

# Parameters & Return Values

Bit.camp()

Predavač:  
mr. Vedad Letić

# Black Box

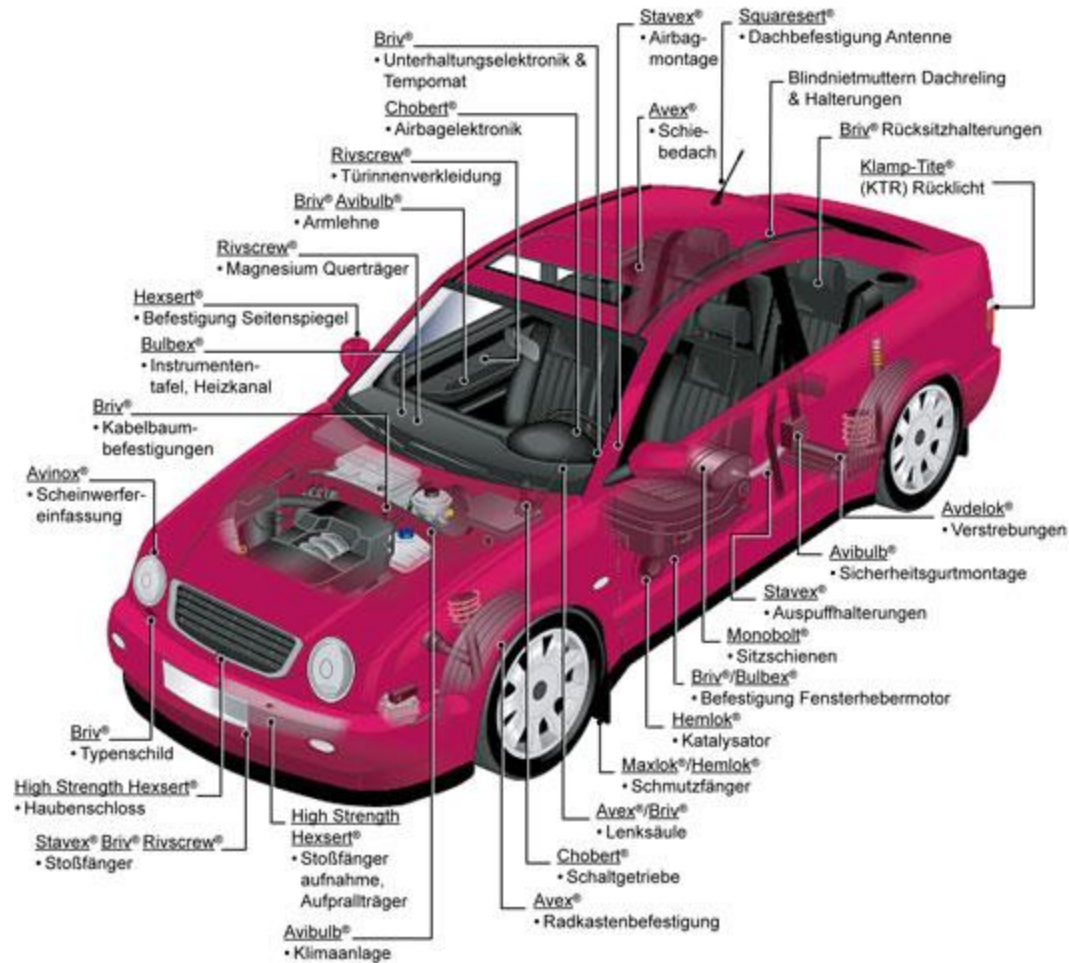
Beskorisna  
(nema interface)



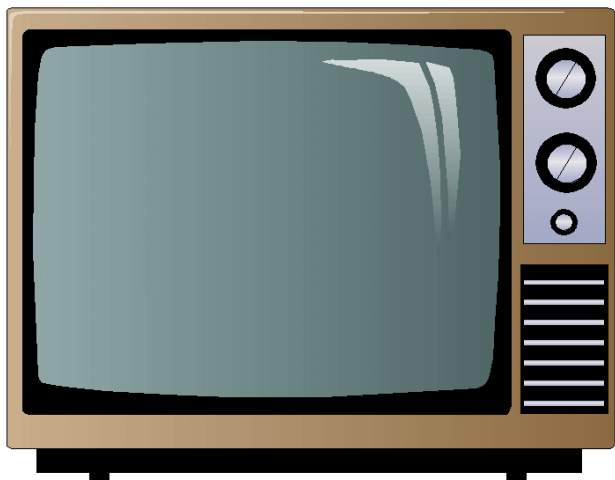
(uslovno rečeno)  
Korisna  
(ima interface)



# Black Box u stvarnom životu



# Black Box u stvarnom životu



# Black Box u stvarnom životu



# Black Box u stvarnom životu

```
System.out.printf("Evo još jedan Black Box!");
```

# Subroutines & Variables

- Deklaracija varijable:

```
<tip> <naziv> [= <vrijednost>];
```

- Deklaracija funkcije ili metode:

```
<modifikatori><tip> <naziv>(<varijable>) {  
    [blok naredbi];  
    ["ako je tip != void"]  
        return <vrijednost>;  
}
```

# Parametri

- Parametri u definiciji potprograma se zovu:  
formalni  
(engl. formal or dummy parameters)
- Parametri proslijeđeni potprogramu se zovu:  
stvarni parametri ili argumenti  
(engl. actual parameters or arguments)



# Korištenje parametara

- Možemo posmatrati problem iz poglavlja 3.2.2

```
while (N > 1) {  
    if (N % 2 == 1) // is N odd?  
        N = 3 * N + 1;  
    else  
        N = N / 2;  
    count++; // count this term  
    System.out.println(N);  
        // print this term  
}  
System.out.println("There were " + count +  
" terms in the sequence.");
```

# Overloading

## ISPRAVNO

- `println(int)`
- `println(boolean)`
- `println(char)`
- `println(String)`
- `println(double)`
- `println()`
- `println(int, char)`
- `println(int, int)`

## POGREŠNO

- `String getln()`
- `double getln()`

# Niz kao parametar (Array Parameter)

Primjer 1.

```
public static void main ( String[] args ) { ... }
```

Primjer 2.

```
static void printValuesInList( int[] list ) {  
    System.out.print('[');  
    int i;  
    for ( i = 0; i < list.length; i++ ) {  
        if ( i > 0 )  
            System.out.print(',');  
            // No comma in front of list[0]  
        System.out.print(list[i]);  
    }  
    System.out.println(']');  
}
```

Primjer poziva

```
startCounting(new int[] {1, 2, 3, 4, 5});
```

# Command Line Arguments

- Pozivanje programa iz command line sa:

```
java myProg one two three 4
```

- JVM prozove `main()` funkciju programa `myProg`
- Postavi u niz `args` vrijednosti

```
{"one", "two", "three", "4"}
```

# throw Exception

- Detaljno ćemo raditi poslije.
- Za sada ćemo to raditi samo naredbom:  
**throw new Exception("ovdje ide kod i objašnjenje");**

# Dozvoljeno, ali to ne želimo raditi

```
static Insanity Insanity( Insanity Insanity ) {  
    ...  
}
```

# Globalne i lokalne varijable

- Vidljivost varijable je samo unutar definisanog scope-a.
- Parametrima spuštamo odgovarajuće vrijednosti u potprograme.
- Varijable definisane u potprogramu ili mu predate putem parametara se nazivaju **lokalne** varijable.
- Ako potprogram koristi promjenljivu koja nije definisana unutar tijela ili mu predana kao parametar, onda se ona naziva **globalnom** za taj potprogram.

# Return Values

- Return statement:

**return** <expression>;

- Primjer:

```
static double pythagoras (double x, double y) {  
    // Computes the length of the  
    // hypotenuse of a rightangled  
    // triangle, where the sides  
    // of the triangle are x and y.  
    return Math.sqrt( x*x + y*y );  
}
```



```
public class KrajPredavanja {  
    public static void main(String[] args) {  
        System.out.println("Bili ste divna publika.");  
        System.out.println("Hvala na pažnji.");  
    }  
}
```