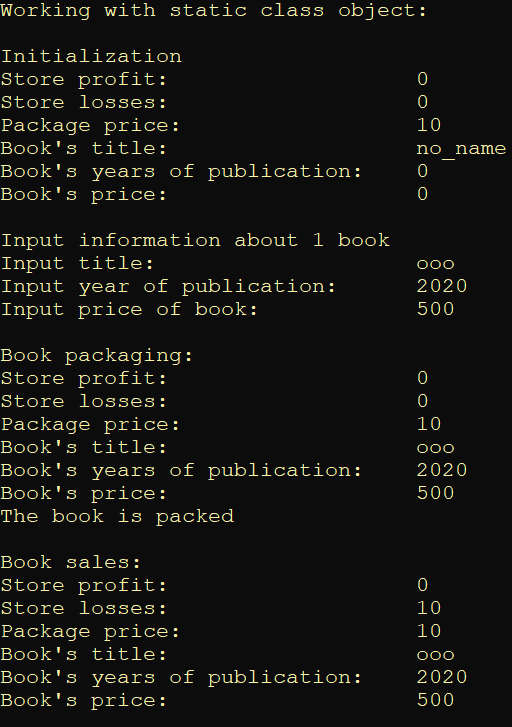
}

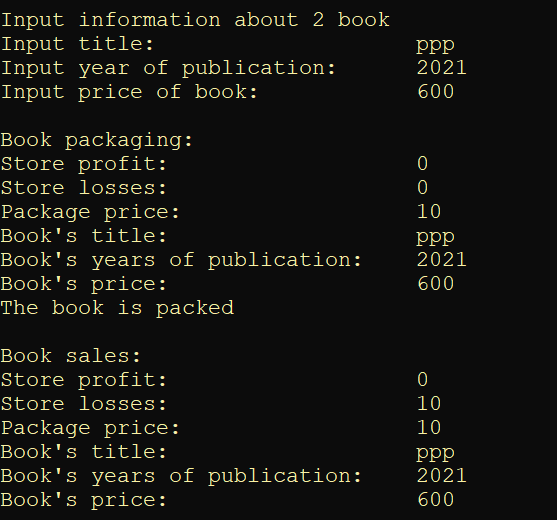
free(booky1);

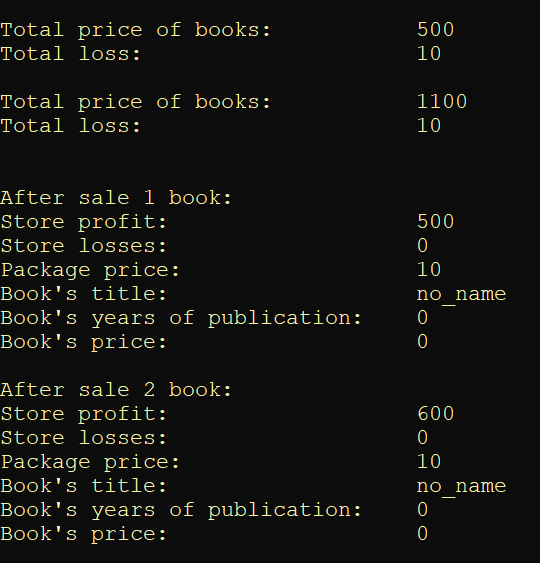
return 0;

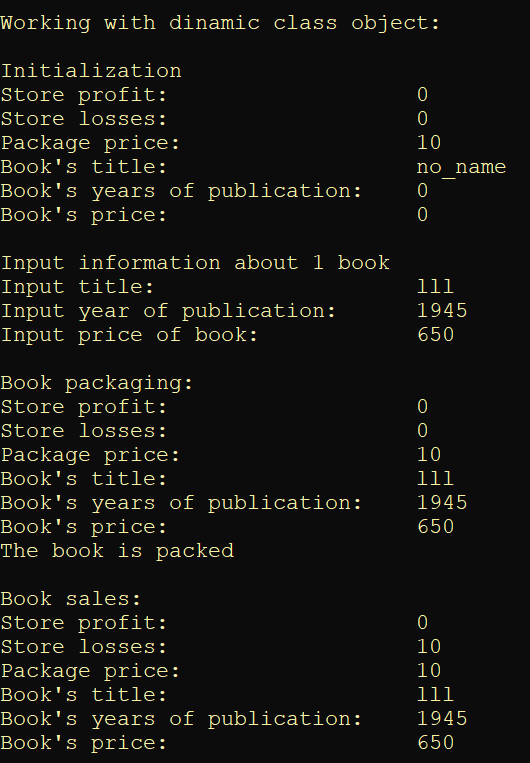
}

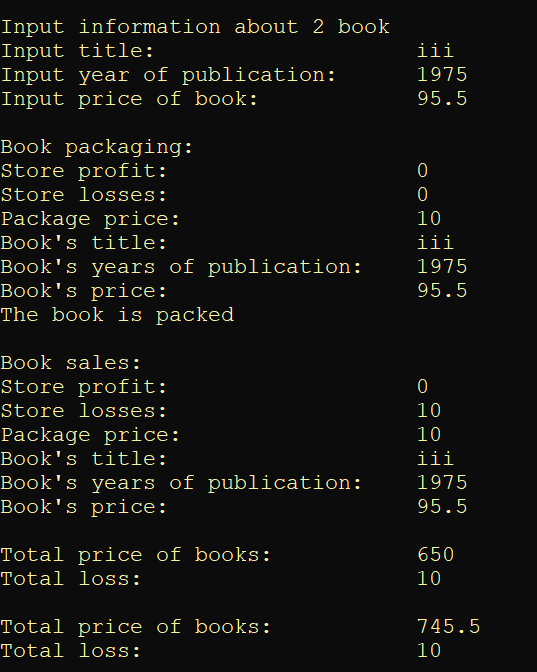
***Тесты***

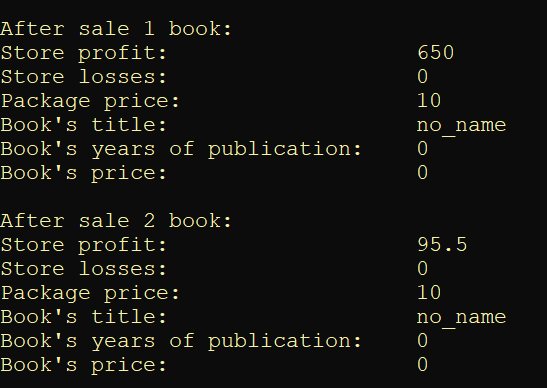
******

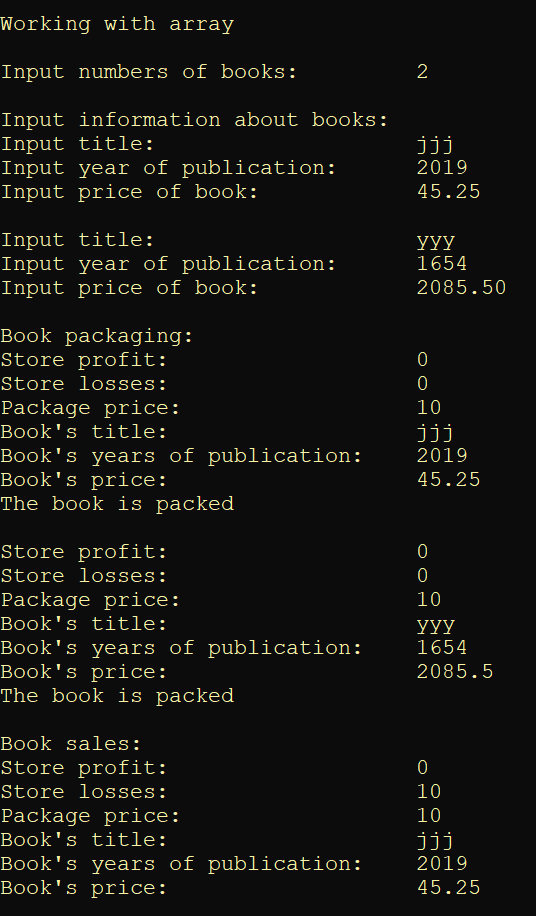
******

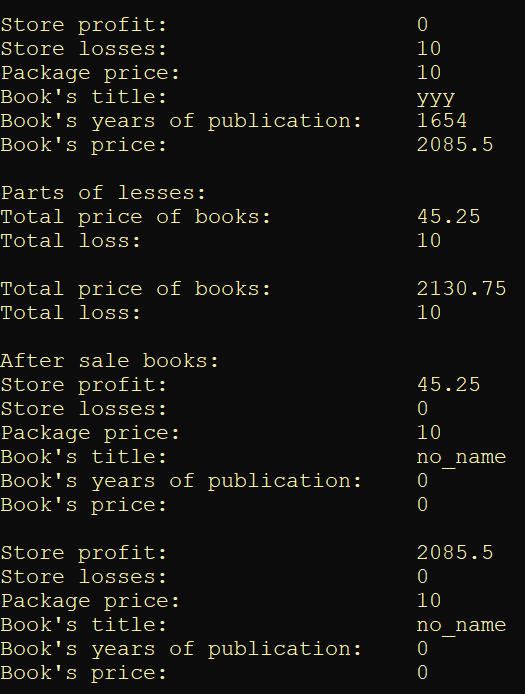
******

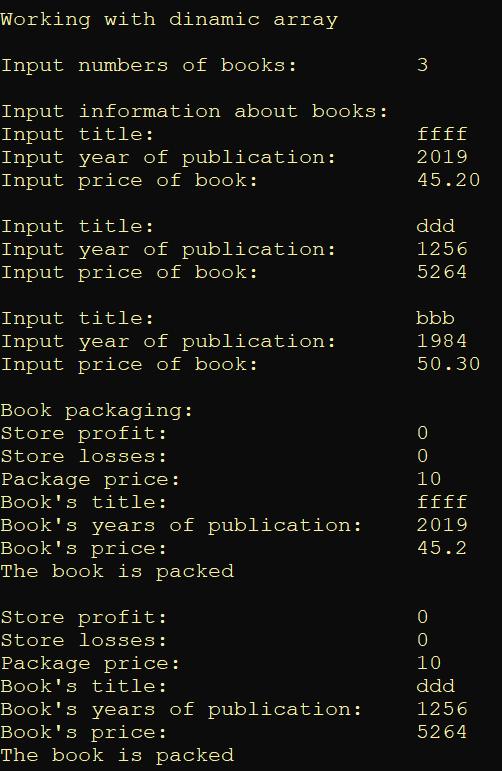
******

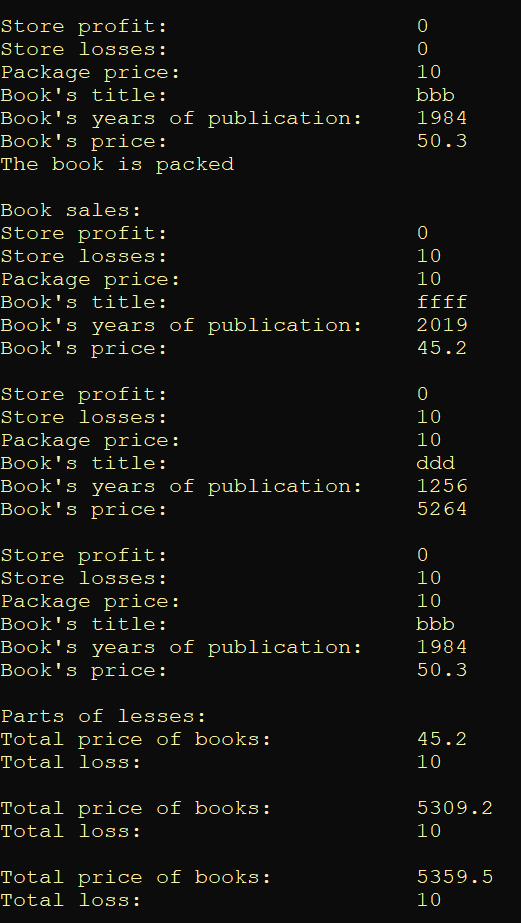
******

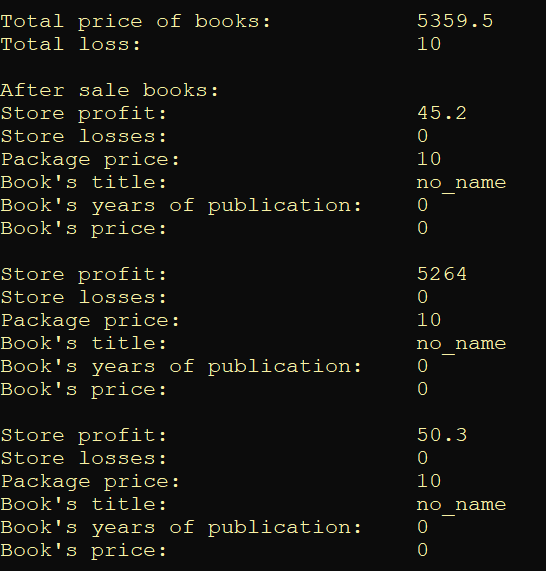
******

******

******

******





***Cсылка на GitHub:***

https://github.com/OlgaIvanova1804/lr3.git