String

Kuliah IF2210- Sem 2 -2013/2014

Risa S. Perdana & Inggriani Liem

String

- String merupakan kelas khusus di Java
  - Java memperlakukan String tidak seperti objek lain
  - Bukan tipe dasar, tapi bagian dari bahasa Java
- String dibahas karena akan banyak dipakai (misalnya untuk menuliskan output)

```
String s = "Hello World";
```

- String merupakan sebuah kelas, dan di dalamnya ada beberapa method, misalnya:
  - `s.length()` --> panjang string
  - `s.substr(0, 2)` --> mengembalikan "He" (karakter ke 0 sampai ke-2 [*bukan* sampai dengan])

Operator String

- Selain memiliki method, string juga memiliki operator "+"
- Operator + akan mengkonversi float/integer secara otomatis jika digabung dengan string

```
/*konversi otomatis 2 ke string*/
String s = "Banyaknya "+ 2;
int z = 4;
String s = "Ada " + z + " buah ";
String s = 5; /*tidak boleh */
```

Perbandingan String

- Untuk membandingkan string tanpa membedakan case, gunakan equalsIgnoreCase
- Untuk membandingkan urutan String menurut kamus (*lexicographicaly*), gunakan method compareTo,
- Contoh:

```
str1.compareTo(str2)
```

  - Nilai kembalian:
    - 0 jika string sama
    - Suatu nilai negatif jika str1 < str2
    - Suatu nilai positif jika str1 > str2

Konversi String ke bilangan

```
int x = Integer.parseInt("2");
long l = Long.parseLong("2L");
double d = Double.parseDouble("2.0");
float f = Float.parseFloat("2.0f");

String ival = Integer.toString (2);
String lval = Long.toString(2L);
String dval = Double.toString(2.0);
String fval = Float.toString(2.0f);
```

Literal String

- Literal string merupakan instans dari objek string
- Method boleh dipanggil langsung dari Literal:
  - `"Hello".length()` menghasilkan 5

Sequence Escape String

- Serangkaian karakter diawali \ (backslash) untuk mengetikkan karakter khusus, escape berikut sama dengan C/C++
  - `\n` : newline
  - `\t` : karakter tab
  - `\\` : backslash
  - `\”` : double quote
  - `\'` : apostrophe
- Escape khusus Java: `\udddd`: karakter dalam unicode dddd adalah digit heksadesimal (0-9, A-F)

Sifat Immutable String

- String sebenarnya *immutable* (tidak bisa diubah)
- Dalam instruksi sbb:

```
String a = "hello";
a = a + " world";
```

- Sebuah objek baru diciptakan, objek lama dibuang (untuk dipungut oleh garbage collector). a menunjuk ke objek yang baru

Operasi `String` Tidak Optimal

- `String` baru diciptakan (string yang lama tetap ada di memori, dan dibuang ketika terjadi garbage collection)
- Untuk operasi yang banyak melibatkan perubahan string, sebaiknya menggunakan `StringBuffer`

StringBuffer

- `StringBuffer` mirip dengan `String`
- Sifatnya mutable
- Tidak ditangani secara transparan oleh Java (harus dilakukan secara manual)
- Lebih cepat untuk manipulasi string yang memerlukan perubahan pada `String`.

Sifat mutable `StringBuffer`

- Untuk mengubah `StringBuffer` tidak perlu objek baru
  - Contoh :

```
StringBuffer nama = new StringBuffer("mataku");
nama.setCharAt(4, 'm');
```

- Untuk mengubah `String` selalu butuh objek baru (objek lama diubah melalui assignment)

Method yang penting

- Beberapa method `String` dan `StringBuffer` yang penting adalah:
  - `length()`: panjang string
  - `replace()`: mengganti suatu karakter
  - `charAt()`: mengakses karakter di posisi tertentu
  - `trim()`: menghilangkan spasi di awal dan di akhir string
- Perhatikan bahwa meskipun namanya sama, sifat keduanya berbeda
  - `String` menciptakan objek baru, sedangkan `StringBuffer` tidak

Membandingkan `String`

- Method `equals()` membandingkan string untuk memeriksa kesamaan
- Method `equalsIngnoreCase()` melakukan hal yang sama, tapi besar kecil huruf tidak diperhatikan
- Method `compareTo()` menghasilkan 0 jika string sama, >0 jika String1>String2 dan <0 jika String1<String2

Bacaan tentang Mutable and Immutable Object

- http://www.javaranch.com/journal/2003/04/immutable.htm

Method String toString()

- Override method `toString()` untuk mencetak objek dengan lebih baik
- Misal, untuk kelas `Point`, isi methodnya:

```
String toString() {
   return "[" + x + ", " + y + "]";
}
```

- Dengan method di atas, Point bisa dicetak dengan mudah:

```
Point p = new Point(1,2);
System.out.println(p); /*mencetak point*/
```

- output potongan kode di atas: `[1, 2]`