

Curs 5 POO

© Conf. univ dr. Crenguta M. Puchianu

- ❑ Tablouri uni- si bidimensionale
- ❑ Clasa String
- ❑ Formatarea datelor
- ❑ Clasa StringBuffer

Tablouri

- ❑ Un **tablou** este o colecție liniară de elemente în care fiecare element este accesibil prin intermediul unui indice ce reprezintă poziția elementului în tablou. În Java, valoarea indicelui pornește de la 0, unde se găsește primul element.
- ❑ Un tablou furnizează spațiu pentru valorile de tip primitiv sau referință la obiectele de un același tip. Aceste valori se numesc *elementele* tabloului.
- ❑ Elementele unui tablou sunt ordonate după unul sau mai multe criterii numite *dimensiuni*. O dimensiune a unui tablou este indicată printr-o pereche de []. După numărul de dimensiuni, un tablou este uni- sau multidimensional.
- ❑ Referința la un tablou este memorată într-o *variabilă tablou*.
- ❑ Sintaxa declarării unei variabile tablou:

[modificatorAcces] [static] **TipDeDate[] id; sau TipDeDate id[];**

Tablouri (cont.)

- ❑ Crearea unui tablou:

`id = new TipDate[expresie];`

unde *Tip* este tipul de date al variabilei tablou cu identificatorul *id*, iar *expresie* este fie un literal întreg, fie o expresie cu valoare un număr întreg.

- ❑ Dupa creare, toate elementele tabloului numere sunt inițializate cu 0, iar cele ale tabloului personal cu valoarea null.
- ❑ Umplerea cu valori:

**`for (int i=0; i<5; i++) numere[i]=2*i;
for (int i=0; i<1000;i++) personal[i]=new Persoana();`**

Tablouri (cont.)

- ❑ Initializarea unui tablou cu lista de initializatori:

TipDeDate[] id={ListaDeInitializatori}

unde ListaInitializatori este o multime de elemente de același tip cu tipul tabloului, ce sunt valorile elementelor tabloului id.

- ❑ Parcurgerea elementelor unui tablou:

for(int i=0; i<numeVariabilaTablou.length; i++) instrucțiune;

Sau

for(*declaratie: expresie*) instructiune;

- ❑ Clasa System are o metoda folosita pentru copierea elementelor unui tablou intr-un alt tablou:

**static void arraycopy (Object arraySursă, int pozSursă,
Object arrayDest, int pozDest, int nrElemente)**

Exemplu

Să scriem un program care conține o metodă ce elimină recursiv câte un element dintr-un tablou primit ca parametru actual.

În corpul metodei generăm un număr aleator între 0 și lungimea tabloului exclusiv, care reprezintă poziția elementului ce va fi șters din tablou.

După fiecare eliminare, creăm un nou tablou din tabloul inițial și îl afișăm.

Aceste operații se repetă până când tabloul conține un singur element.

Metoda `main(args...)` a programului creează un tablou ale cărui elemente sunt generate aleator, îl afișează și apelează metoda recursivă `eliminaElemente()`.

Tablouri multidimensionale

- ❑ Tablourile multidimensionale sunt declarate prin adăugarea dimensiunilor.

```
int[][] m;
```

- ❑ Alocarea memoriei necesare memorării matricii m se face astfel:

```
m=new int[3][3];
```

- ❑ Putem declara și construi tablouri multidimensionale de dimensiuni diferite.

```
int[][] b=new int[2][];  
b[0]=new int[3];  
b[1]=new int[4];
```

- ❑ În cazul tablourilor multidimensionale variabila length are valori diferite.

- ❑ **Obs. b[0][3]**
-

Exemplu

Un program care conține o metodă primește ca parametru al metodei un tablou de valori întregi și returnează un tablou bidimensional, în care prima coloană conține elementele

tabloului initial, iar a doua coloană conține numărul de apariții ale fiecărui element din tabloul initial. Programul afișează elementele sortate ale tabloului initial și elementele tabloului bidimensional. De exemplu:

Input: 1 3 1 5 7 8 5 2

Output:

1 1 2 3 5 5 7 8

1 2

2 1

3 1

5 2

7 1

8 1

Siruri de caractere. Clasa String

Prototipul constructorului	Semnificație
String()	Se creează un obiect String ce conține șirul vid.
String(String s)	Se creează un obiect String ce conține șirul s.
String(char[] arrayChar)	Obiectul creat conține o copie a caracterelor din tabloul arrayChar.
String(char[] arrayChar, int poz, int nr)	Al doilea argument al constructorului indică poziția de la care se copiază elementele tabloului arrayChar, iar al treilea argument specifică numărul de caractere ce vor fi copiate din tablou. Dacă unul din argumente depășește elementele tabloului se lansează excepția StringIndexOutOfBoundsException.
String(byte[] arrayBytes)	Obiectul creat conține o copie a caracterelor ale căror cod Unicode este specificat în tabloul arrayBytes.
String(byte[] arrayBytes, int poz, int nr, String schemaCodificare)	Parametrul schemaCodificare ia una din următoarele valori de tip String: ISO-8859-1, US-ASCII, UTF-8, UTF-16BE, UTF-16LE, UTF-16. Dacă schema de codificare nu este suportată de sistemul de operare se lansează o excepție de tip UnsupportedEncodingException.
String(StringBuffer sb)	Obiectul creat conține șirul din buffer-ul sb.

Metode instanta ale clasei String

Prototipul metoda	Semnificație
char charAt(int index)	Întoarce caracterul de pe pozitia index din șirul pe care este aplicată metoda.
int compareTo(String altSir)	Compară lexicografic două șiruri și returnează una din valorile: un număr negativ, 0 sau un număr pozitiv.
boolean equals(Object altObiect)	Compară (case sensitiv) șirul cu obiectul specificat. Întoarce true numai dacă altObiect este de tip String și conține același șir de caractere cu șirul cu care se compară.
boolean equalsIgnoreCase(String altSir)	Același efect ca mai sus, numai că nu ține cont de tipul literelor: mari sau mici.
int indexOf(int car)	Returnează poziția în șir pe care se găsește prima apariție a caracterului car specificat prin codul Unicode. Dacă nu-l găsește, valoarea rezultată este -1. Această metodă poate fi folosită pentru căutarea unui subșir (dat ca parametru) într-un șir.
int lastIndexOf(String str)	Întoarce poziția ultimei apariții a subșirului str în șir. Dacă nu-l găsește, valoarea returnată este -1.
int length()	Returnează lungimea șirului pe care este aplicată metoda.

Metode instantă ale clasei String (cont.)

Prototipul metoda	Semnificație
String substring(int indiceStart)	Întoarce un nou șir care este un subșir extras din șirul inițial, de la poziția indiceStart și până la sfârșit.
String substring(int indiceStart, int indiceStop)	Întoarce un nou șir care este un subșir extras din șirul inițial, de la poziția indiceStart și până la indiceStop, exclusiv.
String[] split(String er)	Împarte șirul s în subșiruri după unul sau mai mulți delimitatori specificați în al doilea parametru și returnează un tablou ale cărui elemente memorează în ordine subșirurile.
boolean regionMatches(int pozInc1, String s2, int pozInc2, int nrCar)	Compară subșirurile de lungime nrCar, începând cu pozițiile specificate în primul și al treilea parametru. Metoda întoarce true dacă subșirurile sunt egale.
void getChars(int inc, int sf, char[] dest, int incDest)	Se copiază sf-inc-1 caractere din șir în tabloul dest, începând cu poziția incDest.
String toLowerCase()	Converteste toate majusculele la litere mici.
String toUpperCase()	Converteste toate literele mici la litere mari.
String trim()	Șterge spațiile din ambele capete ale șirului.

Formatarea datelor

❑ Sintaxa:

%[numarArgument\$][flag][latime][.precizie]conversie

%[numarArgument\$][flag][latime]conversie

%[flag][latime]conversie

Câmp al unui format	Semnificație	Valori
numarArgument	Specifică numărul de ordine al parametrului căruia se aplică formatul.	orice număr întreg strict pozitiv
flag	Un set de caractere care se aplică formatul.	- sau + sau (sau o sau ,
latime	Specifică numărul de caractere ale parametrului care vor fi luate în considerare.	orice număr întreg pozitiv. Dacă numărul este mai mare decât lungimea parametrului, se completează cu spațiu.
precizie	Se aplică numai valorilor numerice întregi și reale. În cazul numerelor întregi, semnifică numărul maxim de cifre luate în considerare. În cazul numerelor reale, specifică numărul de cifre zecimale ale numărului format.	
Conversie	Specifică conținutul furnizat în urma formatării.	un caracter. Caracterele folosite sunt în tabelul următor

Formatarea datelor (cont.)

Conversie	Categoria argumentului	Descriere
'b', 'B'	general	Dacă parametrul căruia se aplică formatul este de tip boolean sau Boolean, rezultatul este "true" sau "false", în funcție de valoarea parametrului. Altfel, rezultatul este "true".
'h', 'H'	general	Dacă parametrul este null, atunci rezultatul este "null". Altfel, rezultatul este obținut prin apelarea metodei Integer.toHexString(arg.hashCode()).
's', 'S'	general	Dacă parametrul este null, atunci rezultatul este "null". Dacă parametrul implementează Formattable, atunci se apelează metoda parametru.formatTo(). Altfel, rezultatul este obținut prin apelarea metodei parametru.toString().
'c', 'C'	caracter	Rezultatul este un character Unicode.
'd'	număr întreg	Rezultatul este formatat ca un număr întreg în baza 10.
'o'	număr întreg	Rezultatul este formatat ca un număr întreg în baza 8.
'x', 'X'	număr întreg	Rezultatul este formatat ca un număr întreg în baza 16.
'f'	număr real	Rezultatul este formatat ca un număr real în baza 10.
'g', 'G'	număr real	Rezultatul este formatat ca un număr real în baza 10 folosind notația științifică.

Metode instantia ale clasei StringBuffer

Prototipul metoda	Semnificație
char charAt(int index)	Întoarce caracterul de pe pozitia index din șirul pe care este aplicată metoda.
void setCharAt(int poz, char c)	Inserează caracterul c pe poziția poz a șirului curent.
StringBuffer insert(int poz, char[] c)	Inserează caracterele din tabloul c pe poziția poz a șirului memorat în obiectul StringBuffer.
StringBuffer insert(int poz, <i>tip</i> i)	Inserează valoarea lui i pe poziția poz a șirului memorat în obiectul StringBuffer, unde <i>tip</i> ={boolean, int, char, long, float, double, String}.
StringBuffer insert(int poz, Object o)	Inserează șirul o.toString() pe poziția poz a șirului memorat în obiectul StringBuffer.
StringBuffer append(<i>tip</i> i)	Are același efect cu apelul metodei insert(s.length()-1, i), adică adaugă prin concatenare la sfârșitul șirului valoarea lui i, unde <i>tip</i> ={boolean, int, char, long, float, double, String}
String reverse()	Înlocuiește șirul curent cu șirul în oglindă.
String toString()	Întoarce reprezentarea String a șirului conținut în

Exemplu

Să scriem un program care conține o metodă primește un sir ca parametru al metodei și îl modifică, înlocuind fiecare literă mică cu litera mare corespunzătoare și invers.
