Министерство науки и высшего образования Российской Федерации

Новосибирский государственный технический университет

Кафедра вычислительной техники

Лабораторная работа №1

По дисциплине «Программирование»

**.**

Преподаватель: Булатов А.Д.  
Группа:  АВТ-042  
Студент: Галета Н.Е.

# Цель работы

Изучить структуру класса, механизм создания и использования, описание членов-данных класса и методов доступа к ним, возможность инициализации объектов класса с помощью конструкторов и уничтожение их с помощью деструкторов.

# Постановка задачи

Реализовать класс в соответствии с вариантом. Класс должен обеспечивать набор методов для работы с данными. Создать перегруженные конструкторы: конструктор копирования, конструктор по умолчанию, конструктор с параметрами. Реализовать в классах поля с динамическим выделением памяти для хранения некоторых данных. Создать деструктор для освобождения памяти. Изучить порядок вызова конструкторов и деструктора. Обязательно добавить в класс статические члены.

Составить демонстрационную программу, реализующую вызов всех методов класса. Для демонстрационной программы использовать отдельный модуль. Программу построить с использованием проекта. Посмотреть работу программы в отладчике, обратить внимание на представление данных. Построить программу без отладочной информации. Обратить внимание на размер программы.

Вариант 1:

Построить класс Дата-Время. Класс должен включать следующие поля: день, месяц, год, часы, минуты, строковое представление даты. Класс должен обеспечивать простейшие функции для работы с данными класса: увеличение/уменьшение на 1 день, час, минуту, секунду и т.д., методы изменения объекта, вывода на экран объекта.

# Листинг программы

Содержимое “Date.h”

#pragma once

#include <stdio.h>

#include <string>

using namespace std;

class Date

{

public:

static int a;

Date(int day, int month, int year, int hour, int minute);

Date();

Date(const Date& obj);

~Date();

void additionDay();

void additionMonth();

void additionYear();

void additionHour();

void additionMinute();

void subtractionDay();

void subtractionMonth();

void subtractionYear();

void subtractionHour();

void subtractionMinute();

void setDay(int day);

void setMonth(int mounth);

void setYear(int year);

void setHour(int hour);

void setMinute(int minute);

int getDay();

int getMonth();

int getYear();

int getHour();

int getMinute();

void checkadditionDate();

void checksubtractionDate();

void Cout();

char\* ToString();

private:

int day, month, year, hour, minute;

};

Содержимое “Date.cpp”

#include "Date.h"

#include <iostream>

#include <stdio.h>

Date::Date() {//стандарный конструктор

day = 0;

month = 0;

year = 0;

hour = 0;

minute = 0;

}

Date::Date(int day, int month, int year, int hour, int minute) {//конструктор с параметрами

if ((day < 31) && (day > 0)) {

this->day = day;

}

else {

day = 1;

}

if ((month < 13) && (month > 0)) {

this->month = month;

}

else {

month = 1;

}

this->year = year;

if ((minute < 61) && (minute > 0)) {

this->minute = minute;

}

else {

minute = 0;

}

if ((hour < 25) && (hour > 0)) {

this->hour = hour;

}

else {

hour = 0;

}

}

Date::Date(const Date& date) {//конструктор копирования

this->day = date.day;

this->month = date.month;

this->year = date.year;

this->minute = date.minute;

this->hour = date.hour;

}

Date::~Date()//деструктор

{

cout << "Memory is cleared" << std::endl;

}

void Date::additionDay()//функция добавления единицы к значению дней

{

day++;

this->day = day;

}

void Date::additionMonth()//функция добавления единицы к значению месяцев

{

month++;

this->month = month;

}

void Date::additionYear()//функция добавления единицы к значению годов

{

year++;

this->year = year;

}

void Date::additionHour()//функция добавления единицы к зачению часов

{

hour++;

this->hour = hour;

}

void Date::additionMinute()//функция добавления единицы к значени минут

{

minute++;

this->minute = minute;

}

void Date::subtractionDay()//функция вычитания единицы из значения дней

{

day--;

}

void Date::subtractionMonth()//функция вычитания единицы из значения дней

{

month--;

}

void Date::subtractionYear()//функция вычитания единицы из значения годов

{

year--;

}

void Date::subtractionHour()//функция вычитания единицы из значения часов

{

hour--;

}

void Date::subtractionMinute()//функция вычитания единицы из значения минут

{

minute--;

}

void Date::setDay(int day)

{

if (day > 30 || day < 0) { cout << "ERROR"; }

else { this->day = day; }

}

void Date::setMonth(int month)

{

if (month > 13 || month < 0) { cout << "ERROR"; }

else { this->month = month; }

}

void Date::setYear(int year)

{

this->year = year;

}

void Date::setHour(int hour)

{

if (hour > 24 || hour < 0) { cout << "ERROR"; }

else { this->hour = hour; }

}

void Date::setMinute(int minute)

{

if (minute > 0 || minute < 0) { cout << "ERROR"; }

else { this->minute = minute; }

}

int Date::getDay()

{

return day;

}

int Date::getMonth()

{

return month;

}

int Date::getYear()

{

return year;

}

int Date::getHour()

{

return hour;

}

int Date::getMinute()

{

return minute;

}

void Date::Cout() {//функция вывода на экран

cout << ToString();

}

char\* Date::ToString()

{

int date1[] = { day,month,year,minute,hour };

char string[1000] = "";

char helped[12];

for (int i = 0; i < 5; i++) {

sprintf\_s(helped, sizeof(helped), "%d", date1[i]);

strcat\_s(string, sizeof(string), helped);

strcat\_s(string, sizeof(string), " ");

}

return string;

}

Содержимое “Test.h”

#pragma once

using namespace std;

class Test

{

public:

void Test1();

void Test2();

void Test3();

};

Содержимое “Test.cpp”

#include "Test.h"

#include "Date.h"

#include <iostream>

#include <cassert>

using namespace std;

bool equal(Date& date, int a[5]) {

int i = 0;

bool k = false;

if ((a[0] == date.getDay()) && (a[1] == date.getMonth()) && (a[2] == date.getYear()) && (a[3] == date.getMinute()) && ((a[4] == date.getHour()))) {

k = true;

}

return k;

}

void Test::Test1()

{

cout << "First test\n";

int a[5] = { 3,7,2005, 34, 21 };

Date date1(3, 7, 2221, 21, 35);

date1.Cout();

cout << endl;

date1.setYear(2005);

date1.subtractionMinute();

date1.Cout();

bool t = equal(date1, a);

if (t == 1) {

cout << "true" << endl;

}

cout << endl << endl;

}

void Test::Test2()

{

cout << "Second test\n";

int b[5] = { 4,3,2001, 3, 2 };

Date date2(5, 4, 2002, 3, 3);

date2.Cout();

cout << endl;

date2.subtractionHour();

date2.subtractionDay();

date2.subtractionMonth();

date2.subtractionYear();

date2.Cout();

bool t2 = equal(date2, b);

if (t2 == 1) {

cout << "true" << endl;

}

cout << endl << endl;

}

void Test::Test3()

{

cout << "Third test\n";

Date date3(13, 11, 2002, 8, 9);

int c[5] = { 14,12,2003, 10, 9 };

date3.Cout();

cout << endl;

date3.additionHour();

date3.additionDay();

date3.additionMonth();

date3.additionYear();

date3.additionMinute();

date3.Cout();

bool t3 = equal(date3, c);

if (t3 == 1) {

cout << "true" << endl;

}

cout << endl << endl;

char ch;

std::cin >> ch;

}

Содержимое “ LB1.cpp”

#include <iostream>

#include "Date.h"

#include "Test.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

Test t = Test();

t.Test1();

t.Test2();

t.Test3();

}

**Вывод:**  
Построил класс для работы с датой и временем и научился работать с основными принципами класса. Данный класс обеспечивает простейшие функции для работы с данными: прибавление 1, вычитание 1.  
Благодаря использованию класса создается удобно структурированная система. В данной лабораторной работе использование классов облегчило работу программы.