Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им. И.Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление: 710400 «Программная инженерия»

Дисциплина: «Объектно – ориентированное программирование»

**ОТЧЕТ**

По лабораторной работе №3.

Тема: «Классы и объекты»

                                                                       Выполнила: студент группы

ПИ(б)-1-22 Кулбачаев Т.К

Проверила: Мусабаев Э.Б

**Бишкек – 2022**

**Лабораторная работа №3.**

**Тема: Конструктор и деструктор**

1)Разработайте программу с классом **Timer**, в которой таймер действует как таймер обратного отсчета. При создании объекта типа **timer** ему присваивается начальное значение времени. В результате вызова функции **run()** таймер начинает отсчет в сторону уменьшающихся значений, пока не достигнет значения 0, после чего зазвонит звонок. **Создайте 3 варианта конструктора, т.е. конструктор перегрузите 3 раза** для того, чтобы можно было указывать время в секундах с помощью целого числа или строки, или в минутах и секундах, если указываются два целых числа.

Эта программа использует библиотечную функцию **clock(),** возвращающую число тиков, прошедших с момента запуска программы. Поделив это значение на макрос **CLK\_TCK**, получаем значение в секундах. Прототипы для **clock**() и **CLK\_TCK** содержатся в заголовочном файле time.h.

#include <iostream>  
#include <string>  
#include "windows.h"  
#include <sstream>  
#include <chrono>  
#include <thread>  
  
using namespace std;  
  
  
class MyTimer{  
 int seconds;  
  
public:  
 MyTimer(int ent\_second){  
 seconds = ent\_second;  
 }  
  
 MyTimer(string &ent\_txt){  
 int sec;  
 istringstream isd(ent\_txt);  
 isd >> sec;  
 seconds = sec;  
 }  
  
 MyTimer(string &ent\_txt\_first, string &ent\_txt\_second){  
 int sec, min;  
 istringstream iss(ent\_txt\_first);  
 iss >> min;  
 istringstream isd(ent\_txt\_second);  
 isd >> sec;  
 sec += min \* 60;  
 seconds = sec;  
 }  
 int getSecond(){  
 return seconds;  
 }  
  
 void run() {  
 while (seconds > 0) {  
 cout << "Осталось времени: " << seconds << " секунд." << endl;  
 this\_thread::sleep\_for(chrono::seconds(1));  
 seconds--;  
 }  
  
 cout << "Звонок! Таймер истек." << endl;  
 }  
};  
  
  
  
int main() {  
 SetConsoleOutputCP(CP\_UTF8);  
 string min = "1", sec = "15";  
 MyTimer obj(sec);  
 cout << "Секунды: " << obj.getSecond() << endl;  
 obj.run();  
 return 0;  
}

2) Создайте класс **Nomenclature**, описывающий товары на складе магазина. Закрытыми элементами класса будут: название товара, оптовая цена, розничная наценка и количество товаров на складе. Включите в класс открытые функции подсчета возможного чистого дохода при продаже этого товара и вывода всех данных о товаре на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.

#include <iostream>  
#include <string>  
#include "windows.h"  
#include <cmath>  
  
using namespace std;  
  
  
class Nomenclature{  
 string name;  
 float cost\_price;  
 float wholesale\_price;  
 float retail\_markup;  
 int items\_amount;  
 float tax\_percentage;  
  
public:  
 float calculation\_income(int quantity = 1){  
 float revenue;  
 if(quantity > 10){  
 revenue = wholesale\_price \* quantity;  
 }  
 else{  
 revenue = (cost\_price + retail\_markup) \* quantity;  
 }  
  
 return (revenue - (cost\_price \* quantity) - ((cost\_price \* tax\_percentage / 100) \* quantity));  
 }  
  
 void print\_all\_values(){  
 cout << "\t\tИнформация о товаре" << endl;  
 cout << "Название товара: " << name << endl;  
 cout << "Себестоимость: " << cost\_price << endl;  
 cout << "Оптовая цена: " << wholesale\_price << endl;  
 cout << "Розничная наценка: " << retail\_markup << endl;  
 cout << "Процент налогов = " << tax\_percentage << "%" << endl;  
 cout << "Количество товаров на складе: " << items\_amount << endl;  
 }  
  
 Nomenclature(){  
 cout << "Введите название товара: ";  
 getline(cin, name);  
  
 while (true){  
 cout << "Введите себестоимость товара: ";  
 if (cin >> cost\_price) {  
 break;  
 } else {  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 cout << "Ошибка ввода! Надо вводить число." << endl;  
 }  
 }  
  
 while (true){  
 cout << "Введите оптовую цену товара: ";  
 if (cin >> wholesale\_price) {  
 break;  
 } else {  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 cout << "Ошибка ввода! Надо вводить число." << endl;  
 }  
 }  
  
 while (true){  
 cout << "Введите розничную наценку товара: ";  
 if (cin >> retail\_markup) {  
 break;  
 } else {  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 cout << "Ошибка ввода! Надо вводить число." << endl;  
 }  
 }  
  
 while (true){  
 cout << "Введите процент налогов: ";  
 if (cin >> tax\_percentage) {  
 break;  
 } else {  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 cout << "Ошибка ввода! Надо вводить число." << endl;  
 }  
 }  
  
 while (true){  
 cout << "Введите количество товаров на складе: ";  
 if (cin >> items\_amount) {  
 break;  
 } else {  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 cout << "Ошибка ввода! Надо вводить число." << endl;  
 }  
 }  
 }  
  
 void sell(int quantity){  
 if (items\_amount > quantity)  
 items\_amount -= quantity;  
 else  
 cout << "Не хватает товаров! \n Количество на складе: " << items\_amount << endl;  
 }  
  
 ~Nomenclature(){  
 cout << "Товар " << name << " удален." << endl;  
 }  
  
};  
  
int main() {  
 SetConsoleOutputCP(CP\_UTF8);  
 Nomenclature shoes;  
 while(true) {  
 cout << "Подсчет возможного дохода - 1\tПродажа товара - 2\tПолучить список - 3\tВыйти-4" << endl;  
 cout << "Сделайте выбор: ";  
  
 int choice;  
 while (true) {  
 if (cin >> choice) {  
 break;  
 } else {  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 cout << "Ошибка ввода! Надо вводить целое число. [1, 4]." << endl;  
 }  
 }  
 bool end = false;  
 switch (choice) {  
 case 1:  
 int temp;  
 cout << "Введите количество: ";  
 cin >> temp;  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 cout << "Возможный чистый доход = " << shoes.calculation\_income(temp) << endl;  
 break;  
 case 2:  
 int temp1;  
 cout << "Введите количество: ";  
 cin >> temp1;  
 shoes.sell(temp1);  
 break;  
 case 3:  
 shoes.print\_all\_values();  
 break;  
 case 4:  
 end = true;  
 break;  
 default:  
 cout << "Нет такого варианта!!!" << endl;  
 break;  
 }  
  
 if(end){  
 break;  
 }  
 }  
  
 return 0;  
}

3) Создайте класс **Soft**, который содержит информацию об установленном программном обеспечении. Закрытыми элементами класса будут: название программы, разработчик, занимаемый объем, дата завершения лицензии. Включите в класс открытые функции подсчета количества дней до завершения лицензии и вывода всех данных об установленном программном обеспечении на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.

#include <iostream>  
#include <string>  
#include <iomanip>  
#include <ctime>  
#include "windows.h"  
#include <cmath>  
  
using namespace std;  
  
  
class Soft{  
 string name;  
 string developer;  
 float memory;  
 string licenseEnd;  
  
public:  
 Soft(){  
 cout << "Введите название программы: ";  
 getline(cin, name);  
  
 cout << "Введите имя разработчика: ";  
 getline(cin, developer);  
  
 while (true){  
 cout << "Введите объем занимаемой памяти: ";  
 if (cin >> memory) {  
 break;  
 } else {  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 cout << "Ошибка ввода! Надо вводить число." << endl;  
 }  
 }  
 }  
 void counting\_of\_daysLeft();  
 void set\_licenseEndDate();  
 void print\_properties();  
  
 ~Soft(){  
 cout << "Программа " << name << " удален." << endl;  
 }  
};  
  
int main() {  
 SetConsoleOutputCP(CP\_UTF8);  
 Soft program;  
 while(true) {  
 cout << "Вывод свойств программы - 1\tКупить лицензию - 2\tПодсчет количества дней до завершения лицензии - 3\tВыйти-4" << endl;  
 cout << "Сделайте выбор: ";  
  
 int choice;  
 while (true) {  
 if (cin >> choice) {  
 break;  
 } else {  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 cout << "Ошибка ввода! Надо вводить целое число. [1, 4]." << endl;  
 }  
 }  
 bool end = false;  
 switch (choice) {  
 case 1:  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 program.print\_properties();  
 break;  
 case 2:  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 program.set\_licenseEndDate();  
 break;  
 case 3:  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 program.counting\_of\_daysLeft();  
 break;  
 case 4:  
 end = true;  
 break;  
 default:  
 cout << "Нет такого варианта!!!" << endl;  
 break;  
 }  
  
 if(end){  
 break;  
 }  
 }  
  
 return 0;  
}  
  
void Soft :: counting\_of\_daysLeft(){  
 stringstream ss(licenseEnd);  
 int day, month, year;  
  
 ss >> day;  
 ss.ignore();  
 ss >> month;  
 ss.ignore();  
 ss >> year;  
  
 tm licenseEndDate = {};  
 licenseEndDate.tm\_mday = day;  
 licenseEndDate.tm\_mon = month - 1;  
 licenseEndDate.tm\_year = year - 1900;  
  
 time\_t currentTime = time(nullptr);  
 tm currentDate = \*localtime(&currentTime);  
  
 time\_t difference = mktime(&licenseEndDate) - mktime(&currentDate);  
  
 int daysRemaining = difference / (60 \* 60 \* 24);  
  
 cout << "Дней до окончания лицензии: " << daysRemaining << endl;  
}  
  
void Soft :: set\_licenseEndDate(){  
 cout << "Введите дату (dd.mm.yyyy): ";  
 getline(cin, licenseEnd);  
}  
  
void Soft :: print\_properties(){  
 cout << "Название: " << name << endl;  
 cout << "Разработчик: " << developer << endl;  
 cout << "Память: " << memory << endl;  
 cout << "Дата окончания лицензии " << licenseEnd << endl;  
}