

# S07 – Supraincercari de operatori

Alin Zamfiroiu

[alin.zamfiroiu@csie.ase.ro](mailto:alin.zamfiroiu@csie.ase.ro)

# Operator <<

- Operatorul << sau operatorul ostream este operatorul folosit pentru afisarea unui obiect.
- Exemplu de apel:
  - Student s1;
  - cout<<s1;//afiseaza toate attributele obiectului s1;
- Cum sau unde trebuie implementat?
- Cum se asigura posibilitatea apelarii operatorului in cascada: cout<<s1<<s2<<s3?

# Operator<<

- Pentru clasa Student sa se implementeze operatorul << care sa afiseze la consola attributele obiectului pentru care se apeleaza operatorul. Attributele sunt afisate cate unul pe linie.

# Operator>>

- Operator>> sau operator istream este folosit pentru a citi de la tastatura un obiect.
- Exemplu de apel:
  - Student s1;
  - cin>>s1;
- Atentia la memory-leak-uri!!!!
- Cum se asigura posibilitatea apelarii operatorului in cascada cin>>s1>>s2>>s3;

# Operator>>

- Pentru clasa Student sa se implementeze operator>> care sa citeasca datele despre un obiect de tip Student de la tastatura.

# Operator+=

- Sa se imlementeze pentru clasa Student operatorul += pentru a aduna un student unui alt student. Operatorul va adauga toate notele celui de al doilea student primului. Exemplu de apel:
  - Student s1;
  - Student s2;
  - s1+=s2;

# Operator+=

- Sa se implementeze operatorul += pentru a adauga o nota unui student.
- Exemplu de apel:
  - Student s1;
  - s1+=10;//se adauga nota 10 studentului s1;
- Asemănător pentru operator -= și \* =;

# Operator ++

- ++a sau a++ → preincrementare / postincrementare
- Sa se implementeze pentru clasa Student atat operatoul ++ preincrementare cat si operatoul ++ postincrementare pentru marirea tuturor notelor cu o unitate. Diferenta dintre cei doi se face prin intermediul unui parametru de tip int, pentru operatorul ++ postincrementare.
- Mare atentie la operatorul ++ postincrementare, deoarece trebuie sa returneze valoarea nemodificata, insa pointerul this se modifica.

# Operator <, >, <=, >=, ==, !=

- Operatorii de comparatie sunt folositi pentru a compara obiectele create. Modul de comparare il decide programatorul in momentul implementarii acestor operatori.
- Operatorul returneaza o valoarea booleana.
- Sa se implementeze operatorii de comparatie pentru clasa Student, comparatia realizanduse dupa media fiecarui student. Media se calculeaza pe baza notelor studentilor.

# Operator de cast

- Operator de cast sau operator de transformare este folosit atunci cand vrem sa convertim un obiect de un anumit tip la un alt tip. Exemplu:
  - float a=5.4;
  - int b=a;//ce valoare are b?
- Sa se implementeze operatorul de cast la float pentru clasa Student, care sa returneze media acestuia. Exemplu de apel:
  - Student s1;
  - float medie=s1;
- Asemănător se face operatorul de cast la int, double sau orice alt tip dorit.

# Operator [] (index)

- Operatorul index este util atunci cand in clasa avem un atribut vector si dorim obtinerea unui anumit element din cadrul vectorului. Pozitia elementului dorit se da ca parametru operatorului. Exemplu de apel:
  - Student s1;
  - cout<<s1[0]<<s1[1];
- Sa se implementeze operatorul [] pentru clasa Student, pentru vectorul de note.
- Cum putem sa facem sa si modificam valorile din cadrul vectorului de note?

# Tema

- Sa se realizeze o clasa diferita fata de cea de la tema precedenta.
- Clasa sa aiba cel putin un atribut alocat dinamic si un vector
- Sa se realizeze operatorii de lucru cu consola pentru aceasta clasa(<< si >>).
- Sa se implementeze operatorul de cast la int si operatorul de cast la float.
- Sa se realizeze cel putin doi operatori de comparare pentru clasa realizata;
- Sa se implementeze operatorul – (pre si post decrementare)
- Sa se implementeze operatorul [] pentru vectorul atribut al clasei.
- Atributele clasei sunt private si se realizeaza get-eri si set-eri pentru ele.
- Clasa trebuie sa aiba constructori, destructor si operator=.
- Toate metodele implementate sunt testate in functia main().