

PCLP2

Laboratorul 4

1. **Constructori care folosesc argumente cu valori implicite.** Acest program implementează o extensie a clasei Time care folosește un constructor cu 3 parametri care au valori implicite. Programul constă din două fișiere .cpp și un fișier header cu extensia .h. De regulă, declarația unei clase se scrie într-un fișier header, definițiile funcțiilor membre ale unei clase sunt grupate într-un fișier .cpp, iar programul principal în alt fișier .cpp. Testați programul de mai jos alcătuind un proiect în mediul de programare ales de voi.

time2.h

```
#ifndef TIME2_H
#define TIME2_H
class Time
{
public:
    Time(int = 0, int = 0, int = 0); //constructor
    void setTime(int, int, int); //assignarea valorilor
    void printShort(); //tiparire in format scurt
    void printLong (); //tiparire in format lung
private:
    int hour; //0-23
    int minute; //0-59
    int second; //0-59
};
#endif
```

time2.cpp

```
#include <iostream>
using std::cout;
#include "time2.h"
Time::Time(int hr, int min, int sec)
{
    setTime(hr, min, sec);
}
void Time::setTime(int h, int m, int s)
{
    hour = (h >= 0 && h < 24) ? h : 0;
    minute = (m >= 0 && m < 60) ? m : 0;
    second = (s >= 0 && s < 60) ? s : 0;
}
void Time::printShort()
{
    cout << (hour < 10 ? "0" : "") << hour << ":"
         << (minute < 10 ? "0" : "") << minute;
}
void Time::printLong()
{
    cout << ((hour == 0 || hour == 12) ?
             12 : hour % 12)
         << ":" << (minute < 10 ? "0" : "") << minute
         << ":" << (second < 10 ? "0" : "") << second
```

```
        << (hour < 12 ? " AM" : " PM");  
    }  
}
```

test_time2.cpp

```
#include <iostream>  
using std::cout;  
using std::endl;  
#include "time2.h"  
int main()  
{  
    Time t1,          //toate argumentele implicite  
        t2(2),       //minute si second implicite  
        t3(21, 34),  //second implicit  
        t4(23, 59, 59), //toate valorile specificate  
        t5(27, 74, 99); //toate valorile eronate  
  
    cout << "Obiect creat cu: " << endl;  
    cout << "toate argumentele implicite: " << endl;  
    t1.printShort();  
    cout << endl << " ";  
    t1.printLong();  
  
    cout << "\nhour specificat; minute si second implicite:"  
        << endl << " ";  
    t2.printShort();  
    cout << endl << " ";  
    t2.printLong();  
  
    cout << "\nhour si minute specificate; second implicit:"  
        << endl << " ";  
    t3.printShort();  
    cout << endl << " ";  
    t3.printLong();  
  
    cout << "\nhour, minute si second specificate:"  
        << endl << " ";  
    t4.printShort();  
    cout << endl << " ";  
    t4.printLong();  
  
    cout << "\nvalori invalide pentru hour, minute si second:"  
        << endl << " ";  
    t5.printShort();  
    cout << endl << " ";  
    t5.printLong();  
    cout << endl;  
  
    return 0;  
}
```

2. Constructorii clasei Time. Înlocuiți constructorul clasei `Time` cu 4 constructori care să aibă 0, 1, 2 și 3 parametri fără valori implicite. Funcția `main` trebuie să ruleze la fel ca înainte, fără nicio modificare.

3. Funcția tick în clasa Time. Adăugați funcția publică `tick` clasei `Time`. Această funcție incrementează data membră `second`. Asigurați-vă că tratați corect trecerea la un nou minut, la o nouă oră și la o nouă zi. Apelați funcția `tick` pentru obiectele `t3` și `t4` din `main`.

4. Clasa Date. Testați funcționarea clasei `Date`:

```
#include <iostream>
using std::cout;
using std::endl;
//Varianta simplificata a clasei Date
class Date
{
public:
    //constructor implicit
    Date(int = 1, int = 1, int = 1990);
    void print();
private:
    int day;
    int month;
    int year;
};
//constructor fara verificarea valorilor
Date::Date(int d, int m, int y)
{
    day = d;
    month = m;
    year = y;
}
//Tipareste data in forma zi-luna-an
void Date::print()
{
    cout << day << '-' << month << '-' << year;
}
int main()
{
    Date date1(5, 3, 2007), date2;

    cout << "date1 = ";
    date1.print();
    cout << endl << "date2 = ";
    date2.print();

    date2 = date1;//asignare prin copierea membru cu membru
    cout << endl << endl
        << "Dupa copierea membru cu membru, date2 = ";
    date2.print();
    cout << endl;

    return 0;
}
```

5. **Clasa `DateAndTime`.** Scrieți o nouă clasă care să reprezinte modelul unui nou tip de obiect care combină caracteristicile obiectelor de tip `Time` cu cele ale obiectelor de tip `Date`. Clasa `DateAndTime` va avea:

- 6 date membre: `year`, `month`, `day`, `hour`, `minute`, `second`.
- Un constructor cu 6 parametri cu valori implicite.
- O funcție membră de tipărire a celor 6 date membre într-un format ales de voi
- O funcție `tick` care va incrementa secunda. Această funcție va trebui să trateze și cazul 31.12.2012 23:59:59.