КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ФУНДАМЕНТАЛЬНЫХ НАУК

КАФЕДРА ЮНЕСКО ПО ИНФОРМАЦИОННЫМ ВЫЧИСЛИТЕЛЬНЫМ ТЕХНОЛОГИЯМ

ЛАБОРАТОРНАЯ РАБОТА № 1

по дисциплине *«Языки программирования»*

*Семестр 3*

Тема: ***Структуры***

Выполнил:

МОА-195 Шепелев Семен

Проверил:

кафедры ЮНЕСКО по ИВТ

Стуколов С.В. **Регламент сдачи лабораторной работы:**

***Файлы исходного кода*** *должны выкладываться в открытом репозитории на GitHub.com. При отсутствии выложенных файлов лабораторная работа не принимается.*

***Отчет о выполнении лабораторной работы*** *сдается через систему ИнфОУПро. Отчет должен включать титульный лист, формулировку задания, примеры тестирования программы со скриншотами запуска и URL-ссылку на репозиторий на GitHub.com.*

***Защита лабораторной работы*** *выполняется устно. Для этого необходимо продемонстрировать преподавателю лабораторных занятий работу программы на тестовых наборах данных. Нужно уметь объяснять принцип работы программы.*

***Назначаются контрольные даты.*** *Лабораторная работа должна быть сдана до назначенной контрольной даты. Если лабораторная работа сдается с недельным опозданием, то от полученных балов отнимается 25%. Сданные позднее оговоренного срока лабораторные работы принимаются со штрафом в 50%.*

**Реализуйте следующие программы на языке С++.**

***Задание:***

* + - 1. Необходимо описать объект «Дата» с помощью структуры.
      2. Для структуры предусмотреть методы: вывод информации, получение следующей даты, получение предыдущей даты.
      3. В главной программе необходимо создать массив из n объектов. Инициализировать массив данными, считанными из файла.

Вывести информацию о всех объектах на экран в формате

<номер> <дата> <следующая дата> <предыдущая дата>

Количество записей в файле неизвестно.

***Задания по вариантам:***

***Вариант 7:***

Выбрать из массива исходных дат те, для которых предыдущие не попадают на другой месяц. Вывести информацию о таких датах на экран в формате:

<номер> <дата> <предыдущая дата>

***Требования к оформлению программ:***

1. **Функциональная декомпозиция кода.** Программа должна быть оформлена с помощью функций. В главной функции реализовывается только управляющий алгоритм, вспомогательные алгоритмы должны быть реализованы в дополнительных функциях.
2. **Тестирование.** Программа должна быть протестирована на разнообразных входных данных, в том числе и тех, которые могут привести к ошибочному поведению. Код программы должен содержать достаточное количество проверок на корректность входных данных.

***Критерии оценки (максимально 10 баллов):***

В случае, если в задаче получен верный результат, то за решение начисляются баллы.

|  |  |
| --- | --- |
| Правильно используются структуры для описания пользовательского типа данных в соответствии с заданием. | **2** |
| Правильно реализована работа с внешними файлами данных | **1** |
| Правильно реализована работа с динамическими массивами структур. Памяти выделяется необходимое количество. Нет утечки памяти. | **2** |
| Правильно находятся следующие даты. | **1** |
| Правильно находятся предыдущие даты. | **1** |
| Правильно реализована выборка. | **3** |

***Срок сдачи:*** две недели.

КОД:

#include <iostream>

#include <fstream>

using namespace std;

struct Date

{

int year, month, day;

void Show()

{

cout << endl << "Сейчас: " << day << "." << month << "." << year << endl;

}

//cледующий день

void Next\_date()

{

day++;

if ((month == 2) && (year % 4 == 0) && (day == 30)) // проверка на ошибку в феврале, где нет 30 дня никогда, перенос на 1 число

{

day = 1;

month = 3;

}

else if ((month == 2) && (year % 4 != 0) && (day == 29)) // проверка на високосный год, тут его нет

{

day = 1;

month = 3;

}

if ((day == 32) && ((month == 3) || (month == 5) || (month == 7) || (month == 8) || (month == 10) || (month == 1))) // проверка на неправильное число в месяцах где 31 день

{

day = 1;

month += 1;

}

if ((day == 31) && ((month == 4) || (month == 6) || (month == 9) || (month == 11))) // проверка на неправильное число в месяцах где 30 дней

{

day = 1;

month += 1;

}

else if ((day == 32) && (month == 12)) // проверка на неправильное число в декабре

{

day = 1;

month = 1;

year += 1;

}

}

//перевод на день назад

void Prev\_date()

{

day--; //день назад

if (day == 0)

{

if ((month == 3) && (year % 4 != 0)) // проверка на 1 марта в невисокосном году

{

day = 28;

month--;

}

else if ((month == 3) && (year % 4 == 0)) // проверка на 1 марта в високосном году

{

day = 29;

month--;

}

else if ((month - 1 == 3) || (month - 1 == 5) || (month - 1 == 7) || (month - 1 == 8) || (month - 1 == 10) || (month - 1 == 0)) // перенос в месяцах где предыдущие 31 день имеют

{

day = 31;

month--;

if (month == 0) // проверка на 1 января

{

month = 12;

year--;

}

}

else

{

day = 30; // месяцы где 30 дней

month--;

}

}

}

};

int main()

{

setlocale(LC\_ALL, "ru");

ifstream fin("C://data.txt");

if (!fin.is\_open())

cout << "Файл не был отрыт!" << endl;

Date temp;

int n = 0;

while (!fin.eof())

{

fin >> temp.day >> temp.month >> temp.year;

n++;

}

fin.close();

//создание массива по кол-во дат

Date\* DT;

DT = new Date[n];

fin.open("data.txt");

int i = 0;

while (!fin.eof())

{

//считывание дат в отдельный элемент массива каждой даты

fin >> DT[i].day >> DT[i].month >> DT[i].year;

i++;

}

//вывод

cout << "Запись из файла" << endl;

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

{

cout << endl << i + 1 << " - Нынешняя дата: " << DT[i].day << "." << DT[i].month << "." << DT[i].year;

DT[i].Prev\_date();

cout << " | Предыдущая: " << DT[i].day << "." << DT[i].month << "." << DT[i].year;

DT[i].Next\_date();

cout << endl;

}

//вариант 7 задание

cout << endl << endl << "Вариантное задание 8" << endl;

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

if ((DT[i].day == 1))

continue;

else

{

cout << i + 1 << ":" << DT[i].day << "." << DT[i].month << "." << DT[i].year << "-";

DT[i].Prev\_date();

cout << DT[i].day << "." << DT[i].month << "." << DT[i].year << endl;

DT[i].Prev\_date();

}

return 0;

}

<https://github.com/Semyon1234/struct_lab1>

Ссылка на Git-hub

*Кемерово, 2020*

