Peter Bucur Mihai-Viorin

A3.
```
class Person_Type : public Person{
  public:
    using Person::Person;
    int getAge() const;
      return age;
    }
    char * getName const{
      return name;
    }

};
operator
ostream & operator<<(ostream & stream, const Person_Type & ps){
  stream<<ps.getName()<<", "<<ps.getAge();
  return stream;
}
Person_Type operator+(const Person_Type & p, int n){
  int a=p.getAge()+n;
  cout<<p.getName()<<", "<<a<<endl;
}
Person_Type operator+(int n, const Person_Type & p){
  int a = p.getAge() + 2*n;
  cout<<p.getName()<<", "<<a<<endl;
}
```

Detcalescu Mihai - Liviu

F3. a) B.m()

B.m()

b) B.m()

B.set() : SETARE...

Trebuie facuta ...

B.m()

D.set().

E3. 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2

C3. 1 0 0 0 1 2 1 1

B3. a) 8

b) 8

Petrulescu Mihai-Liviu

63. a) Rezultat:

1

2.

Modelul este o clasă cu o variabilă x (default 0) pe care o incrementează prin funcția incr() și returnează noua valoare.

Componenta View deschide 2 fișiere (unul de intrare și unul de ieșire) și prin funcția get_command() returnează conținutul din input (prima linie). Afișarea se realizează în funcția write_result();

Controllerul verifică prin metoda process() dacă comanda corespunde cerinței (= "add") apoi provoacă răspunderea către instanța clasei model (executarea incr()) și înapoi către View pt. afișare.

În main se creează instanțele celor 3 și se apelează de 2 ori controllerul (.process()) astfel afișându-se rezultatul.

b) class View2 : public AbstractView {

public

View2(){

out.open("output.txt", ios::out);

}

string get_command()

string rez;

getline(std::cin, rez);

return rez;

}

// ... identice ca în View

}

Tetelbaum Mihai-Silviu

H_3 - Rezultat:

100

200.

Clasa C conține o valoare privată x, specifică instanței, 2 funcții de afișare și o funcție în care creează o nouă variabilă locală, denumită tot x pe care o inițializează cu 100.

În main creăm o instanță a lui C cu val. internă de 200 apoi etc. fct m. Această apelează display(x) cu val atribuită variabilei loc din ~~bb~~ blocul de instrucțiuni aferent (1 deci se afișează 100), apoi display_x() ce afișează 200 (noua var x nu are valoare de vedere în afara bloc. de instr. și deci e apelează val instanței

Tatulescu Mihai-Silviu

I3.
```
class NumePrenume implements Display{
    public void display(Person p){
        System.out.println(p.get_name() + " " + p.get_prenume());
    }
}

class PrenumeNume implements Display{
    public void display(Person p){
        System.out.println(p.get_prenume() + " " + p.get_name());
    }
}
```

Tetcuborca Mihai-Silviu

73.

a) Deoarece p și q sunt instanțe ale aceleiași clase Person, linia p=q are ca efect alocarea pointerului p către zona de memorie specifică variabilei/instanței q. Astfel după această linie amândouă variabile pointează către aceeași instanță, deci vor avea aceeași valoare. (30).

b) class Person {

public:

//...

int getAge const { return *age; }

Person & operator = (const Person& p) {

int i = p.getAge();

*age = i;

return *this;

}

//...

}

Se va afișa:

20

0

0

30

// Acum pe linia p=q, p salvează valoarea atributului lui q și nu adresa pointerului.