

Дәріс-7. Java тілінде сөз тіркестерін пайдалану

Жоспар:

- 1 String класы
- 2 StringBuffer класы
- 3 String класы

Java жүйелік кітапханасының java.lang пакетінде анықталған, сөз тіркестерімен жұмыс істеуге арналған String және StringBuffer кластары бар. Бұл кластар final модификаторымен жарияланған, сондықтан олар ары қарай пайдаланылатын сөз тіркестерінің қасиеттері бар өз кластарын жасай алмайды.

String класы бірнеше конструкторларды сүйемелдей алады, олар: **String()**, **String(String str)**, **String(char[] c)**, **String(byte asciichar[])**. Бұлар класс объектілерін инициалдау үшін пайдаланылады.

String класы объектісін класс нұсқауышына бір тіркестің мәнін меншіктеу арқылы немесе конструктор мен new сөзін пайдалану жолымен жасауға болады, мысалы :

```
String s1 "Строка"; немесе  
String s1 = new String ("Строка");
```

String класының сөз тіркестерімен жұмыс істеуге арналған тәсілдері
String s=new String(); - бос тіркес құрастыру; **char chars[]={‘K’,‘A’, ‘З’, ‘Н’,‘У’};**

String s1=new String(chars); - String типті объект құрастырып, оған мән беру; немесе

String s2="UKGU"; int length(); -сөз тіркестерінің ұзындығын анықтау
String s1 = "Университетімізге " + age + " жыл."; - жолдарды біріктіру;

Мысал:

```
import java.io.*; class lengthDemo {  
public static void main(String args[]) throws IOException {  
    PrintWriter pw=new PrintWriter (new  
    OutputStreamWriter(System.out,"Cp866"), true);  
    int age=70;  
    String s = "Шымкент"; pw.println(s.length());  
    String s1 = "Университетімізге " + age + " жыл."; pw.println(s1);  
    }  
}
```

Нәтижесі: 6

Университетімізге 70 жыл

Сөз тіркестерінен символды қиып алу тәсілдері char charAt(int n);
тіркестен нөмірі көрсетілген символды қиып алу;

Void getChars(int start, int end, char buf, int n); - бірнеше қатар орналасқан символдарды қиып алу

Start – бірінші символ нөмірі

End – соңғы символ нөмірі

Buf – қиып алынған символдарды меншіктейтін жиым(массив)

n – қиып алынған символдардың жиым элементтерінің қандай нөмірінен бастап жазылатындығын көрсетеді

Символдарды қиып алу мысалдары

```
import java.io.*; class getCharsDemo {
```

```
public static void main(String args[]) throws IOException{
```

```
PrintWriter pw=new PrintWriter (new OutputStreamWriter (System.out,  
"Cp866"), true);
```

```
Strings = "Демонстрация метода getChars на примере."; // заданная строка  
pw.println(s);
```

```
pw.println(s.charAt(3)); // печать 3-символа intstart = 13;
```

```
int end = 18;
```

```
char buf[] = new char[end - start]; s.getChars(start, end, buf, 0);  
pw.println(buf); } }
```

Мысал нәтижесі :

Демонстрация метода getChars на примере. (3 орындағы символ – тіркестің 4-символы)

метод (тіркестің 13-18-символдары, яғни 14-19-орындағы символдар)

Сөз тіркестерін салыстыру

Boolean equals (objects); - екі тіркесті салыстыру, салыстыру кезінде бас және кіші әріптер әр түрлі болып саналады;

Boolean equals IgnoreCase(objects); - екі тіркесті салыстыру, салыстыру кезінде кіші және бас әріптердің айырмашылығы жоқ;

Сөз тіркестерін реттеу (сортировка)

IntcompareTo(Stringstr); - str тіркесін регистрді есепке ала отырып,

алфавит бойынша (лексикографикалық) реттейді.

IntcompareToIgnoreCase (Stringstr); - str тіркесін регистрді есепке алмай–бас әріппен кіші әріп бірдей), алфавит бойынша (лексикографикалық) реттейді.

Егер тәсілдің қайтарған мәні теріс болса, онда салыстырылған тіркес параметр тіркестен кіші, оң болса – үлкен, 0 болса тең болғаны.

Сөз тіркестеріндегі символды іздеу

intindexOf(charch); - ізделінетін символдың тіркестегі бірінші кездескен орнын анықтайды (басынан бастап іздеу);

IntlastIndexOf(charch); - ізделінетін символдың тіркестегі соңғы кездескен орнын анықтайды (артынан іздеу);

Осы тәсілдердің көмегімен бір символ емес сол тіркестегі ішкі тіркестің бірінші және соңғы символдарының орнын да анықтауға болады. Сонымен бірге ізделінетін символды немесе ішкі тіркесті нешінші позициядан бастап іздеу керек екендігін де көрсетуге болады.

Сөз тіркестерін түрлендіру

Stringsubstring(intstartindex, intendindex); - берілген сөз тіркесінен бірнеше символды немесе ішкі тіркесті қиып алу;

String concat(string str); - екі сөз тіркесін біріктіру;

String replace(char original, char replecement); - берілген тіркестегі бір символды көрсетілген символмен алмастыру;

String toLowerCase(); - бас әріптерді кіші әріптермен алмастыру; **String toUpperCase();** - кіші әріптерді бас әріптермен алмастыру;

String trim(); - берілген сөз тіркесінің алдындағы және соңындағы бос орындарды алып тастайды;

StringBuffer класы

StringBuffer класының **String** класынан айырмашылығы бастапқы берілген жолдың ұзындығын өзгертуге болады.

StringBuffer класының тәсілдері

length(); - **StringBuffer** ағымдағы ұзындығын анықтау; **setLength(int len);** - буфердің ұзындығын алдын ала орнату; **capacity();** - **StringBuffer**-ге бөлінген жады көлемін анықтау;

ensureCapacity(int capacity); - буферге бөлінетін жады көлемін алдын ала анықтау;**charAt(int n);** - n-ші орындағы символды анықтау;

setCharAt(int n, char ch); - n-ші орындағы символды ch символына алмастыру;

insert(int index, string str); - берілген сөз тіркесі ішіне str тіркесін көрсетілген index-тен бастап енгізу

reverse(); - берілген тіркесті кері оқу;

delete(int startindex, int endindex); - берілген тіркестегі startindex және endindex аралығындағы символдарды алып тастау

deleteCharAt(int k); - k позициясындағы символды өшіру;

replace(int startindex, int endindex, string str); - берілген тіркестегі startindex және endindex аралығындағы символдарды str ішкі тіркесімен алмастыру.

Бақылау сұрақтар:

1. String класы объектісін қандай жолымен жасауға болады?
2. StringBuffer класының String класынан айырмашылығы қандай?
3. StringBuffer класының тәсілдері қандай?