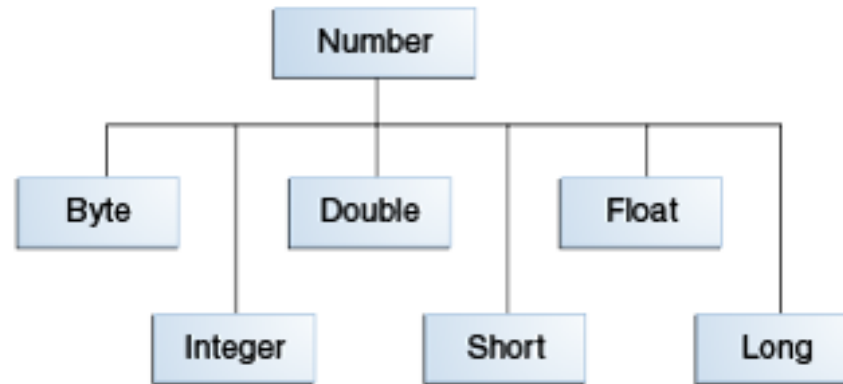


# Brojevi i Stringovi

# Klase brojeva



- Pored ovih postoje i BigDecimal, BigInteger, AtomicInteger i AtomicLong
- Svaka ima statički definisano polje MIN\_VALUE, MAX\_VALUE (za pozitivan opseg vrednosti)

# Metode nasleđene iz klase Number

Metoda	Opis
<code>byte byteValue()</code> <code>short shortValue()</code> <code>int intValue()</code> <code>long longValue()</code> <code>float floatValue()</code> <code>double doubleValue()</code>	Konverzija broja u odgovarajući primitivni tip
<code>int compareTo(Byte anotherByte)</code> <code>int compareTo(Double anotherDouble)</code> <code>int compareTo(Float anotherFloat)</code> <code>int compareTo(Integer anotherInteger)</code> <code>int compareTo(Long anotherLong)</code> <code>int compareTo(Short anotherShort)</code>	Poredi broj sa prosleđenim brojem i vraća <0, 0 ili >0. 0 ako su isti, <0 ako je ovaj broj pre, a >0 ako je prosleđeni broj pre.
<code>boolean equals(Object obj)</code>	Proverava da li je neki broj jednak prosleđenom broju po vrednosti. Kod brojeva se može koristiti <code>==</code> . Ovo sa <code>==</code> ne važi u opštem slučaju za bilo koju klasu, jer se na taj način nekad proverava samo referenca.

# Metode za konverziju u klasi Integer (slično i za ostale)

Metoda u klasi Integer	Opis
<code>static Integer decode(String s)</code>	Pretvara String (tekst) u ceo broj, može da radi i za druge osnovne, a ne samo za dekadnu, npr. <code>Ox452</code>
<code>static int parseInt(String s)</code>	Pretvara tekst u ceo broj, samo dekadna osnova
<code>static int parseInt(String s, int radix)</code>	Pretvara tekst u ceo broj za datu osnovu
<code>String toString()</code>	Vraća tekstualnu reprezentaciju ovog broja ( <code>this</code> )
<code>static String toString(int i)</code>	Isto što i ovo iznad samo statička metoda
<code>static Integer valueOf(int i)</code>	Pretvara primitivni tip u klasu (boxing – umotavnje)
<code>static Integer valueOf(String s)</code>	Pretvara prosleđeni tekst u ceo broj
<code>static Integer valueOf(String s, int radix)</code>	Kao ovo iznad, samo za datu osnovu

# Formatiranje ispisa broja

```
long n = 461012;
```

```
System.out.format("%d%n", n); // --> "461012"
```

```
System.out.format("%08d%n", n); // --> "00461012"
```

```
System.out.format("%+8d%n", n); // --> " +461012"
```

```
System.out.format("%%,8d%n", n); // --> " 461,012"
```

```
System.out.format("%+,8d%n%n", n); // --> "+461,012"
```

```
double pi = Math.PI;
```

```
System.out.format("%f%n", pi); // --> "3.141593,,"
```

```
System.out.format("%.3f%n", pi); // --> "3.142"
```

```
System.out.format("%10.3f%n", pi); // --> " 3.142"
```

```
System.out.format("%-10.3f%n", pi); // --> "3.142"
```

# Matematičke operacije

- Klasa Math ima preko 40 statičkih metoda i neke bitne konstante
- Neke od njih su:
  - `Math.abs(broj)`, `Math.floor(broj)`, `Math.round(broj)`,...
  - `Math.log(broj)`, `Math.exp(broj)`,...
  - `Math.PI`, `Math.E`,...
  - `Math.cos(broj)`, `Math.tan(broj)`,...
  - `Math.random()` – vraća slučajan realan broj iz  $[0,1]$ .  
**Kako formirati ceo slučajan broj iz opsega  $[m,n]$ ?**

# Karaktereri

- Primitivni tip char
- Objektni tip Character

```
char ch = 'a';
```

```
// Unicode karakter za grčko slovo omega
```

```
char uniChar = '\u03A9';
```

```
// niz karaktera, kao u C-u
```

```
char[] charArray = { 'a', 'b', 'c', 'd', 'e' };
```

```
Character ch = new Character('a')
```

# Korisne informacije o klasi Character

- isLetter(ch)
- isDigit(ch)
- isWhitespace(ch)
- isUpperCase(ch)
- isLowerCase(ch)
- toUpperCase(ch)
- toLowerCase(ch)
  
- Neki specijalni karakteri:
  - \t, \n, \', \", \\,...



# Stringovi (tekst)

- Kreiranje String-a:

```
//direktno
```

```
String pozdravString1= "Zdravo svete!";
```

```
//od niza karaktera
```

```
char[] pozdravNiz= { 'z', 'd', 'r', 'a', 'v', 'o' };
```

```
String pozdravString2= new String(pozdravNiz);
```

```
System.out.println(pozdravString2);
```

# Neke korisne operacije nad Stringom

```
String str="Neki string.\n";
```

```
System.out.println(str.length()); //?
```

```
System.out.println(str.charAt(4)); //?
```

```
System.out.println(str.concat(" Jos jedan. "))//?
```

```
System.out.println((str+"Novi. "+str).length());
```

# Formatiranje Stringa

```
String fs;
```

```
fs = String.format("Broj %f i String %s", 2.34, "abc");
```

```
System.out.println(fs);
```

|||

```
System.out.printf("Broj %f i String %s", 2.34, "abc");
```

# Konverzija Stringa u broj i obrnuto

```
Integer i1 = 2;
```

```
int i2 = 4;
```

```
String s1 = "24";
```

```
String s2 = "35.3";
```

```
System.out.println(i1.toString());
```

```
System.out.println(String.valueOf(i2));
```

```
int i3 = Integer.parseInt(s1);
```

```
Float f1 = Float.parseFloat(s2);
```

```
Integer i4 = Integer.parseInt(s2); //?
```

# Manipulacija Stringovima

```
String str1 = "Neki tekst!";
```

```
String str2 = str1.substring(1, 4);//?
```

- Pogledati i sledeće metode:
  - `String[] split(String separator)`
  - `int indexOf(char c)`
  - `String toLowerCase(String s)`
  - `String replace(String s1, String s2)`

# Poređenje Stringova

```
String s1 = "Ovo je jedan nekreativan string.";
```

```
String s2 = "Ovo je";
```

```
String s3 = "Ovo je treći.";
```

```
System.out.println(s1.startsWith(s2));//?
```

```
System.out.println(s1.compareTo(s2)); //?
```

```
System.out.println(s2.compareTo(s3)); //?
```

# StringBuilder

- Upotrebljavati kada se kreira neki komplikovani string, jer je efikasnije, npr:

```
String s1="";  
for(int i=0; i<1000; i++)  
    s1=s1+String.valueOf(i);  
System.out.println(s1);
```

```
//Bolje ovako  
StringBuilder sb=new StringBuilder();  
for(int i=0; i<1000; i++)  
    sb.append(i);  
System.out.println(sb.toString());
```

# Klasa Scanner za unos

```
//Kreiramo Scanner objekat i povezujemo ga na standardni //ulaz  
(konzolu) System.in
```

```
Scanner skener=new Scanner(System.in);
```

```
//učitavamo ceo red (liniju)
```

```
String unos = skener.nextLine();
```

```
System.out.println(unos);
```

```
//učitavamo broj
```

```
Integer broj = skener.nextInt();
```

```
System.out.println(broj);
```

```
//obavezno zatvoriti Scanner
```

```
skener.close();
```



# Zadatak

- Korisnik sa konzole unosi u svakom redu informacije o proizvodima na sledeći način:  
šifra [ime proizvoda] cena [na lageru]
- Pretpostaviti da se neće uneti više od 100 proizvoda
- Proizvode smeštati u niz klase Proizvod koja ima sve potrebne informacije

# Zadatak - nastavak

- Korisnik završava unos kada ukuca ključnu reč KRAJ
- Ne sme se desiti da korisnik unese istu šifru, ili identičan naziv proizvoda koji je već unet ranije:
  - U tom slučaju ispisati poruku o grešci i izaći iz programa

# Zadatak - nastavak

- Nakon što je korisnik korektno sve uneo potrebno je na konzoli ispisati sledeće informacije
  - Spisak svih unetih proizvoda u formatu:  
naziv \t [cena na 2 decimale] \t [preostalo]  
(ovo definisati u toString metodi klase Proizvod)
  - Nakon toga ispisati i sledeće statistike:
    - Prosečnu cenu proizvoda
    - Najskuplji i najjeftiniji proizvod
    - Ukupnu vrednost celog inventara
    - Proizvod koji ima nadjuži naziv