

Curingas delimitados (bounded wildcards)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Problema 1

Vamos fazer um método para retornar a soma das áreas de uma lista de figuras.

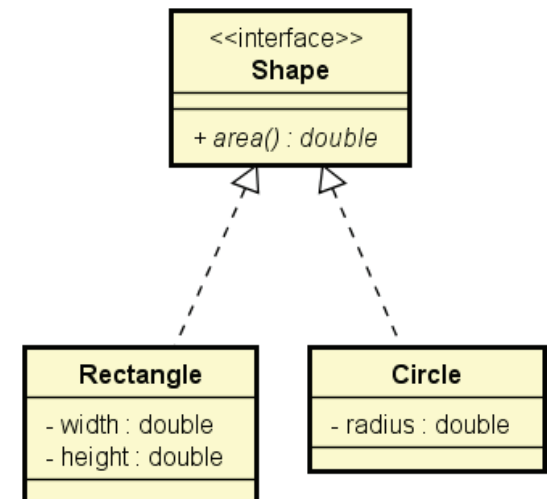
Nota 1: soluções impróprias:

```
public double totalArea(List<Shape> list)
```

```
public double totalArea(List<?> list)
```

Nota 2: não conseguiremos adicionar elementos na lista do método

<https://github.com/acenelio/generics4-java>



Problema 2 (princípio *get/put*)

Vamos fazer um método que copia os elementos de uma lista para uma outra lista que pode ser mais genérica que a primeira.

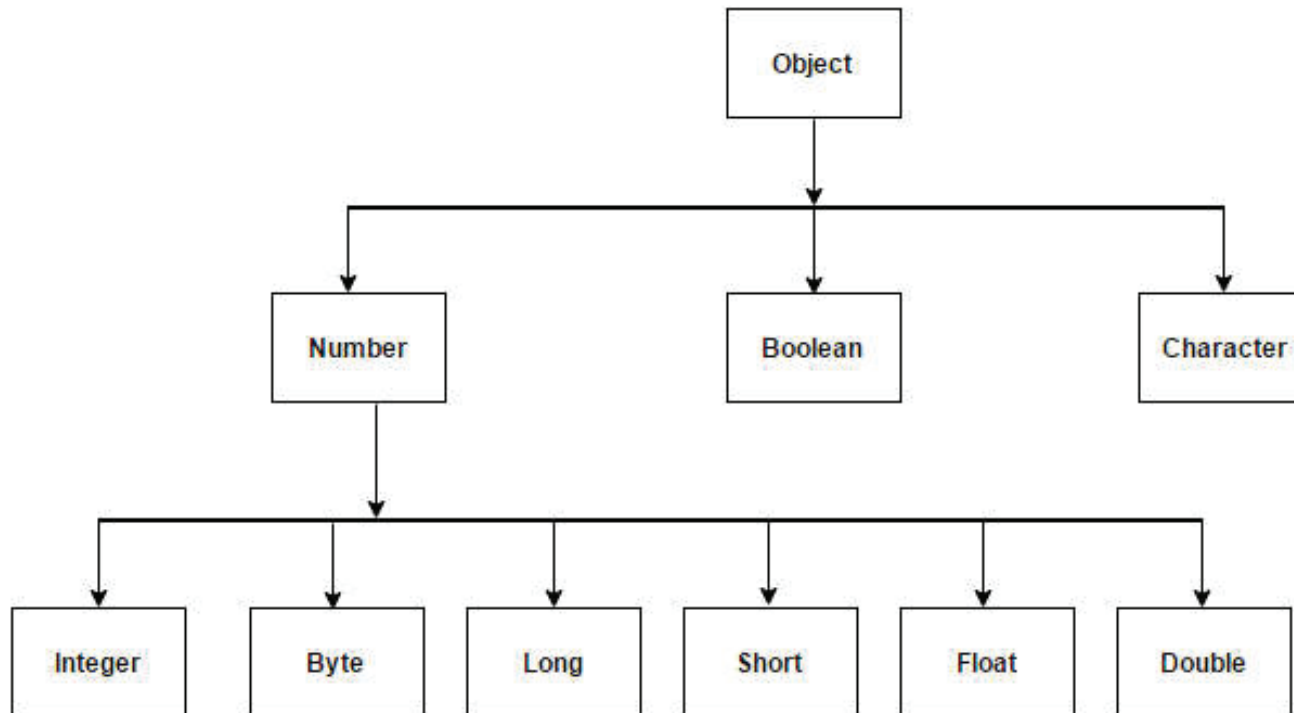
```
List<Integer> myInts = Arrays.asList(1, 2, 3, 4);  
List<Double> myDoubles = Arrays.asList(3.14, 6.28);  
List<Object> myObjs = new ArrayList<Object>();
```

```
copy(myInts, myObjs);
```

```
copy(myDoubles, myObjs);
```

<https://stackoverflow.com/questions/1368166/what-is-a-difference-between-super-e-and-extends-e>

Java wrapper types (próximos exemplos)



Princípio get/put - covariância

```
List<Integer> intList = new ArrayList<Integer>();  
intList.add(10);  
intList.add(5);
```

```
List<? extends Number> list = intList;
```

```
Number x = list.get(0);
```

```
list.add(20); // erro de compilacao
```

get - OK

put - ERROR

Princípio get/put - contravariância

```
List<Object> myObjs = new ArrayList<Object>();  
myObjs.add("Maria");  
myObjs.add("Alex");
```

```
List<? super Number> myNums = myObjs;
```

```
myNums.add(10);  
myNums.add(3.14);
```

```
Number x = myNums.get(0); // erro de compilacao
```

get - ERROR

put - OK

```
package application;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Program {

    public static void main(String[] args) {

        List<Integer> myInts = Arrays.asList(1, 2, 3, 4);
        List<Double> myDoubles = Arrays.asList(3.14, 6.28);
        List<Object> myObjs = new ArrayList<Object>();

        copy(myInts, myObjs);
        printList(myObjs);
        copy(myDoubles, myObjs);
        printList(myObjs);
    }

    public static void copy(List<? extends Number> source, List<? super Number> destiny) {
        for(Number number : source) {
            destiny.add(number);
        }
    }

    public static void printList(List<?> list) {
        for (Object obj : list) {
            System.out.print(obj + " ");
        }
        System.out.println();
    }
}
```