Lezione 3

Un primo strumento utilizzabile per eseguire piccoli test è la `jshell`. E' una console dove è possibile eseguire comandi Java in modo da ricevere output immediato.

Java è un linguaggio **fortemente tipato**: ad ogni valore viene associato un tipo specifico; l'algoritmo di typing, presente nel compiler, rende possibile il riconoscimento del tipo di un dato valore, bloccando la compilazione se un operazione non è compatibile con un certo tipo di dato.

Ogni tipo di dato ha:

- Un nome (`int`, `boolean`, ecc...);
- Un insieme di dati, con relativa grandezza;
- Le operazioni possibile su tale tipo di dato;
  Operazioni del tipo

```
5 + true = "qualcosa"
```

non è considerato un programma Java, in quanto il `+` non permette operazioni tra tipi diversi.

Il tipo `boolean`

- Nome del tipo: `boolean`
- Valori ammessi: `true` e `false`
- Ha un operatore unario, il NOT `!`
- Ha vari operatori binari: AND `&`, OR `|`, XOR `^`, AND-C `&&`, OR-C `||`.
  - Gli operatori AND-C e OR-C permettono la short-circuit evaluation.
- Sono restituiti da operazioni di confronto numerici `>`, `<`, `>=`, `<=`, o da operazioni di uguaglianza (funziona su tutti i tipi di dato) `==`.
- Ha un operatore ternario, `<exp1> ? <exp2> : <exp3>`.
  - `<exp2>` e `<exp3>` devono essere dello stesso tipo

I tipi numerici

`byte`, `int`, `long`, `short`:

- Valori ammessi: numeri positivi e negativi. Massimi e minimi sono definiti dal tipo di dato specifico

- Hanno moltissimi operatori:
  - Di base, `+`, `-`, `/`, `*`, `%`
  - Bit a bit, poco usati;
- Sono rappresentati in complemento a 2 (ossia vanno da -2^n-1 a +2^n-1 - 1 con n numero di bit che richiedono).
- E' possibile rappresentare i numeri anche in esadecimale o in ottale, anche se si usa molto poco.
- Ogni numero specificato senza tipo è automaticamente considerato come `int`; qualsiasi valore più grande del massimo valore ammissibile da `int` restituisce un errore semantico.
  `float`, `double`:
- Valori ammessi: numeri positivi e negativi a virgola mobile in formato IEEE-754.
- Stessi operatori di base dei tipi numerici interi
- I numeri con la virgola con tipo non specificato vengono considerati `double`.
- È possibile incorrere in errori di arrotondamento utilizzando variabili in virgola mobile. È possibile eseguire delle conversioni di tipi, ma è possibile farlo solo da tipi più "piccoli" a tipi più grandi
  `char`:
- Non c'entrano con i tipi numerici, ma sono implementati tramite lo standard UTF-16

Array in Java

Gli array in Java sono oggetti, ossia possiedono metodi che si trovano all'interno della libreria standard di Java. Essi si dichiarano nel seguente modo:

```
<type>[] <name> = new <type>[(size)]({values});
```

- `type` è il tipo di dati che l'array conterrà
- `name` è il tipo
- `size` è la grandezza del vettore. Non va specificata se si specificano i valori presenti al suo interno (`values`)
  Se inizializzato con `size`, quindi senza `values`, tutti i valori vengono inizializzati sulla base dei valori di inizializzazione del tipo di dato dell'array.

Ad esempio, un array di double chiamato "pippo" si indica in questo modo:

```
double[] pippo;
```

Esiste una specifica classe, `java.util.Arrays`, che possiede una serie di metodi che permettono di modificare / gestire e operare con gli array in facilità; per esempio, per

stampare l'array si può usare:

```
int array[] = new array[]{10, 20, 30};
String s = java.util.Arrays.toString(array);
```

```
[10, 20, 30]
```

Oppure, utilizzando `import` è possibile rendere implicito l'uso di `java.util`:

```
import java.util.Arrays
...

int array[] = new array[]{10, 20, 30};
String s = Arrays.toString(array);
```

```
[10, 20, 30]
```