**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. И. Раззакова**

**ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Кафедра: **Программное обеспечение компьютерных систем**

Дисциплина «**Объектно-Ориентированное Программирование**»

**Отчет**

**Лабораторная работа №3**

Выполнил: студент группы ПИ-2-21

Газиев Давид

Проверил: Мусабаев Э. Б.

**Бишкек 2024**

**Задание 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.

**Код программы**

#include <iostream>

#include <string>

#include <ctime>

using namespace std;

class Timer {

private:

int seconds;

public:

Timer(int secs) : seconds(secs) {}

Timer(const string& timeStr) {

int seconds = 0;

sscanf\_s(timeStr.c\_str(), "%d", &seconds);

this->seconds = seconds;

}

Timer(int mins, int secs) {

this->seconds = mins \* 60 + secs;

}

void run() {

clock\_t start = clock();

while (seconds > 0) {

if ((clock() - start) / CLOCKS\_PER\_SEC == 1) {

start = clock();

cout << "Осталось времени: " << seconds << " секунд.\n";

seconds--;

}

}

cout << "\*ЗВОНОК!\* \*ЗВОНОК!\* \*ЗВОНОК!\*\a\n";

}

};

int main() {

setlocale(LC\_ALL, "ru");

Timer timer1(10);

Timer timer2("5");

Timer timer3(1, 5);

cout << "Таймер 1 int (10 сек):" << endl;

timer1.run();

cout << "\nТаймер 2 string (5 сек):" << endl;

timer2.run();

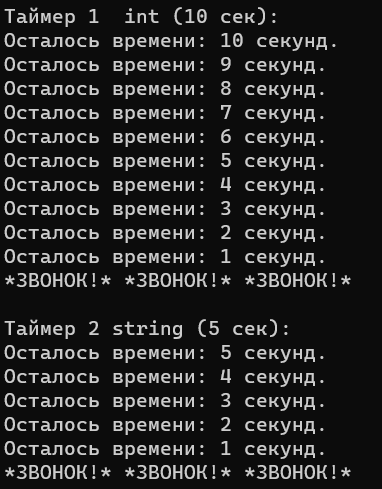
cout << "\nТаймер 3 int int (1 мин 30 сек):" << endl;

timer3.run();

return 0;

}

**Проверка выполнения**

****

**Блок схема**



**Задание 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.

**Код программы**

#include <iostream>

#include <string>

#include <vector>

#include <iomanip>

#include <windows.h>

using namespace std;

class Nomenclature {

private:

string itemName;

double wholesalePrice;

double retailMarkup;

int quantity;

public:

Nomenclature(const string& name = "", double wholesale = 0.0, double markup = 0.0, int qty = 0)

: itemName(name), wholesalePrice(wholesale), retailMarkup(markup), quantity(qty) {}

void setItem(const string& name, double wholesale, double markup, int qty) {

itemName = name;

wholesalePrice = wholesale;

retailMarkup = markup;

quantity = qty;

}

double calculateProfit() const {

double retailPrice = wholesalePrice \* (1 + retailMarkup);

return retailPrice \* quantity - wholesalePrice \* quantity;

}

bool hasItems() const {

return quantity > 0;

}

const string& getItemName() const {

return itemName;

}

double getWholesalePrice() const {

return wholesalePrice;

}

double getRetailMarkup() const {

return retailMarkup;

}

int getQuantity() const {

return quantity;

}

};

void printMenu() {

cout << "Меню:\n";

cout << "1. Ввести новый товар\n";

cout << "2. Вывести информацию о товарах\n";

cout << "0. Выйти из программы\n";

}

int main() {

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

setlocale(LC\_ALL, "ru");

vector<Nomenclature> items;

int choice;

do {

printMenu();

cout << "Введите ваш выбор: ";

cin >> choice;

switch (choice) {

case 1: {

string name;

double wholesale, markup;

int qty;

cout << "Введите название товара: ";

cin.ignore();

getline(cin, name);

cout << "Введите оптовую цену: ";

cin >> wholesale;

cout << "Введите розничную наценку (в долях, например, 0.2 для 20%): ";

cin >> markup;

cout << "Введите количество товара: ";

cin >> qty;

items.emplace\_back(name, wholesale, markup, qty);

break;

}

case 2: {

if (!items.empty()) {

cout << setw(20) << left << "Название товара" << setw(15) << "Оптовая цена" << setw(25) << "Розничная наценка (%)" << setw(15) << "Количество" << setw(20) << "Чистый доход" << endl;

cout << "---------------------------------------------------------------------------------------------------\n";

for (const auto& item : items) {

cout << setw(20) << left << item.getItemName() << setw(15) << item.getWholesalePrice() << setw(25) << item.getRetailMarkup() \* 100 << setw(15) << item.getQuantity() << setw(20) << item.calculateProfit() << endl;

}

cout << "---------------------------------------------------------------------------------------------------\n";

}

else {

cout << "Нет данных о товарах. Сначала введите товар.\n";

}

break;

}

case 0:

cout << "Выход из программы.\n";

break;

default:

cout << "Неверный выбор. Пожалуйста, выберите опцию из меню.\n";

break;

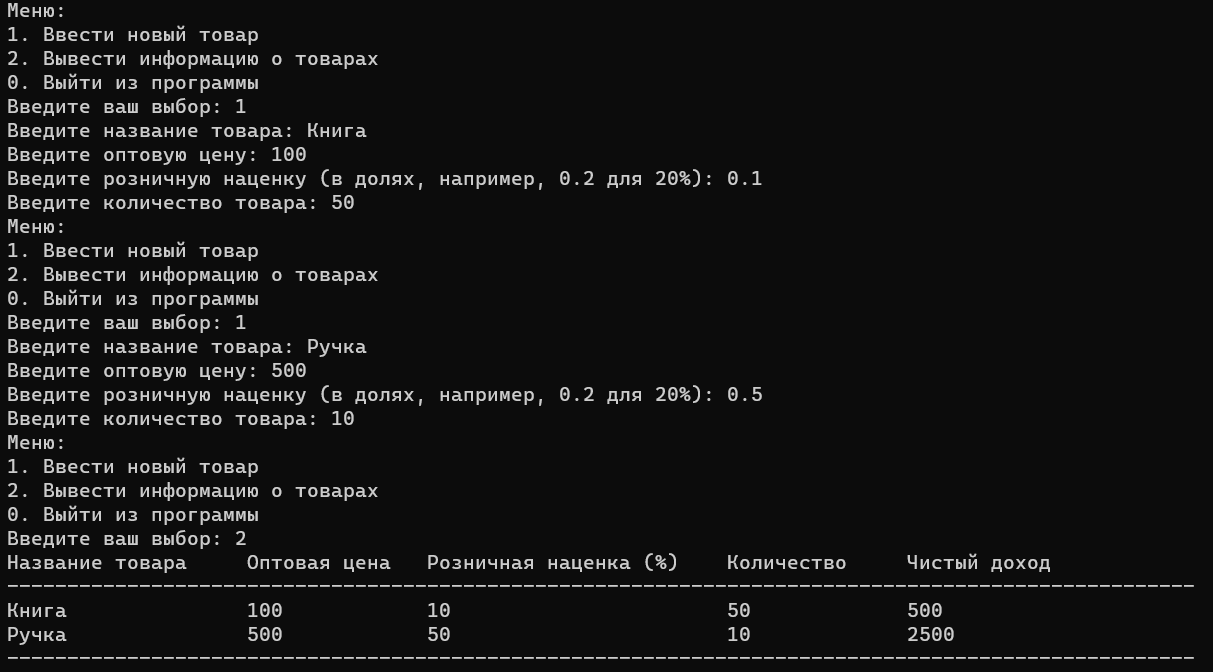
}

} while (choice != 0);

return 0;

}

**Проверка выполнения**

****

**Блок схема**



**Задание 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.

**Код программы**

#include <iostream>

#include <string>

#include <vector>

#include <iomanip>

using namespace std;

class Soft {

private:

string\* programName;

string\* developer;

double sizeInGB;

string\* licenseExpirationDate;

public:

Soft(const string& name = "", const string& dev = "", double size = 0.0, const string& expiration = "")

: programName(new string(name)), developer(new string(dev)), sizeInGB(size), licenseExpirationDate(new string(expiration)) {}

~Soft() {

delete programName;

delete developer;

delete licenseExpirationDate;

}

void display() const {

cout << "Название программы: " << \*programName << endl;

cout << "Разработчик: " << \*developer << endl;

cout << "Занимаемый объем: " << sizeInGB << " GB" << endl;

cout << "Дата завершения лицензии: " << \*licenseExpirationDate << endl;

cout << "Дней до завершения лицензии: " << daysUntilExpiration() << endl;

}

int daysUntilExpiration() const {

string currentDate = getCurrentDate();

int expirationYear, expirationMonth, expirationDay;

int currentYear, currentMonth, currentDay;

sscanf\_s(licenseExpirationDate->c\_str(), "%d-%d-%d", &expirationYear, &expirationMonth, &expirationDay);

sscanf\_s(currentDate.c\_str(), "%d-%d-%d", &currentYear, &currentMonth, &currentDay);

int expirationTotalDays = expirationYear \* 365 + expirationMonth \* 30 + expirationDay;

int currentTotalDays = currentYear \* 365 + currentMonth \* 30 + currentDay;

int daysLeft = expirationTotalDays - currentTotalDays;

return daysLeft;

}

string getProgramName() const {

return \*programName;

}

string getDeveloper() const {

return \*developer;

}

double getSize() const {

return sizeInGB;

}

string getLicenseExpDate() const {

return \*licenseExpirationDate;

}

string getCurrentDate() const {

return "2024-01-30";

}

Soft(const Soft& other)

: programName(new string(\*(other.programName))), developer(new string(\*(other.developer))),

sizeInGB(other.sizeInGB), licenseExpirationDate(new string(\*(other.licenseExpirationDate))) {}

Soft& operator=(const Soft& other) {

if (this != &other) {

\*programName = \*(other.programName);

\*developer = \*(other.developer);

sizeInGB = other.sizeInGB;

\*licenseExpirationDate = \*(other.licenseExpirationDate);

}

return \*this;

}

};

void printMenu() {

cout << "\nМеню выбора опций:\n";

cout << "1. Добавить программу\n";

cout << "2. Вывести информацию о программе\n";

cout << "0. Выход\n";

}

int main() {

setlocale(LC\_ALL, "ru");

vector<Soft> softwareList;

int choice;

do {

printMenu();

cout << "Введите номер выбранной опции: ";

cin >> choice;

switch (choice) {

case 1: {

string name, developer, expiration;

double size;

cout << "\nВведите название программы: ";

cin.ignore();

getline(cin, name);

cout << "Введите разработчика: ";

getline(cin, developer);

cout << "Введите размер программы в GB: ";

cin >> size;

cout << "Введите дату завершения лицензии (гггг-мм-дд): ";

cin >> expiration;

softwareList.emplace\_back(name, developer, size, expiration);

cout << "Программа добавлена.\n";

break;

}

case 2: {

if (softwareList.empty()) {

cout << "\nНет данных о программе. Добавьте программу сначала.\n";

}

else {

cout << "\nИнформация о программном обеспечении:\n";

cout << setw(20) << left << "Название" << setw(20) << "Разработчик" << setw(15) << "Размер (GB)" << setw(20) << "Дата завершения" << setw(15) << "Дней до завершения" << endl;

cout << "---------------------------------------------------------------------------------------------------------\n";

for (const auto& software : softwareList) {

cout << setw(20) << left << software.getProgramName() << setw(20) << software.getDeveloper() << setw(15) << software.getSize() <<

setw(20) << software.getLicenseExpDate() << setw(15) << software.daysUntilExpiration() << endl;

}

}

break;

}

case 0:

cout << "Программа завершена.\n";

break;

default:

cout << "Неверный выбор. Пожалуйста, выберите опцию из меню.\n";

break;

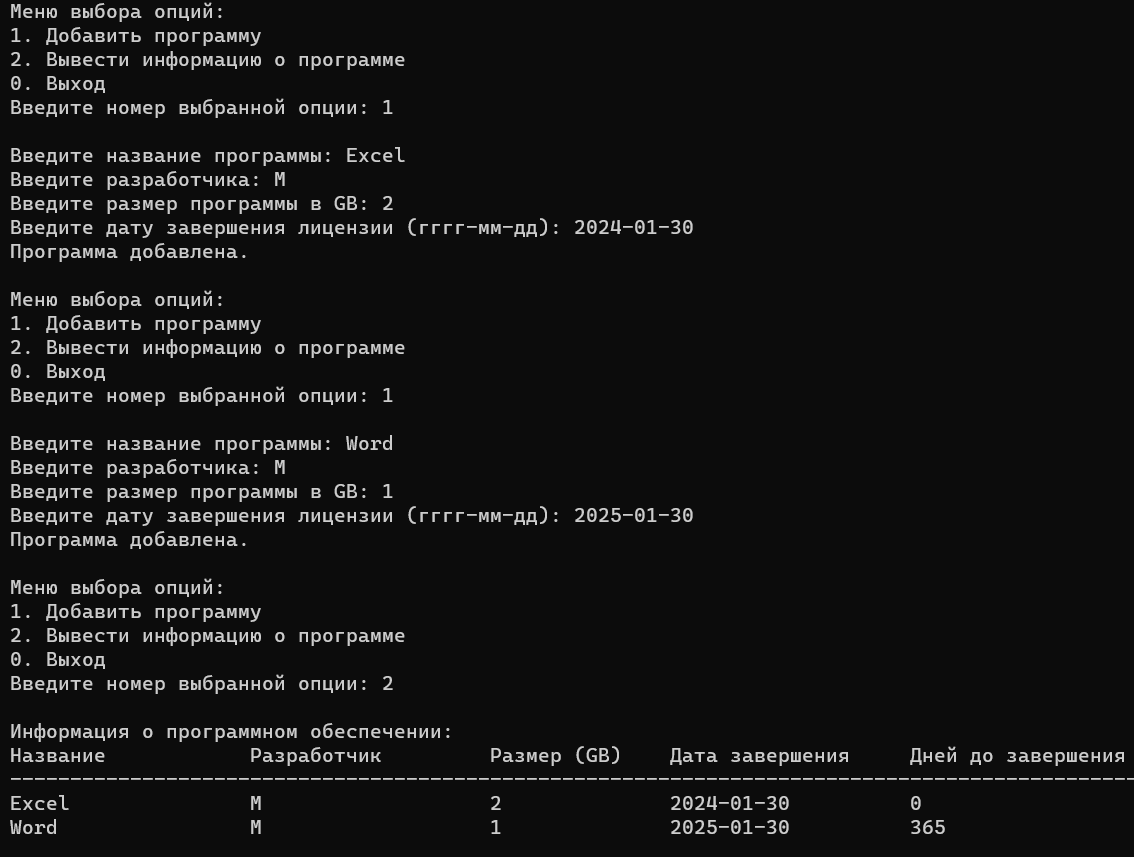
}

} while (choice != 0);

return 0;

}

**Проверка выполнения**

****

**Блок схема**

