

1 Стандардне класе омотачи примитивних типова

Класе омотачи примитивних типова података налазе се у пакету *java.lang*: *Boolean*, *Byte*, *Character*, *Short*, *Integer*, *Long*, *Float*, *Double*.

Објекти ових класа обавијају вредност припадајућег примитивног типа, која се чува као податак (атрибут) објекта.

Свака од ових класа има метод *toString()* (статички и нестатички) за добијање *String*-репрезентације вредности примитивног типа.

Постоје и методи за издвајање вредности примитивног типа из *String* објекта.

Поред тога, свака од класа обезбеђује и разне друге корисне методе и константе.

Издајање вредности примитивног типа из стринга

Статички методи *parse*(String s)* прихватају *String*-репрезентацију вредности примитивног типа као аргумент и враћају еквивалентну вредност тог примитивног типа (уместо знаке '*' наводи се конкретан тип).

Класа *Integer* садржи различите статичке методе за издвајање целобројне вредности из стринга:

- *int parseInt(String s)*
издваја и враћа вредност као декадни цео број; стринг *s* сме да садржи само декадне цифре, са евентуалним предзнаком '+' или '-', водеће нуле се занемарују, али не и водеће белине или белине на крају стринга
- *int parseInt(String s, int radix)*
издваја и враћа вредност као цео број у систему са датом основом *radix*
- *int parseUnsignedInt(String s)*
издваја и враћа вредност као цео неозначен број
- *int parseUnsignedInt(String s, int radix)*
издваја и враћа вредност као цео неозначен број у систему са датом основом *radix*

У класи постоји и статички метод *decode(String s)* који издваја декадни, октални (водећа нула) или хексадекадни (префикс '#', '0x', или '0X') цео број из стринга. "

```
String str = "123";  
int broj = Integer.parseInt(str);
```

Уколико стринг који је аргумент метода *parse*()* не одговара формату вредности примитивног типа коју метод може да препозна и издвоји, метод избацује изузетак типа ***NumberFormatException***.

И друге класе омотачи садрже сличне методе. На пример, метод

```
double parseDouble(String s)
```

у класи *Double* издваја и враћа вредност као реалан број типа *double*. Метод занемарује водеће нуле или нуле на крају стринга, као и белине. Слова суфикси ('f', 'd') која означавају тип реалног литерала не утичу на резултат метода.

Издајање вредности примитивног типа из објекта класе омотача

Свака класа омотач такође дефинише и метод **Value()* који враћа вредност примитивног типа из објекта те класе (уместо знаке '*' наводи се конкретан тип):

```
Double broj;  
double vrednost = broj.doubleValue();
```

Свака класа омотач има `static` константе `MAX_VALUE` и `MIN_VALUE` које дефинишу највећу и најмању вредност које могу бити представљене тим типом.

Класе омотачи реалних типова података (floating-point) имају дефинисане и константе: `POSITIVE_INFINITY`, `NEGATIVE_INFINITY` и `NaN` које се могу користити за поређење приликом израчунавања.

Постоје и статички методи `isFinite(double)`, `isInfinite(double)` и `isNaN(double)` за проверу floating-point вредности (да ли је коначна или бесконачна вредност, односно да ли је у питању вредност која није број).

Статички методи за поређење две вредности примитивног типа, одређивање максимума и минимума

Класа `Integer`:

- `int compare(int a, int b)`
враћа 0 ако су бројеви једнаки, вредност >0 ако је $a > b$, односно <0 ако је $a < b$
- `int compareUnsigned(int a, int b)`
као и метод `compare` само што посматра бинарне репрезентације бројева као репрезентације неозначених целих бројева
- `int max(int a, int b)`
максимум два цела броја
- `int min(int a, int b)`
минимум два цела броја

Слично важи и за остале класе омотаче.

Статички метод `valueOf()`

Статички метод `valueOf()` за дати аргумент примитивног типа враћа одговарајући објекат класе омотача.

```
double x = 12.5;
Double xD = Double.valueOf(x);
```

Нестатички метод `compareTo()`

Нестатички метод `compareTo()` пореди објекат класе омотача над којим се позива са другим објектом те класе чија референца је прослеђена методу као аргумент:

```
double x = 12.5;
double y = 15.7;
Double xD = Double.valueOf(x);
Double yD = Double.valueOf(y);
int rez = xD.compareTo(yD);
```

Имплицитно кастовање вредности примитивног типа

Ако се методу, који као аргумент захтева референцу на објекат класе омотача, проследи вредност одговарајућег примитивног типа, компајлер ће (уколико околности дозвољавају) извршити имплицитно кастовање (autoboxing) вредности примитивног типа у одговарајући класни тип.

Компајлер такође ради и инверзно кастовање – референцу на објекат класе омотача кастује у вредност примитивног типа коју он обавија (unboxing).

2 Класа *java.lang.Math*

Класа садржи само **статичке** методе за реализовање различитих математичких операција и функција. Постоје и две константе, које су такође **статички** чланови класе:

- `static final double PI`
- `static final double E`

Неки од метода ове класе:

- `double abs(double)`
- тригонометријске функције: `double sin(double)`, `double cos(double)`, ...
- `double ceil(double)`, `double floor(double)`
- `double pow(double a, double b)`, `double sqrt(double)`, `double cbrt(double)`
- `double exp(double)`, `double log(double)`, `double log10(double)`
- `double hypot(double x, double y)`
- `max(a, b)`, `min(a, b)` – а и b могу бити типа `int`, `long`, `float`, `double`
- ...