Clase – Supraîncărcare Operatori

[Observații: 1](#_Toc467016824)

[Observații suplimentare 1](#_Toc467016825)

[Operatori Input/Output << >> 1](#_Toc467016826)

[Metode friend 1](#_Toc467016827)

[Metode de acces (recomandat) 2](#_Toc467016828)

[Operator atribuire (Operator=) 4](#_Toc467016829)

[Operatori unari 5](#_Toc467016830)

[Operatori binari 5](#_Toc467016831)

[Aritmetici 5](#_Toc467016832)

[Comparație 5](#_Toc467016833)

[Operator() 5](#_Toc467016834)

[Operator [] 6](#_Toc467016835)

[Bibliografie suplimentară 6](#_Toc467016836)

# Observații:

* Documentul este gândit pentru studenții care au participat la seminar. Este posibil sa întâmpinați dificultăți in parcurgerea / înțelegerea lui daca nu ați fost prezenți.
* Eventualele dezacorduri sau greșeli de scriere au fost făcute intenționat, cu rolul de a vă binedispune în timpul citirii acestui document. ☺

# Observații suplimentare

* Parcurgeți capitolul de supraîncărcare a operatorilor din manual;
* Implementați exemplele de la finalul capitolului (în special clasele Complex, String și Fracție).

# Operatori Input/Output << >>

## Metode friend

|  |  |
| --- | --- |
|  | Execrițiul se bazează pe codul din exemplul <http://online.ase.ro> – “StudentClass - Constructor de copiere” |

1. Dorim să putem face preluarea și afișarea datelor din/în consolă cu ajutorul operatorilor << >>, ca în exemplul de mai jos.

|  |
| --- |
| int marks**[]** **=** **{** 10**,**9 **};**  Student s**(**"Nume"**,** 21**,** 2**,** marks**);**  cout **<<** s**;**  cin **>>** s**;**  cout **<<** s**;** |

1. Supraîncărcarea operatorilor << și >> poate fi implementată ca mai jos. Metodele se vor adăuga în afara clasei Student.

|  |
| --- |
| ostream **&** **operator<<(**ostream **&** consola**,** const Student **&** st**)**  **{**  consola **<<** endl **<<** "Nume: "**;**  **if(**st**.**nume **!=** **NULL)**  consola**<<** st**.**nume**;**  consola **<<** endl **<<** "Varsta: " **<<** st**.**varsta **;**  consola **<<** endl **<<** "Note: "**;**  **for** **(**int i **=** 0**;** i **<** st**.**nrNote**;** i**++)**  consola **<<** " " **<<** st**.**note**[**i**];**  **return** consola**;**  **}**  istream **&** **operator** **>>** **(**istream **&** consola**,** Student **&** st**)**  **{**  #pragma region Nume  cout **<<** endl **<<** "Nume: "**;**  char buffer**[**200**];**  consola **>>** buffer**;**    **if** **(**st**.**nume **!=** **NULL)**  **delete[]** st**.**nume**;**  st**.**nume **=** **new** char**[**strlen**(**buffer**)** **+** 1**];**  strcpy**(**st**.**nume**,** buffer**);**  #pragma endregion  #pragma region Varsta  cout **<<** "Varsta: "**;**  cin **>>** st**.**varsta**;**  #pragma endregion  #pragma region Note  cout **<<** "NrNote: "**;**  cin **>>** st**.**nrNote**;**  **if** **(**st**.**note **!=** **NULL)**  **delete** **[]** st**.**note**;**  st**.**note **=** **new** int**[**st**.**nrNote**];**  **for** **(**int i **=** 0**;** i **<** st**.**nrNote**;** i**++)**  **{**  cout **<<** "nota[" **<<** i **<<** "]: "**;**  cin **>>** st**.**note**[**i**];**  **}**  #pragma endregion  **return** consola**;**  **}** |

1. Codul implementat nu se va compila deoarece metodele implementate accesează membrii privați din clasa Student. Soluția constă în a declara metodele implementate ca metode “friend” în clasa Student (ca mai jos) sau de a utiliza metodele de acces (ca în subcapitolul următor).

|  |
| --- |
| friend ostream**&** **operator<<(**ostream**&** consola**,** const Student **&**st**);**  friend istream**&** **operator** **>>** **(**istream**&** consola**,** Student **&**st**);** |

## Metode de acces (recomandat)

|  |  |
| --- | --- |
|  | Execrițiul se bazează pe codul din exemplul <http://online.ase.ro> – “StudentClass - Constructor de copiere” |

1. Supraîncărcarea operatorilor << și >> poate fi implementată utilizând metodele de acces din clasa Student, nemaifind astfel necesară declararea metodelor implementate ca metode “friend” în clasa Student.

|  |
| --- |
| ostream **&** **operator<<(**ostream **&** consola**,** const Student **&** st**)**  **{**  consola **<<** endl **<<** "Nume: "**;**  **if** **(**st**.**getNume**()** **!=** **NULL)**  consola **<<** st**.**getNume**();**  consola **<<** endl **<<** "Varsta: " **<<** st**.**getVarsta**();**  consola **<<** endl **<<** "Note: "**;**  **for** **(**int i **=** 0**;** i **<** st**.**getNrNote**();** i**++)**  consola **<<** " " **<<** st**.**getNote**()[**i**];**  **return** consola**;**  **}**  istream **&** **operator** **>>** **(**istream **&** consola**,** Student **&** st**)**  **{**  #pragma region Nume  cout **<<** endl **<<** "Nume: "**;**  char buffer**[**200**];**  consola **>>** buffer**;**  /\*if (st.nume != NULL)  delete[] st.nume;  st.nume = new char[strlen(buffer) + 1];  strcpy(st.nume, buffer);\*/  st**.**setNume**(**buffer**);**  #pragma endregion  #pragma region Varsta  cout **<<** "Varsta: "**;**  //cin >> st.varsta;  int varsta**;**  cin **>>** varsta**;**  st**.**setVarsta**(**varsta**);**  #pragma endregion  #pragma region Note  cout **<<** "NrNote: "**;**  /\*  cin >> st.nrNote;  if (st.note != NULL)  delete[] st.note;  st.note = new int[st.nrNote];  for (int i = 0; i < st.nrNote; i++)  {  cout << "nota[" << i << "]: ";  cin >> st.note[i];  }\*/    int nrNote**;**  cin **>>** nrNote**;**  int **\***note **=** **new** int**[**nrNote**];**  **for** **(**int i **=** 0**;** i **<** nrNote**;** i**++)**  **{**  cout **<<** "nota[" **<<** i **<<** "]: "**;**  cin **>>** note**[**i**];**  **}**  st**.**setNote**(**nrNote**,** note**);**  #pragma endregion  **return** consola**;**  **}** |

# Operator atribuire (Operator=)

1. Modificați codul din metoda main ca mai jos.

|  |
| --- |
| **void** main**()**  **{**  **int** marks**[]** **=** **{** **10,9** **};**  Student s**(**"Nume"**,** **21,** **2,** marks**);**  *//Constructor de copiere*  *//OMetodaSimpla(s); //apeleaza constructorul de copiere*  *//Student s2 = s; //apeleaza constructorul de copiere*  Student s2**(**"Nume2"**,** **22,** **2,** marks**);**  *//Operator=*  s2 **=** s**;** *//apeleaza operator=*  *//s2.operator=(s); //echivalent cu s2 = s;*  **}** |

1. Rulați programul și observați eroarea afișată. Încercați să determinați linia de cod la care se produce eroarea.
2. Adăugați în clasa Student implementarea pentru operatorul egal

|  |
| --- |
| Student **&** **operator=(const** Student **&** source**)**  **{**  cout **<<** "Operator =" **<<** endl**;**  *//1.*  **this->**varsta **=** source**.**varsta**;**  **this->**nrNote **=** source**.**nrNote**;**  *//2. copiere atribute \**    *//Nume*  *//a) stergere spatiu existent*  **if** **(this->**nume **!=** NULL**)**  **delete[]** **this->**nume**;**  **if** **(**source**.**nume **==** NULL**)**  **this->**nume **=** NULL**;**  **else**  **{**  *//b) alocare spatiu*  **this->**nume **=** **new** **char[**strlen**(**source**.**nume**)+1];**  *//c) copiere*  strcpy**(this->**nume**,** source**.**nume**);**  **}**  *//Note*  *//a) stergere spatiu existent*  **if** **(this->**note **!=** NULL**)**  **delete[]** **this->**note**;**  *//b) alocare spatiu*  **this->**note **=** **new** **int[this->**nrNote**];**  *//c) copiere*  **for** **(int** i **=** **0;** i **<** **this->**nrNote**;** i**++)**  **this->**note**[**i**]** **=** source**.**note**[**i**];**  **}** |

# Operatori unari

Exemple: ! ([logical NOT](https://msdn.microsoft.com/en-us/library/1k6w8551.aspx)), ++ ([increment](https://msdn.microsoft.com/en-us/library/dy3d35h8.aspx)), -- ([decrement](https://msdn.microsoft.com/en-us/library/dy3d35h8.aspx)), operator cast

|  |  |
| --- | --- |
|  | Lista completă de operatori unari: <https://msdn.microsoft.com/en-us/library/f672kxz8.aspx> |

|  |  |
| --- | --- |
|  | Implementarea operatorilor pentru clasa Student este disponibila pe <http://online.ase.ro> – “StudentClass - Operatori unari” |

# Operatori binari

## Aritmetici

Exemple: +, -, \*, /, -=, +=

|  |  |
| --- | --- |
|  | Lista completă de operatori binari: <https://msdn.microsoft.com/en-us/library/czs2584d.aspx> |

|  |  |
| --- | --- |
|  | Implementarea operatorilor pentru clasa Student este disponibila pe <http://online.ase.ro> – “StudentClass - Operatori binari aritmetici” |

## Comparație

Exemple: == , >, <. >= , =<

|  |  |
| --- | --- |
|  | Lista completă de operatori binari: <https://msdn.microsoft.com/en-us/library/czs2584d.aspx> |

|  |  |
| --- | --- |
|  | Implementarea operatorilor pentru clasa Student este disponibila pe <http://online.ase.ro> – “StudentClass - Operatori binari comparatie” |

# Operator()

|  |  |
| --- | --- |
|  | Implementarea operatorilor pentru clasa Student este disponibila pe <http://online.ase.ro> – “StudentClass - Operator()” |

# Operator []

|  |  |
| --- | --- |
|  | Implementarea operatorilor pentru clasa Student este disponibila pe <http://online.ase.ro> – “StudentClass – Operator[]” |

# Bibliografie suplimentară

* <http://stackoverflow.com/questions/4421706/operator-overloading/4421708#4421708>
* <http://en.cppreference.com/w/cpp/language/operators>