

Указатели и наследование,
аргументы командной строки,
статические переменные – Домашнее задание 28

Пеганов Антон
peganoff2@mail.ru

25 июня 2020 г.

Задача 1

Создайте проект *fibonacci_project* с тремя файлами: *fibonacci.hpp*, *fibonacci.cpp*, *main.cpp*. С помощью глобальной переменной `count_calls` посчитайте количество вызовов функции, которая используется для рекурсивного вычисления числа Фибоначчи. Реализуйте 2 варианта вычисления числа Фибоначчи в функциях `fibonacci1()` и `fibonacci2()`.

1. Функция `fibonacci1()` должна вычислять число Фибоначчи по определению (согласно формуле $F_n = F_{n-1} + F_{n-2}$).
2. Функция `fibonacci2()` должна использовать более эффективный способ, основанный на формулах:

- $F_{2n} = F_{n+1}^2 - F_{n-1}^2$,
- $F_{2n+1} = F_n^2 + F_{n+1}^2$.

Для n в диапазоне от 1 до 40 запишите количества вызовов функций `fibonacci1()` и `fibonacci2()` в файлы *n_calls_fibonacci1.txt* и *n_calls_fibonacci2.txt*.

Для варианта 2 нарисуйте блок-схему алгоритма. Разделите ее на две независимые части: функцию `main()` и функцию `fibonacci2()`.

Задача 2

Напишите функцию, которая печатает номер своего вызова. Воспользуйтесь для этой цели статической переменной.

Задача 3

Создайте проект *datetime_project*, в котором будут два класса: `Date` и `DateTime`, причем класс `DateTime` должен наследовать классу `Date`.

Класс `Date` моделирует дату в Григорианском календаре. У объектов класса должны быть следующие признаки:

- `uint8_t day` — номер дня в месяце (принимает значения от 1 до 31),
- `uint8_t month` — номер месяца (принимает значения от 1 до 12),
- `int16_t year` — номер года (принимает любые значения, кроме 0, так как нулевого года не было: за минус первым годом сразу следует первый).

Также класс `Date` обладает следующими методами:

- конструктор по умолчанию, который инициализирует дату рождения Иисуса Христа (25.12.1);
- конструктор, принимающий на вход номера дня, месяца и года;
- `str()`, возвращающий строку с датой в формате Д.М.Г.

Класс `DateTime` объединяет в себе дату по грегорианскому календарю и время. Он обладает признаками:

- `uint8_t hour` — час,
- `uint8_t minute` — минута,
- `uint8_t second` — секунда.

Также класс `DateTime` обладает следующими методами:

- конструктор по умолчанию, который инициализирует время 00:00:00 и дату рождения Иисуса Христа (25.12.1);
- конструктор, принимающий на вход час, минуту, секунду, день, месяц, год;
- `str()`, возвращающий время и дату в формате Д.МЕСЯЦ.Г Ч:МИНУТА:С.

В методе `DateTime::str()` воспользуйтесь методом `Date::str()` так, как это показано в листинге 1.

Листинг 1: Вызов родительского метода при наличии метода с таким же именем в дочернем классе.

```
#include <iostream>

using namespace std;

class A {
public:
    void foo(){cout << "foo in A\n";}
};

class B: public A
{
public:
    void foo(){
        cout << "foo in B\n";
        A::foo(); // This is the call of the method of the base class.
    }
};
```

Напишите функцию `void write_date_time_to_file(const Date& d, const string file_name)`, которая будет дописывать в конец файла `file_name` дату, возвращаемую методом `get_string()` объекта `d`. Убедитесь, что функция правильно работает и с объектами класса `Date`, и с объектами класса `DateTime`.

Задача 4

Ознакомьтесь учебными материалами по статическим членам в папке `cpp/hw28/static_members`. Допишите в класс `Date` постоянные статические члены:

- `uint8_t n_days_in_month[12]`, содержащий количество дней по месяцам (количество дней в феврале установите равным нулю);
- `uint8_t n_days_in_year`, равный 12.

Аналогичные члены добавьте в класс `DateTime` (это будут количество часов в сутках, количество минут в часе и количество секунд в минуте).

В класс `Date` добавьте статические методы:

- `uint8_t get_n_days_in_february(int16_t year)`, возвращающий количество дней в феврале;
- `uint8_t get_n_days_in_month(uint8_t month, int16_t year)`, возвращающий количество дней в месяце (функция должна правильно работать в том числе для второго месяца, т.е. февраля).

Убедитесь в работоспособности статических методов при вызове от класса и при вызове от объекта.

Задача 5

Напишите программу, которая складывает свои аргументы (используйте переменные `argc` и `argv`).