

Комментарий к работе

По-хорошему структура и ее методы должны быть объявлены и прописаны в отдельном файле типа «хедор», например так:

```
class Roman
{
public:
    // Конструктор
    Roman(int valueNumber)
    {
        number = valueNumber;
        roman = transferRoman();
    }

    int GetNumber();    // Вернуть число в 10ой системе счисления
    string GetRoman();  // Вернуть число в римской системе счисления
    void printResult(); // Печать перехода
private:
    int number;
    string roman;

    // формируем словарь из всех римских чисел и новых комбинаций
    string all_roman[26] = { ... };
    string transferRoman();
};

int Roman::GetNumber() { ... }

string Roman::GetRoman() { ... }

void Roman::printResult() { ... }

string Roman::transferRoman() { ... }
```

Однако, работа простая, и поэтому код структуры и все ее методы будет находиться в одном файле с main() просто для удобства.

Ответ:

Консоль отладки Microsoft Visual Studio

transferRoman()		GetNumber()	GetRoman()
376	CCCLXXVI	376	CCCLXXVI
14	XIV	14	XIV
692	DCXCII	692	DCXCII
717	DCCXVII	717	DCCXVII
262	CCLXII	262	CCLXII
359	CCCLIX	359	CCCLIX
187	CLXXXVII	187	CLXXXVII
538	DXXXVIII	538	DXXXVIII
383	CCCLXXXIII	383	CCCLXXXIII
481	CDLXXXI	481	CDLXXXI

D:\Project C++\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe

Код программы:

```
#include <iostream>
#include <string>
#include <stdlib.h>

using namespace std;

class Roman
{
public:
    /* Конструктор */
    Roman(int valueNumber)
    {
        number = valueNumber;
        roman = transferRoman();
    }

    /* Возвращает число в 10-ой системе счисления */
    int GetNumber()
    {
        return number;
    }

    /* Возвращает число в римской системе счисления */
    string GetRoman()
    {
        return roman;
    }

    /* Демонстрирует (выводит в консоль) результат трансформации числа*/
    void printResult()
    {
        printf("\t%-10s", (to_string(number)).c_str());
        printf("\t%-15s", roman.c_str());
    }

private:
    /* Поля нашей структуры */
    int number;      // Число в 10-ой системе счисления
    string roman;    // Число в римской системе счисления

    /* Формируем словарь из всех римских чисел и новых комбинаций */
    string all_roman[26] = {
        "1000", "M",
        "900", "CM",
        "500", "D",
        "400", "CD",
        "100", "C",
        "90", "XC",
        "50", "L",
        "40", "XL",
        "10", "X",
        "9", "IX",
        "5", "V",
        "4", "IV",
        "1", "I" };

    /* Функция перевода в римскую систему счисления */
    string transferRoman()
    {
        // на старте в римском числе ничего нет
        string roman;
        int num = number;
```

```

        // пока наше число больше нуля
        while (num > 0)
        {
            // перебираем все пары из словаря
            for (int i = 0; i < 25; i += 2)
            {
                // пока наше число больше или равно числу из словаря
                while (num >= stoi(all_roman[i]))
                {
                    // добавляем соответствующую букву в римское число
                    roman += all_roman[i + 1];
                    // вычитаем словарное число из нашего числа
                    num -= stoi(all_roman[i]);
                }
            }

            return roman;
        }
    };

void main()
{
    setlocale(LC_ALL, "ru");
    srand(time(NULL));

    string s1 = "transferRoman()"; printf("\t%-25s", s1.c_str());
    string s2 = "GetNumber()"; printf("\t%10s", s2.c_str());
    string s3 = "GetRoman()"; printf("\t%10s\n\n", s3.c_str());

    for (int i = 0; i < 10; i++)
    {
        Roman myNumber(rand() % 1000);
        myNumber.printResult();
        printf("\t%-10s", (to_string(myNumber.GetNumber())).c_str());
        printf("\t%-10s\n", (myNumber.GetRoman()).c_str());
    }
}

```