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 С++»

**ОТЧЕТ**

**По лабораторной работе № 3**

Тема: «**Конструкторы и деструкторы**»

Выполнила: студент группы

ПИ(б)-2-21 Мааданбекова Аяна

Проверил: Мусабаев Э.Б

Бишкек-2024

**Задания:**

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.

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.

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.

**Задание 1**

**Код программы**

#include<iostream>

#include<Windows.h>

#include<stdlib.h>

#include<time.h>

using namespace std;

class timer

{

int seconds;

public:

// Секунды задаются строкой в конструкторе - перегрузка

timer(char\* t)

{

seconds = atoi(t);

}

// Секунды задаются целым числом в конструкторе- перегрузка

timer(int t)

{

seconds = t;

}

// Время задается в минутах и секундах - перегрузка

timer(int min, int sec)

{

seconds = min \* 60 + sec;

}

void run()

{

clock\_t t1, t2;

t1 = t2 = clock() / CLK\_TCK;

while (seconds) {

if (t1 / CLK\_TCK + 1 <= (t2 = clock()) / CLK\_TCK) {

seconds--;

t1 = t2;

cout << seconds << " ";

}

}

cout << "\n\a"; // звонок}

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int secInput;

cout << "Введите количество секунд: ";

cin >> secInput;

timer a(secInput);

cout << "Обратный отсчет на " << secInput << " секунд :" << endl;

a.run();

int minInput, secInput2;

cout << "Введите количество минут: ";

cin >> minInput;

cout << "Введите количество секунд: ";

cin >> secInput2;

timer b(minInput, secInput2);

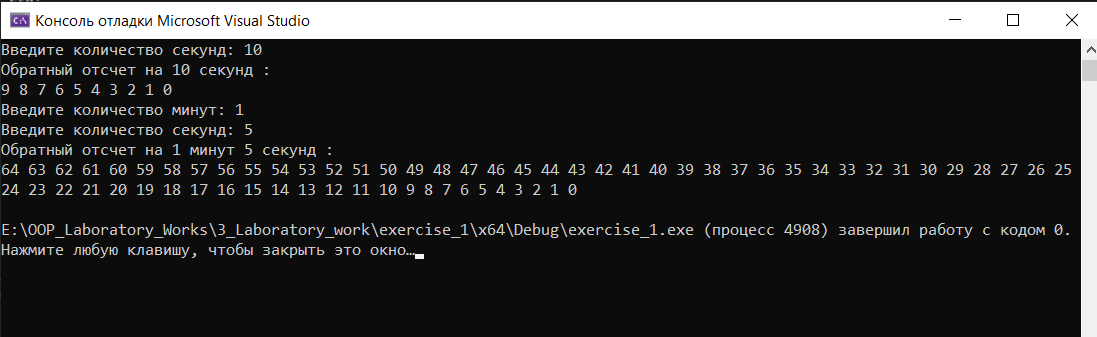
cout << "Обратный отсчет на " << minInput << " минут " << secInput2 << " секунд :" << endl;

b.run();

return 0;

}

**Результат программы**



**Задание 2**

**Код программы**

#include <iostream>

#include<Windows.h>

#include <iomanip>

using namespace std;

class Nomenclature

{

private:

// Закрытые поля класса

string Nproduct;

int Wprice;

int Rmargin;

int Cproduct = 0;

int col = 0;

public:

Nomenclature() :Nproduct(""), Wprice(0), Rmargin(0), Cproduct(0) {} // Инициализируем поля класса нулями

void input() { // Функция ввода данных

cout << "Название товара: ";

cin >> Nproduct;

cout << "Оптовая цена (сом): ";

cin >> Wprice;

cout << "Розничная наценка (сом): ";

cin >> Rmargin;

cout << "Количество товара: " << endl;

cin >> Cproduct;

col++;

}

~Nomenclature() { cout << endl << "Успешно выполнен!" << endl; } // Деструктор

void Searchprice(int n, int\* Sum) { // Функция для нахождения чистого дохода от каждого товара

float Ch\_D = 0, Val\_D = 0;

Val\_D = Wprice \* Cproduct; // Умножаем оптовую цену и количество товара

Ch\_D = Val\_D - n; // от полученного результата отнимаем налоговый платеж. Таким образом находим чистый доход от товара

cout << "Чистый доход от товара: " << "' " << Nproduct << " '" << ", составляет : " << Ch\_D << endl; // Выводим доход

\*Sum = Ch\_D; // Записываем доход в переменную

}

void Show() { // Вывод данных

cout << setw(5) << Nproduct;

cout << setw(25) << Wprice;

cout << setw(15) << Rmargin;

cout << setw(15) << Cproduct;

cout << endl;

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "RUS");

int val = 0, col = 0;

int Sum = 0;

float Sum2 = 0;

Nomenclature Obj[10]; // количество элементов

while (val != 4) {

system("CLS");

cout << "\tМеню" << endl << endl;

cout << "1) Ввод данных" << "\n2) Вывод данных" << "\n3) Вывод дохода" << "\n4) Выход" << endl << endl;

cout << "Выберите из меню: ";

cin >> val;

if (val == 1) {

Obj[col].input(); // Добавляем данные

col++; // Увеличиваем количество

cout << "Данные записаны!" << endl;

system("pause");

}

else if (val == 2) {

system("CLS");

cout << "\t\tВсе товары" << endl << endl;

cout << setw(5) << "Название товара" << setw(20) << "Оптовая цена" << setw(20) << "Розничная наценка" << setw(20) << "Количество товара" << endl;

for (int i = 0; i < col; i++) {

Obj[i].Show(); // Вывод данных

}

system("pause");

}

else if (val == 3) {

int n = 0;

system("CLS");

cout << "Введите сумму налогового платежа: ";

cin >> n;

for (int i = 0; i < col; i++) {

Obj[i].Searchprice(n, &Sum); // Вызываем функция с двумя параметрами.

Sum2 += Sum; // Суммируем доходы для вычисления общего дохода от товаров

}

cout << endl << "Общий доход: ";

cout << Sum2 << endl; // Вывод дохода

Sum2 = 0; // Обнуляем переменную

system("pause");

}

}

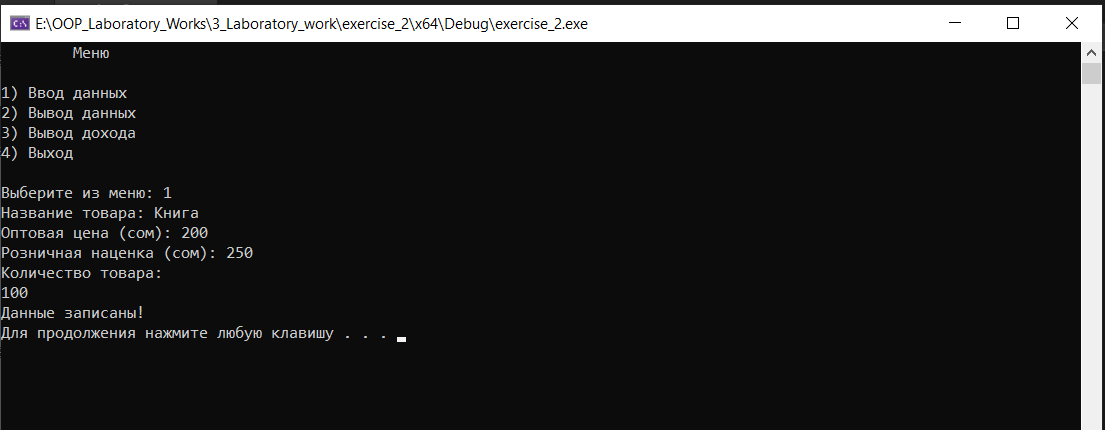
system("CLS");

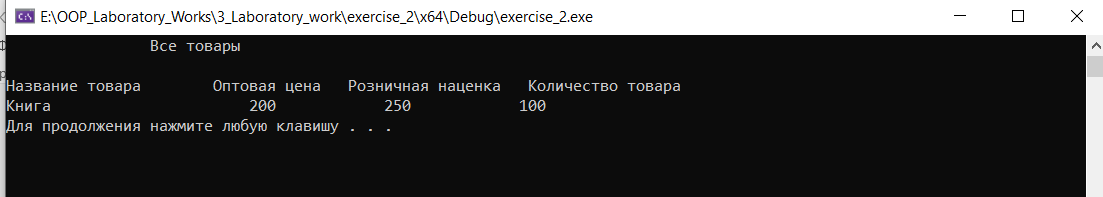
cout << "До свидания!" << endl;

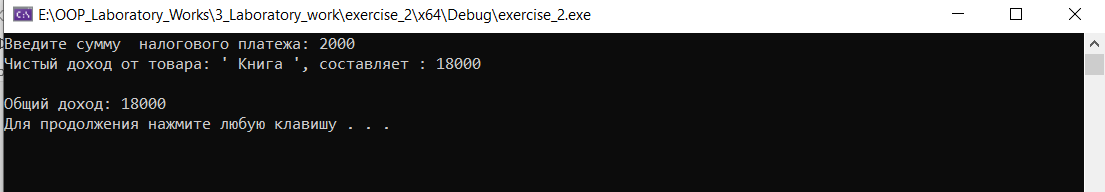
system("pause");

}

**Результат программы**







**Задание 3**

**Код программы**

#include<iostream>

#include<Windows.h>

#include<iomanip>

#include<ctime>

#include<cstdlib>

using namespace std;

class Soft

{

private:

// Закрытые поля класса

string SName;

string DName;

string SSize;

int Day;

int Month;

int Year;

public:

// Инициализируем поля класса нулями(конструктор по умолчанию

Soft() :SName(""), DName(""), SSize(""), Month(0), Day(0), Year(0) {}

// Деструктор

~Soft()

{

cout << endl << "Успешно выполнено" << endl;

}

// Функция ввода

void input()

{

cout << "Введите название программы: ";

cin >> SName;

cout << "Введите наименование разработчика: ";

cin >> DName;

cout << "Введите занимаемый объем: ";

cin >> SSize;

f:cout << "Введите дату завершения лицензии \*День, Месяц, Год\*: ";

// Ввод даты

cin >> Day >> Month >> Year;

// Проверка правильность введенных данных

if (Day > 31 || Month > 12)

{

// Если не правильно, то выводим сообщение

cout << "Введите день, не больше 31, месяц не больше 12!" << endl;

goto f;

}

}

// Функция для вычисления дня

void ColDay()

{

// для ввода даты

int DY, MN, YR;

system("CLS");

// Вводим дату

f2: cout << "Введите сегодняшнюю дату (День, Месяц, Год): ";

cin >> DY >> MN >> YR;

// Проверка введенных данных

if (DY > 31 || MN > 12)

{

cout << "Ошибка! Введите день, не больше 31 и месяц не больше 12!!" << endl;

goto f2;

}

else

{

// date\_1 - это сегодняшняя дата в секундах

time\_t date\_1 = dateSoft(DY, MN, YR);

// date\_2 - это дата окончания лицензии ПО в секундах

time\_t date\_2 = dateSoft(Day, Month, Year);

// Проверяем даты. Если дата окончания лицензии меньше сегод.даты, то выводим сообщение

if (date\_1 > date\_2)

{

cout << "Срок действия лицензии уже истек!" << endl;

}

else

{

// Если иначе, то вычисляем разницу между двумя датами

// Возвращает разность в секундах между date\_2 и date\_1

double sec = difftime(date\_2, date\_1);

// Получаем количество дней. Переменная длинного целого числа

long days = (sec / ((3600 \* 24)));

cout << "Число дней между ";

if (Day < 10)

{

cout << "0" << Day << ".";

}

else

{

cout << Day << ".";

}

if (Month < 10)

{

cout << "0" << Month << ".";

}

else

{

cout << Month << ".";

}

cout << Year << " и ";

if (DY < 10)

{

cout << "0" << DY << ".";

}

else

{

cout << DY << ".";

}

if (MN < 10)

{

cout << "0" << MN << ".";

}

else

{

cout << MN << ".";

}

cout << YR << " составляет: " << days << endl;

}

}

cout << endl << endl;

}

// Функция для получения даты из введенной даты

time\_t dateSoft(int day, int month, int year)

{

// Временная структура

tm tmp = tm();

// Поле структуры

tmp.tm\_mday = day;

// из месяца отнимаем 1, так как диапазон месяца от 0 до 11

tmp.tm\_mon = month - 1;

// Количество лет с 1900 года

tmp.tm\_year = year - 1900;

// Функция восстанавливает полную дату и время. Возвращает количество секунд из даты

return mktime(&tmp);

}

// Вывод данных

void Show()

{

cout << "\*--------------------------------------------------------------------------\*" << endl;

cout << "| Название программы | Имя разработчика | Занимаемый объем | Дата лицензии |" << endl;

cout << "\*--------------------|------------------|------------------|---------------\*" << endl;

cout << "| " << SName << setw(17) << "| " << DName << setw(10) << "| " << SSize << setw(16) << "| ";

// Условия для более кооректного вывода даты

if (Day < 10)

{

cout << "0" << Day << ".";

}

else

{

cout << Day << ".";

}

if (Month < 10)

{

cout << "0" << Month << ".";

}

else

{

cout << Month << ".";

}

cout << Year << setw(5) << "|";

cout << endl << endl;

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

// Создаем объект

Soft prog;

int value = 0;

// Пока пользователь не введет 4

while (value != 4)

{

SetConsoleTextAttribute(h, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

system("CLS");

cout << "\tМеню" << endl;

cout << "1) Ввод ПО" << "\n2) Вывод ПО" << "\n3) Вычислить количество дней" << endl;

cout << endl << "Введите номер операции: ";

cin >> value;

if (value == 1)

{

SetConsoleTextAttribute(h, FOREGROUND\_BLUE | FOREGROUND\_INTENSITY);

system("CLS");

prog.input();

cout << "Данные записаны!" << endl;

system("pause");

system("CLS");

}

else if (value == 2)

{

SetConsoleTextAttribute(h, FOREGROUND\_RED | FOREGROUND\_INTENSITY);

system("CLS");

cout << "\*--------------------------------------------------------------------------\*" << endl;

cout << "| Сведения о программном обеспечении |" << endl;

prog.Show();

system("pause");

system("CLS");

}

else if (value == 3)

{

SetConsoleTextAttribute(h, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

prog.ColDay();

system("pause");

system("CLS");

}

}

system("CLS");

cout << "До свидания!" << endl;

system("pause");

}

**Результат программы**

