

Java memo

While(true)...break;

Scanner reader = new Scanner(System.in);

int a = Integer.parseInt(reader.nextLine());

accessor (get) : return something

mutator (set) : set a value

constructeur = même nom que class

return "" + number = cast en string

int : 4 octets

double : 8 octets

float : 4 octets

byte, char, boolean

% = modulo = reste division entière

if (data instanceof int)

```
for(Iterator<Track> it = tracks.iterator();
    it.hasNext(); ) {
    Track track = it.next();
    if(track.getArtist().equals(artist)) {
        it.remove();
    }
}
```

```
public int getCount(String animal, int spotter, int period){
return sightings.stream()
.filter(sighting -> animal.equals(sighting.getAnimal()))
.map(sighting -> sighting.getCount())
.reduce(0, (total, count) -> total + count);
}
```

The break statement has no effect on if statements. It only works on switch, for, while and do loops. So in your example the break would terminate the for loop.

Interface : toutes les méthodes sont publiques.

```
try{}
catch{}
finally{}
```

RuntimeException -> unchecked (not expected, not recoverable)

Exception -> checked (expected, maybe recoverable)

Assertions -> internal checks -> don't include functionality

Runtime errors often lead to program failure.

Exceptions provide a reporting and recovery mechanism.

/!\ Héritage que si constructeur pas private

! On ne peut pas créer de sous classe a partir d'une final class

! Les méthodes final ne peuvent pas être @Override

! Les méthodes d'une interface sont abstract

\i/

Instanceof n'est pas orienté objet, Notez que si vous devez utiliser cet opérateur très souvent, c'est généralement un indice que votre design a quelques défauts. Donc, dans une application bien conçue, vous devriez utiliser cet opérateur aussi peu que possible (bien sûr, il y a des exceptions à cette règle générale).

```
public class PetOwner {
    public void talking(Object obj) {
        if (obj instanceof Hamster) {
            Hamster hamster = (Hamster) obj;
            hamster.talk();
        } else if (obj instanceof Axolotl) {
            Axolotl axolotl = (Axolotl) obj;
            axolotl.talk();
        }
    }
    public static void main(String[] args) {
        PetOwner pwner = new PetOwner();
        pwner.talking(new Hamster());
        pwner.talking(new Axolotl());
    }
}
```

Foo f = new Bar();

Foo = type statique -> méthodes statiques

Bar = type dynamique