PCLP2

Laboratorul 3

1. **Structura `Time`.** Rulați programul de mai jos care implementează şi apoi foloseşte structura Time, încercând să înțelegeți fiecare instrucțiune:

```
#include <iostream>
using std::cout;
using std::endl;
struct Time //definitia structurii
{
    int hour; //0-23
    int minute; //0-59
    int second; //0-59
};
void printShort(const Time&);
void printLong (const Time&);
int main()
{
    Time dinnerTime; //obiect de tip Time
    //asignarea unor valori valide membrilor obiectului
    dinnerTime.hour = 18;
    dinnerTime.minute = 30;
    dinnerTime.second = 0;
    cout << "Cina va avea loc la ora ";
    printShort(dinnerTime);
    cout << " (format scurt), \nadica la ora ";
    printLong(dinnerTime);
    cout << " (format lung).\n" << endl;
    //asignarea unor valori invalide membrilor obiectului
    dinnerTime.hour = 29;
    dinnerTime.minute = 73;
    cout << "Obiect de tip Time cu valori invalide: ";
    printShort(dinnerTime);
    cout << endl;
    system("pause");
    return 0;
}
void printShort(const Time &t)
{
    cout << (t.hour < 10 ? "0" : "") << t.hour << ":"
    << (t.minute < 10 ? "0" : "") << t.minute;
}
void printLong(const Time &t)
{
    cout << ((t.hour == 0 || t.hour == 12) ? 12 : t.hour % 12)
    << ":" << (t.minute < 10 ? "0" : "") << t.minute
    << ":" << (t.second < 10 ? "0" : "") << t.second
    << (t.hour < 12 ? " AM" : " PM");
}
```

2. **Clasa `Time`**. Înlocuiți mai întâi cuvântul `struct` cu cuvântul `class` în programul de mai sus şi compilați. Remarcați eroarea compilatorului. În structuri membrii sunt publici, iar in clase sunt privați dacă nu se specifică altfel în definiția structurii sau a clasei. Clasele se conformează astfel principiului încapsulării datelor promovat de programarea orientată pe obiecte prin care membrii sunt privați în mod implicit. Rulați şi înțelegeți programul următor:

```
#include <iostream>
using std::cout;
using std::endl;
class Time
{
    public:
        Time(); //constructor
        void setTime(int, int, int); //asignarea 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
};
Time::Time()
{
    hour = minute = second = 0;
}
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");
}
int main()
{
    Time t; //instantiaza obiectul t de tip Time
    cout << "Valoarea initialia in format scurt este ";
    t.printShort();
    cout << "\nValoarea initialia in format lung este ";
    t.printLong();
    t.setTime(13, 27, 6);
    cout << "\n\nOra in format scurt dupa setTime este ";
    t.printShort();
    cout << "\nOra in format lung dupa setTime este ";
```

```
    t.printLong();
    //asignarea unor valori invalide membrilor obiectului
    t.setTime(99, 99, 99);
    cout << "\n\nDupa asignarea valorilor invalide:"
         << "\nOra in format scurt: ";
    t.printShort();
    cout << "\nOra in format lung: ";
    t.printLong();
    cout << endl;
    system("pause");
    return 0;
}
```

3. **Constructorul unei clase**. O clasă poate avea unul sau mai multe funcții constructor. Constructorul este apelat automat atunci când declaram un obiect al clasei. Identificați constructorul clasei `Time`. Adăugați în constructor următoarea instrucțiune de tipărire:

```
cout << "S-a rulat constructorul clasei Time" << endl;
```

şi rulați din nou programul. Remarcați în ce moment al derulării programului s-a apelat constructorul.

4. **Funcții setter şi getter**. Conform principiului încapsulării datelor, datele membre ale unei clase sunt private, iar accesul la ele se face prin funcții setter – cele prin care se incarcă o valoare într-o dată membră a obiectului, respectiv prin funcțiile getter – cele prin care se citeşte valoarea unei date membre a obiectului. Adăugați clasei `Time` funcțiile de mai jos, urmând modelul funcției `setTime`:

```
void setHour(int);
void setMinute(int);
void setSecond(int);
int getHour();
int getMinute();
int getSecond();
```

Funcțiile setter se caracterizează prin faptul că sunt de tip void, iar cele getter prin faptul că nu au parametri.

Modificați programul de la exercițiul 2 astfel:

- Înlocuiți apelul

  ```
  t.setTime(13, 27, 6);
  ```

  cu

  ```
  t.setHour(13);
  t.setMinute(27);
  t.setSecond(6);
  ```

- Apelați cele trei funcții getter într-o nouă instrucțiune de afişare adaugată funcției `main`.