# Laborator 2

Dovada de functionalitate la fiecare problema va fi in main fiecare problema fiind apelata din functii diferite. Main-ul va fi dupa toate probleme care au mentionate fisierele in care au fost scrise clasele.

### Problema 3

Să se scrie o aplicaţie în care se modelează clasa Student cu nume, prenume şi notele din sesiunea din iarnă. Să se afişeze numele studenţilor din grupă care au restanțe și apoi numele primilor 3 studenţi din grupă în ordinea mediilor

**first_set.hpp**

```
#ifndef Student_hpp
#define Student_hpp

#include <stdio.h>
#include <iostream>
#include <string>
using namespace std;


class Student {
    string nume;
    string prenume;
    string grupa;
    int note[4];
public: // membri publici
    Student();
    Student(string nume, string prenume, string grupa);
    void set_note(int note[]);
    string get_full_name();
    string get_nume();
    string get_grupa();
    string get_prenume();
    bool are_examene_restante();
    double get_medie();
};


#endif /* Student_hpp */
```

**first_set.cpp**

```
#include "first_set.hpp"

Student::Student() {}

Student::Student(string nume_input, string prenume_input, string grupa_input) {
    nume = nume_input;
    prenume = prenume_input;
```

```
    grupa = grupa_input;
}

void Student::set_note(int note_input[4]) {
    int i;
    for (i = 0; i < 4; i++) {
        note[i] = note_input[i];
    }
}
string Student::get_full_name() {
    return nume + " " + prenume;
}

string Student::get_nume() {
    return nume;
}
string Student::get_grupa() {
    return grupa;
}
string Student::get_prenume() {
    return prenume;
}

bool Student::are_examene_restante() {
    int i;
    for (i = 0; i < 4; i++) {
        if (note[i] < 4) {
            return true;
        }
    }
    return false;
}

double Student::get_medie() {
    return (double)(note[0] + note[1] + note[2] + note[3]) / 4;
}
```

### Problema 9

Implement the program presented in the third example and examine the compilation errors if are by eliminating the existing comments? Modify the program so the object obiect_derivat will be able to access the aduna( ) and scade( ) methods, from the main( ) function keeping the private inheritance.

**second_set.hpp**

```
#ifndef baza_deriv_hpp
#define baza_deriv_hpp

#include <stdio.h>
class Baza {
protected:
    int a, b;
```

```
public:
    Baza( ) { a = 1; b = 1; }
    void setA(int a) {
        this->a = a;
    }
    void setB(int b) {
        this->b = b;
    }
    int getA( ) {
        return a;
    }
    int getB( ) {
        return b;
    }
    int aduna( ) {
        return a + b;
    }
    int scade( ) {
        return a - b;
    }
};

class Derivata : private Baza {
public:
    int inmulteste() {
        return a * b;
    }
    int aduna( ) {
        return a + b;
    }
    int scade( ) {
        return a - b;
    }
};

#endif /* baza_deriv_hpp */
```

### Problema 18

La exemplul al treilea extindeţi clasa de bază cu alte metode virtuale, redefinite în clasele derivate, cum ar fi metode get( ) şi set( ) pentru greutatea vehiculului (variabila greutate).

**third_set.hpp**

```
#ifndef third_set_hpp
#define third_set_hpp

#include <stdio.h>
#include <iostream>
using namespace std;

class Vehicul {
```

```
    int roti;
    float greutate;
public:
    virtual void mesaj( ) {
        cout << "Mesaj din clasa Vehicul\n";
    }
    virtual void set_greutate(float greutate) {
        this->greutate = greutate;
    }
    virtual float get_greutate() {
        return this->greutate;
    }
};

class Automobil : public Vehicul {
    int incarcatura_pasageri;
public:
    void mesaj( ) override {
        cout << "Mesaj din clasa Automobil\n";
    }
    void set_incarcatura_pasageri(int incarcatura_pasageri) {
        this->incarcatura_pasageri = incarcatura_pasageri;
    }
    int get_incarcatura_pasageri() {
        return this->incarcatura_pasageri;
    }
    void set_greutate(float greutate) override {
        this->Vehicul::set_greutate(greutate + incarcatura_pasageri);
    }
    float get_greutate() override{
        return this->Vehicul::get_greutate();
    }
};

class Camion : public Vehicul {
    int incarcatura_pasageri;
    float incarcatura_utila;
public:
    int pasageri( ) {
        return incarcatura_pasageri;
    }
    void set_greutate(float greutate) override {
        this->Vehicul::set_greutate(greutate + incarcatura_pasageri +
incarcatura_utila);
    }
    float get_greutate() override{
        return this->Vehicul::get_greutate();
    }
};

class Barca : public Vehicul {
    int incarcatura_pasageri;
```

```
public:
    int pasegeri( ){
        return incarcatura_pasageri;
    }
    void mesaj( ) override {
        cout << "Mesaj din clasa Barca\n";
    }
    void set_greutate(float greutate) override {
        this->Vehicul::set_greutate(greutate + incarcatura_pasageri);
    }
    float get_greutate() override{
        return this->Vehicul::get_greutate();
    }
};

#endif /* third_set_hpp */
```

### main function

Here all the above classes are shown.

**main.cpp**

```
#include <iostream>
using namespace std;
#include "first_set.hpp"
#include "second_set.hpp"
#include "third_set.hpp"


void run_first() {

    cout << "Din primul set de probleme problema 3" << "\n";

    Student gal_oscar = Student("Gal", "Oscar", "1");
    int note[4] = {10, 10, 10, 10};
    gal_oscar.set_note(note);
    Student badau_florica = Student("Badau", "Florica", "1");
    note[3] = 7;
    badau_florica.set_note(note);
    Student marian_alexandru = Student("Marian", "Alexandru", "1");
    note[0] = 3;
    note[2] = 5;
    marian_alexandru.set_note(note);
    Student florin_dumitrescu = Student("Florin", "Dumitrescu", "1");
    florin_dumitrescu.set_note(note);
    Student florin_scarlatescu = Student("Florin", "Scarlatescu", "1");
    note[0] = 10;
    note[2] = 6;
    florin_scarlatescu.set_note(note);
    Student geanina_alexandrescu = Student("Geanina", "Alexandrascu", "1");
    note[3] = 7;
    geanina_alexandrescu.set_note(note);
```

```
    Student students[] = {
        gal_oscar, badau_florica, marian_alexandru, florin_dumitrescu,
florin_scarlatescu, geanina_alexandrescu
    };
    cout << "Studentii cu restante sunt: \n";
    int i;
    for (i = 0; i < 6; i++) {
        if (students[i].are_examene_restante()) {
            cout << students[i].get_full_name() << '\n';
        }
    }

    cout << "Studentii in ordinea mediilor sunt: \n";
    Student primul = students[0];
    Student aldoilea;
    Student altreilea;
    for (i = 1; i < 6; i++) {
        if (students[i].get_medie() > primul.get_medie()) {
            altreilea = aldoilea;
            aldoilea = primul;
            primul = students[i];
        } else if (students[i].get_medie() > aldoilea.get_medie()) {
            altreilea = aldoilea;
            aldoilea = students[i];
        } else if (students[i].get_medie() > altreilea.get_medie()) {
            altreilea = students[i];
        }
    }
    cout << "1. " << primul.get_full_name() << " - media: " << primul.get_medie() <<
"\n";
    cout << "2. " << aldoilea.get_full_name() << " - media: " << aldoilea.get_medie()
<< "\n";
    cout << "3. " << altreilea.get_full_name() << " - media: " <<
altreilea.get_medie() << "\n";

}

void run_second() {

    cout << "\nDin al doilea set de probleme problema 9" << "\n";
    Baza obiect_baza;
    cout << "\nAfis din baza (val. initiale): " << obiect_baza.getA( ) << " " <<
obiect_baza.getB( ) << '\n';
    cout << "\nSuma este (cu val. initiale, baza) = " << obiect_baza.aduna( ); //
corect aduna( )
    cout << "\nDiferenta este (cu val. initiale, baza) = " << obiect_baza.scade( );
    obiect_baza.setA(2);
    obiect_baza.setB(3);
    cout << "\nAfis din baza (modificat): " << obiect_baza.getA( ) << " " <<
obiect_baza.getB( ) << '\n';
    cout << "\nSuma/Diferenta dupa setare= " <<obiect_baza.aduna( ) << "/"<<
```

```
obiect_baza.scade( )<<'\n';
    Derivata obiect_derivat;
    cout << "\nProdusul este (din derivat cu val. initiale) = " <<
obiect_derivat.inmulteste( ) << '\n';
    cout << "\nSuma este (din derivat cu val. initiale, baza) = " <<
obiect_derivat.aduna( ) << '\n';
    cout << "Diferenta dupa setare= " << obiect_derivat.scade( ) << '\n';

}

void run_third() {
    cout << "\nDin al treilea set de probleme problema 9" << "\n";

    Vehicul monocicleta;
    Automobil ford;
    Camion semi;
    Barca barca_de_pescuit;
    monocicleta.mesaj( );
    monocicleta.set_greutate(123.12);
    cout << monocicleta.get_greutate() << "\n";
    ford.set_incarcatura_pasageri(12);
    ford.set_greutate(30);
    cout << ford.get_greutate() << "\n";
    ford.mesaj( );
    semi.mesaj( );
    barca_de_pescuit.mesaj( );
    cout << monocicleta.get_greutate();
    cout << ford.get_greutate();
    Vehicul *pmonocicleta;
    Automobil *pford;
    Camion *psemi;
    Barca *pbarca_de_pescuit;
    cout << "\n";
    pmonocicleta = &monocicleta;
    pmonocicleta->mesaj( );
    pford = &ford;
    pford->mesaj( );
    psemi = &semi;
    psemi->mesaj( );//din CB
    pbarca_de_pescuit = &barca_de_pescuit;
    pbarca_de_pescuit->mesaj( );
    cout << "\n";
    pmonocicleta = &monocicleta;
    pmonocicleta->mesaj( );
    pmonocicleta = &ford;
    pmonocicleta->mesaj( );
    pmonocicleta = &semi;
    pmonocicleta->mesaj( );
    pmonocicleta = &barca_de_pescuit;
    pmonocicleta->mesaj( );
}
```

```
int main(int argc, const char * argv[]) {
    run_first();
    run_second();
    run_third();
}
```