



Laborator 8 - Java

Siruri de caractere (clasa String)

1. Siruri de caractere (clasa String)

Extragerea unui caracter dintr-un sir de caractere

Metoda charAt este folosita pentru a extrage un caracter aflat la o anumita pozitie intr-un sir de caractere. Precizam ca, primul caracter dintr-un sir se afla pe pozitia 0.

Antetul metodei este:

```
public char charAt(int <index>)
```

unde:- <index> - pozitia caracterului in sir, intre 0 si length() - 1.

Metoda charAt returneaza o valoare de tip char care reprezinta caracterul aflat la pozitia specificata de parametrul <index> in sir.

De exemplu:

```
String mesaj = "program in java";  
int lungimeSir = mesaj.length(); // lungimeSir are valoarea 15  
char carExtras = mesaj.charAt(3); // carExtras are valoarea 'n'
```

Extragerea unui subsir dintr-un sir de caractere

Metoda substring este folosita pentru a construi un nou sir de caractere obtinut prin extragerea unui subsir dintr-un sir de caractere. Exista doua variante ale acestei metode.

Antetul metodei **substring** in prima varianta este:

```
public String substring(int <indexInceput>)
```

unde:- <indexInceput> - pozitia in sirul de caractere de la care incepe extragerea subsirului.

Metoda substring returneaza un nou sir format dintr-un subsir al vechiului sir. Subsirusul incepe cu caracterul aflat pe pozitia specificata in parametrul <indexInceput> si se extinde pana la sfarsitul vechiului sir.

Antetul metodei **substring** in a doua varianta este:

```
public String substring(int <indexInceput>, int <indexSfarsit>)
```

unde:

- <indexInceput> - pozitia in sirul de caractere de la care incepe extragerea subsirului;

- <indexSfarsit> - pozitia in sirul de caractere pana la care se face extragerea (caracterul de pe aceasta pozitie nu este inclus in subsir)

Metoda substring in a doua varianta returneaza un nou sir format dintr-un subsir al vechiul sir. Subsirus incepe cu caracterul aflat pe pozitia specificata in parametrul <indexInceput> si se extinde pana la caracterul aflat pe pozitia specificata in parametrul <indexSfarsit> - 1.

Deci, lungimea noului sir creat este: <indexSfarsit> - <indexInceput>

De exemplu:

```
String sirVechi = "program in java";
String sirNou = sirVechi.substring(12); // sirNou are valoarea "java"
String sirNou1 = sirVechi.substring(10, 12); //sirNou1 are valoarea "in "
```

Cautarea primei aparitii a unui sir intr-un alt sir

Metoda startsWith este folosita pentru a cauta prima aparitie a unui sir in sirul curent. Antetul metodei este:

```
public boolean startsWith(String <sirCautare>)
```

unde:- <sirCautare> - sirul de cautat in sirul curent.

Metoda startsWith returneaza valoarea *true*, daca secventa de caractere din sirul specificat de parametrul <sirCautare> este aceeaasi cu cea din sirul curent, si valoarea *false*, altfel. Daca parametrul <sirCautare> este sirul vid, metoda returneaza, de asemenea, valoarea *true*.

Cautarea primei aparitii a unui sir intr-un alt sir folosind o pozitie de inceput a cautarii

Metoda startsWith poate fi folosita si pentru a cauta prima aparitie a unui sir in sirul curent folosind o pozitie de inceput a cautarii in sirul curent.

Antetul metodei **startsWith** in a doua varianta este:

```
public boolean startsWith(String <sirCautare>, int <indexInceputCautare>)
```

unde:

- <sirCautare> - sirul de cautat in sirul curent;

- <indexInceputCautare> - indexul, in sirul curent, de la care incepe cautarea.

Metoda startsWith returneaza valoarea *true* daca secventa de caractere din sirul specificat de parametrul <sirCautare> este aceeaasi cu cea din sirul curent, incepand de la pozitia specificata in parametrul <indexInceputCautare>. Altfel, metoda returneaza valoarea *false*.

Rezultatul aplicarii metodei, in acest caz, este acelasi cu cel al expresiei:

```
<sirCurent>.substring(<indexInceputCautare>).startsWith(<sirCautare>)
```

unde: - <sirCurent> specifica sirul curent in care se face cautarea.

Determinarea pozitiei primei aparitii a unui sir intr-un alt sir

Metoda `indexOf` este folosita pentru a determina pozitia de inceput, in sirul curent, la care se afla prima aparitie a unui sir.

Antetul metodei este:

```
public int indexOf(String <sirCautare>)
```

unde:- <sirCautare> - sirul de cautat in sirul curent.

Metoda `indexOf` returneaza indexul, din sirul curent, la care se afla prima aparitie a sirului specificat in parametrul <sirCautare>, daca secventa de caractere din sirul specificat de parametrul <sirCautare> este aceeași cu cea din sirul curent, și valoarea -1, altfel.

Determinarea pozitiei primei aparitii a unui sir intr-un alt sir folosind o pozitie de inceput a cautarii

Metoda `indexOf` poate fi folosita și pentru a determina indexul primei aparitii a unui sir in sirul curent folosind pozitia de inceput a cautarii in sirul curent.

Antetul metodei **`indexOf`** in a doua varianta este:

```
public int indexOf (String <sirCautare>, int <indexInceputCautare>)
```

unde:

- <sirCautare> - sirul de cautat in sirul curent;

- <indexInceputCautare> - indexul, in sirul curent, de la care incepe cautarea spre sfarsitul sirului curent.

Metoda `indexOf` returneaza indexul, din sirul curent, incepand de la pozitia specificata in parametrul <indexInceputCautare>, la care se afla prima aparitie a sirului specificat de parametrul <sirCautare>. Altfel, metoda returneaza valoarea -1.

Rezultatul aplicarii metodei, in acest caz, este același cu cel al expresiei:

```
<sirCurent>.substring(<indexInceputCautare>).indexOf(<sirCautare>)
```

unde: - <sirCurent> specifica sirul curent in care se face cautarea.

Cautarea ultimei aparitii a unui sir intr-un alt sir

Metoda `endsWith` este folosita pentru a cauta ultima aparitie a unui sir in sirul curent.

Antetul metodei este:

```
public boolean endsWith(String <sirCautare>)
```

unde:- <sirCautare> - sirul de cautat in sirul curent.

Metoda `endsWith` returneaza valoarea *true*, daca secventa de caractere din sirul specificat de parametrul <sirCautare> este aceeași cu cea din sirul curent, și valoarea *false*, altfel. Daca parametrul <sirCautare> este sirul vid, metoda returneaza, de asemenea, valoarea *true*.

Determinarea pozitiei ultimei aparitii a unui sir intr-un alt sir

Metoda `lastIndexOf` este folosita pentru a determina pozitia ultimei aparitii a unui sir in sirul curent. Aceasta metoda se poate aplica in doua variante.

Antetul metodei in prima varianata este:

```
public int lastIndexOf(String <sirCautare>)
```

unde: - <sirCautare> - sirul de cautat in sirul curent.

Metoda `lastIndexOf` returneaza indexul, din sirul curent, la care se afla ultima aparitie a sirului specificat in parametrul <sirCautare>, daca secventa de caractere din sirul specificat de parametrul <sirCautare> este aceeaasi cu cea din sirul curent, si valoarea -1, altfel.

Nota: Sirul vid (“”) cel mai din dreapta este considerat ca apare la indexul <sirCurent>.length().

Antetul metodei `lastIndexOf` in a doua varianta este:

```
public int lastIndexOf (String <sirCautare>, int <indexInceputCautare>)
```

unde:

- <sirCautare> - sirul de cautat in sirul curent;

- <indexInceputCautare> - indexul, in sirul curent, de la care incepe cautarea spre inceputul sirului curent.

Metoda `lastIndexOf` returneaza indexul, din sirul curent, incepand de la pozitia specificata in parametrul <indexInceputCautare>, la care se afla ultima aparitie a sirului specificat de parametrul <sirCautare>.Altfel, metoda returneaza valoarea -1.

Inlocuirea aparitiilor unui caracter dintr-un sir cu un alt caracter

Metoda `replace` este folosita pentru a construi un nou sir de caractere obtinut prin inlocuirea aparitiilor unui caracter cu un alt caracter, in sirul curent.

Antetul metodei este:

```
public String replace(char <caracterVechi>, char <caracterNou>)
```

unde:

- <caracterVechi> - caracterul de inlocuit;

- <caracterNou> - caracterul cu care se inlocuieste.

Metoda `replace` returneaza un sir nou de caractere construit cu caracterele inlocuite, date de parametrul <caracterNou>, daca caracterele specificate in parametrul <caracterVechi> sunt gasite in sirul curent sau returneaza un sir nou de caractere care este identic cu cel initial, daca caracterul specificat in parametrul <caracterVechi> nu apare in sirul curent.

De exemplu:

```
String sir1 = “Ionescu Ion”;
```

```
String sir2 = sir1.replace(‘I’, ‘J’); // sir2 va avea valoarea “Jonescu Jon”
```

Inlocuirea aparitiilor unei secvente de caractere (subsir), dintr-un sir, cu o alta secventa de caractere (subsir)

Metoda `replace` este folosita **si** pentru a construi un nou sir de caractere obtinut prin inlocuirea aparitiilor unei secvente de caractere cu o alta secventa de caractere, in sirul curent.

Antetul metodei **`replace`** in aceasta varianta este:

`public String replace(String <secventaVeche>, String <secventaNoua>)`

- <SecventaVeche> - secventa de caractere de inlocuit;
- <secventaNoua> - secventa de caractere cu care se inlocuieste.

Metoda `replace`, in aceasta varianata, returneaza un sir nou de caractere construit cu secventele de caractere inlocuite, date de parametrul <secventaNoua>, daca secventele de caractere specificate in parametrul <secventaVeche> sunt gasite in sirul curent sau returneaza un sir nou de caractere care este identic cu cel initial, daca secventa de caractere specificata in parametrul <SecventaVeche> nu apare in sirul curent.

Eliminarea caracterelor albe de la inceputul si sfarsitul unui sir de caractere

Metoda `trim` este folosita pentru eliminarea caracterelor albe de la inceputul si sfarsitul sirului de caractere curent.

Antetul metodei este:

`public String trim()`

Metoda `trim()` returneaza un nou sir de caractere din care sunt eliminate caracterele albe de la inceputul sau sfarsitul sirului curent, daca acestea exista in sirul curent, sau returneaza sirul curent, daca caracterele albe nu exista in sir. Daca in sirul curent exista numai caractere albe atunci sirul nou creat este sirul vid (“”).

Convertirea unui sir de caractere la un tablou de caractere

Metoda `toCharArray` converteste un sir de caractere la un nou tablou de caractere.

Antetul metodei este:

`public char[] toCharArray()`

Metoda `toCharArray` alocă o zonă de memorie pentru un tablou de caractere al cărui conținut este inițializat cu secvența de caractere reprezentată de sirul de caractere.

Următorul program (`ReplaceSiruri.java`) ilustrează modul de folosire a metodei **`replace`** în varianta înlocuirii unui subsir de caractere cu un alt subsir de caractere, citite de la tastatură, într-un sir de caractere, de asemenea citit de la tastatură, a metodei **`trim`** pentru înlocuirea caracterelor albe (`'\r'` și `'\n'`) dintr-o secvență de caractere, precum și a metodei **`toCharArray`** pentru convertirea sirului, din care s-au eliminat caracterele albe, la un tablou de caractere.

De asemenea, programul arata si modul de apelare a metodei **length()** pentru obtinerea lungimii unui sir de caractere.

```

1 package poo;
2 import java.io.*;
3 public class Lab4_12 {
4     public static void main(String[] args) throws IOException {
5         char[] caractere = new char[20];
6         for (int i=0; i <=19; i++) caractere[i] = ' ';
7         BufferedReader br1 = new BufferedReader(new InputStreamReader(System.in));
8         System.out.println ("Introduceti sirul supus modificarii:");
9         br1.read(caractere, 0, 20);
10        for (int i = 0; i <= (caractere.length - 1); i++)
11            System.out.print((int) caractere[i] + " ");
12        System.out.println();
13        String sirVechi = new String(caractere).trim();
14        char [] caractereVechi = sirVechi.toCharArray();
15        for (int i = 0; i <= (caractereVechi.length - 1); i++)
16            System.out.print((int) caractereVechi[i] + " ");
17        System.out.println();
18        for (int i=0; i <=19; i++) caractere[i] = ' ';
19        BufferedReader br2 = new BufferedReader(new InputStreamReader(System.in));
20        System.out.println ("Introduceti secventa de caractere de inlocuit:");
21        br2.read(caractere, 0, 20);
22        String subsirVechi = new String(caractere).trim();
23        char [] caractereSubsirVechi = subsirVechi.toCharArray();
24        for (int i = 0; i <= (caractereSubsirVechi.length - 1); i++)
25            System.out.print((int) caractereSubsirVechi[i] + " ");
26        System.out.println();
27        for (int i=0; i <=19; i++)
28            caractere[i] = ' ';
29        BufferedReader br3 = new BufferedReader(new InputStreamReader(System.in));
30        System.out.println ("Introduceti secventa de caractere cu care se face inlocuirea:");
31        br3.read(caractere, 0, 20);
32        String subsirNou = new String(caractere).trim();
33        char [] caractereSubsirNou = subsirNou.toCharArray();
34        for (int i = 0; i <= (caractereSubsirNou.length - 1); i++)
35            System.out.print((int) caractereSubsirNou[i] + " ");
36        System.out.println();
37        String sirNou = sirVechi.replace(subsirVechi, subsirNou);
38        System.out.println("Lungimea noului sir este:" + sirNou.length());
39        System.out.println("Sirul nou este: " + sirNou);
40    }
41 }

```

Dupa executia programului pe ecran se afiseaza urmatoarele rezultate:

```

Problems @ Javadoc Declaration Console
<terminated> Lab4_12 [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (9 nov. 2014, 11:40:14)
Introduceti sirul supus modificarii:
Ion Ionescu
73 111 110 32 73 111 110 101 115 99 117 13 10 32 32 32 32 32 32
73 111 110 32 73 111 110 101 115 99 117
Introduceti secventa de caractere de inlocuit:
Io
73 111
Introduceti secventa de caractere cu care se face inlocuirea:
Mari
|77 97 114 105
Lungimea noului sir este:15
Sirul nou este: Marin Marinescu

```

Convertirea unui caracter la un sir de caractere

Metoda valueOf converteste un caracter la un sir de caractere.

Antetul metodei este:

```
public static String valueOf(char <c>)
```

unde:- <c> - un caracter.

Metoda valueOf returneaza un sir de caractere cu lungimea de 1 continand un singur caracter dat de parametrul <c>.

Convertirea unui tablou de caractere la un sir de caractere

Metoda valueOf poate fi folosita si pentru conversia unui tablou de caractere la un sir de caractere.

Antetul metodei este:

```
public static String valueOf(char [] <tablouCaractere>)
```

Metoda valueOf returneaza un nou sir de caractere in care este copiat continutul tabloului de caractere. Modificari ulterioare ale tabloului de caractere nu afecteaza continutul noului sir de caractere creat.

Convertirea, in sirul de caractere, a literelor mari in litere mici

Metoda toLowerCase este folosita pentru a transforma literele mari in litere mici, in sirul curent.

Antetul metodei este:

```
public String toLowerCase()
```

Convertirea, in sirul de caractere, a literelor mici in litere mari

Metoda toUpperCase este folosita pentru a transforma literele mici in litere mari, in sirul curent.

Antetul metodei este:

```
public String toUpperCase()
```

Probleme propuse spre rezolvare

Lab8_1: Se da urmatorul tablou de cuvinte (siruri de caractere): "Martie", "Aprilie", "Luna", "An". Sa se afiseze câte un mesaj care sa specifice cu ce litera începe fiecare cuvânt dat, folosind instructiunea switch.

Lab8_2: Sa se scrie o functie pentru inlocuirea tuturor aparitiilor unui sir dat s1 printr-un alt sir s2 in cadrul unui sir s, folosind numai metode ale clasei String.

Lab8_3: Să se afișeze numărul de vocale dintr-un text scris cu litere mici, memorat într-o variabilă de tip șir de caractere.

Exemplu:

Date de intrare: Programarea calculatoarelor

Date de ieșire: Exista 12 vocale in text

Lab8_4: Să se afișeze cu litere mari un text dat, de maxim 255 caractere.

Exemplu:

Date de intrare text: liMBajUl jaVa

Date de ieșire LIMBAJUL JAVA

Lab8_5: Se citește un șir de caractere. Să se afișeze litera cea mai des întâlnită.

Exemplu:

Date de intrare text: carte de informatica si programare

Date de ieșire: a apare de 5 ori

Lab8_6: Să se afișeze toate prefixele și sufixele unui cuvânt citit de la tastatură.

Exemplu:

Date de intrare: informatica

Date de ieșire:

| prefixe | sufixe |
|-------------|-------------|
| i | a |
| in | ca |
| inf | ica |
| info | tica |
| infor | atica |
| inform | matica |
| informa | rmatca |
| informat | ormatica |
| informati | formatica |
| informatic | nformatica |
| informatica | informatica |

Lab8_7: Se citește de la tastatură un caracter c și un text de maxim 100 de caractere. Afișați de câte ori apare caracterul c în cadrul textului. Literele mari se vor considera diferite de literele mici.

Exemplu:

Date de intrare:

Acesta este un simplu exemplu

Caracterul e

Date de ieșire: Litera ‘e’ apare de 5 ori

Lab8_8: Se citește de la tastatură un șir de maxim 100 de caractere format numai din litere și cifre. Afișați numărul literelor mari, numărul literelor mici și numărul caracterelor de tip cifră din textul dat.

Exemplu:

Date de intrare:

S-a nascut la Targu Jiu in 1968 pe 21 martie

Date de ieșire: Se gasesc 25 de litere mici, 2 litere mari si 6 cifre

Lab8_9: Să se scrie un program Java care verifică dacă două șiruri de caractere sunt anagrame. Două șiruri sunt anagrame dacă conțin aceleași caractere, dar într-o ordine diferită.

Exemplu:

Introduceti un sir de caractere: abracadabra

Sirul rezultat: abrcd

Lab8_10: Să se scrie un program Java care verifică dacă două șiruri de caractere sunt anagrame. Două șiruri sunt anagrame dacă conțin aceleași caractere, dar într-o ordine diferită.

Exemplu:

Introduceti primul sir de caractere: adrian

Introduceti al doilea sir de caractere: nairda

Cele doua siruri sunt anagrame.

Introduceti primul sir de caractere: hello

Introduceti al doilea sir de caractere: world

Cele doua siruri nu sunt anagrame.

Bibliografie:

[1] <http://www.pbinfo.ro> *Descrierea site-ului: “www.pbinfo.ro îți propune să rezolvi probleme de informatică, cu evaluator automat. Știi pe loc dacă soluția ta este corectă sau dacă trebuie să mai lucrezi la ea. Problemele sunt grupate după programa de informatică pentru liceu. Dar nu trebuie să fii la liceu ca să rezolvi aceste probleme. Poți fi elev de gimnaziu, student, profesor sau pur și simplu pasionat de informatică. De fapt, trebuie doar să vrei!!”*

[2] <https://www.runceanu.ro/adrian>

[3] Adrian Runceanu „Programarea și utilizarea calculatoarelor”, Editura Academica Brâncuși din Târgu-Jiu, 2003, ISBN 973-8436-44-3

[4] Adrian Runceanu, Mihaela Runceanu, „Noțiuni de programare – limbajul C++”, Editura Academica Brâncuși din Târgu-Jiu, 2012, ISBN 978-973-144-550-2

[5] Adrian Runceanu, Mihaela Runceanu, „Algoritmi implementati in limbajul C++. Volumul I – Algoritmi elementari”, Editura Academica Brâncuși din Târgu Jiu, 2021, ISBN 978-606-9614-06-8