

# OOP

## **PRAKTIKUM 4**

Muutujad myBox1 ja myBox2 – viittüüpi (*reference type*)

```
Box myBox1;
```

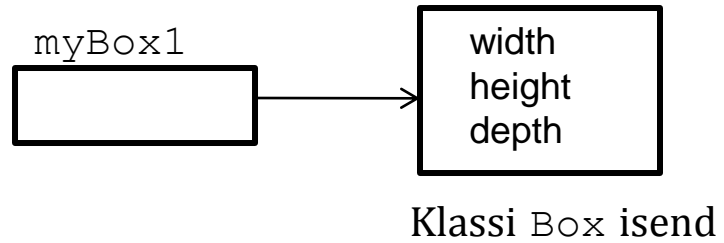
myBox1

null

```
class Box {  
    double width;  
    double height;  
    double depth;  
}
```

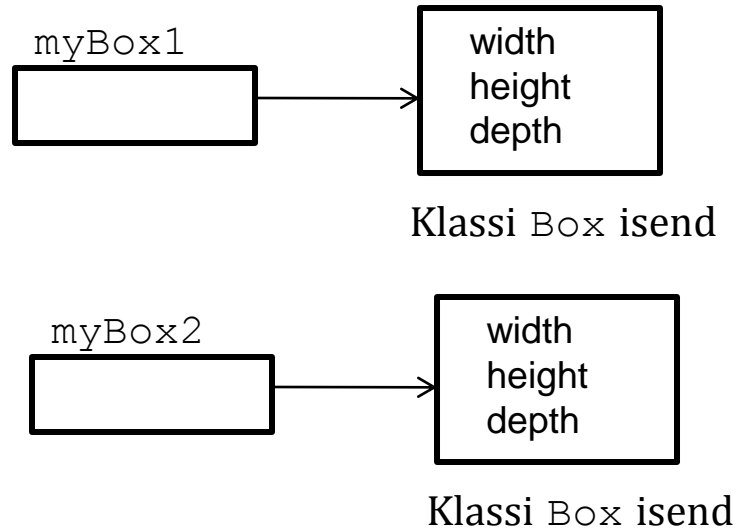
Muutujad `myBox1` ja `myBox2` – viittüüpi (*reference type*)

```
Box myBox1;  
myBox1 = new Box();
```



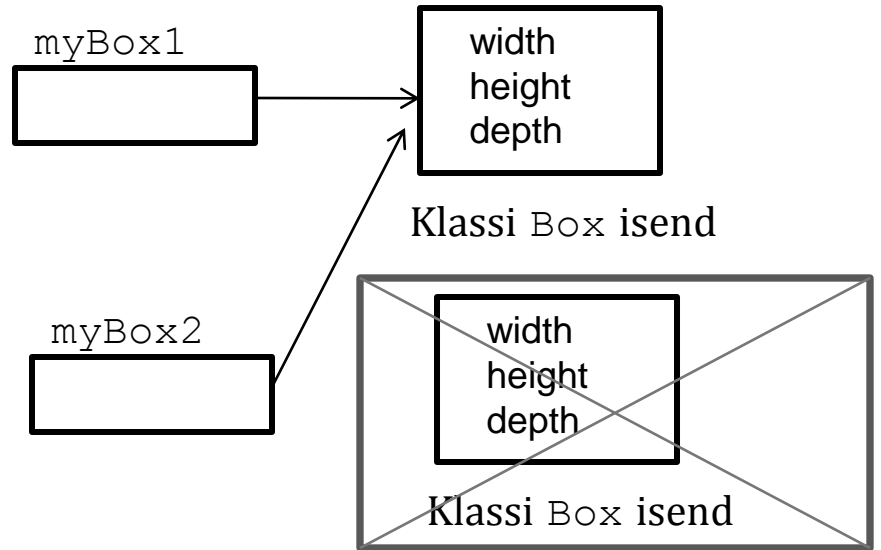
Muutujad myBox1 ja myBox2 – viittüüpi (*reference type*)

```
Box myBox1;  
myBox1 = new Box();  
Box myBox2 = new Box();
```



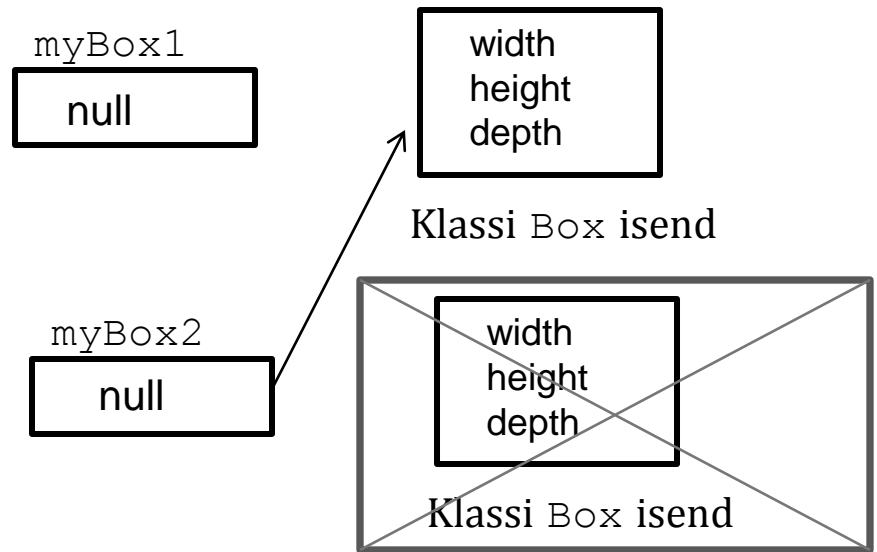
## Muutujad myBox1 ja myBox2 – viittüüpi (*reference type*)

```
Box myBox1;  
myBox1 = new Box();  
Box myBox2 = new Box();  
myBox2 = myBox1;
```



## Muutujad myBox1 ja myBox2 – viittüüpi (*reference type*)

```
Box myBox1;  
myBox1 = new Box();  
Box myBox2 = new Box();  
myBox2 = myBox1;  
myBox1 = null;
```



*Java Garbage Collection*  
annab mälu vabaks

## Meetodid Javas

### Def

```
piiritlejad tagastustüüp meetodiNimi(formaalsed parameetrid) {  
//meetodi sisu  
}
```

## Meetodid Javas

```
piiritleja tagastustüüp meetodiNimi(formaalsed parameetrid) {  
//meetodi sisu  
}
```

Näiteks:

```
public static int leiaSumma(int num1, int num2) {  
    int result = 0;  
    for (int i = num1; i <= num2; i++){  
        result += i;  
    }  
    return result;  
}
```

Diagram labels and arrows:

- Piiritlejad (points to `public static`)
- Tagastustüüp (points to `int`)
- Meetodi nimi (points to `leiaSumma`)
- Formaalsed parameetrid (points to `int num1, int num2`)
- Meetodi päis (points to `{`)
- Tagastatav väärtus (points to `return result;`)

Tagastustüüp `void` – tegemist on protseduuriga



# Meetodid Javas

```
piiritleja tagastustüüp meetodiNimi(formaalsed parameetrid) {  
//meetodi sisu  
}
```

Näiteks:

```
public static int leiaSumma(int num1, int num2) {  
    int result = 0;  
    for (int i = num1; i <= num2; i++){  
        result += i;  
    }  
    return result;  
}
```

Diagram labels and arrows:

- Piiritlejad (points to `public static`)
- Tagastustüüp (points to `int`)
- Meetodi nimi (points to `leiaSumma`)
- Formaalsed parameetrid (points to `(int num1, int num2)`)
- Meetodi päis (points to `{`)
- Tagastatav väärtus (points to `return result;`)

Meetodi **signatuur** - nimi koos parameetrite tüüpidega: `leiaSumma (int, int)`

## Meetodid Javas

```
piiritleja tagastustüüp meetodiNimi(formaalsed parameetrid) {  
//meetodi sisu  
}
```

Näiteks:

```
public static int leiaSumma(int num1, int num2) { ← Meetodi päis  
    int result = 0;  
    for (int i = num1; i <= num2; i++){  
        result += i;  
    }  
    return result; ← Tagastatav väärtus  
}
```

Diagram labels and arrows:

- Piiritlejad Tagastustüüp: points to `int`
- Meetodi nimi: points to `leiaSumma`
- Formaalsed parameetrid: points to `(int num1, int num2)`
- Meetodi päis: points to `{`
- Tagastatav väärtus: points to `return result;`

Meetodi **signatuur** - nimi koos parameetrite tüüpidega: `leiaSumma (int, int)`

Pöördumine: `int s = leiaSumma(4, 6)`

15

← Tegelikud parameetrid (argumendid)

# Muutujate ulatus (*scope*) meetodis - I

- Meetodis defineeritud muutuja on lokaalne
- Lokaalse muutuja ulatus (kättesaadavus) algab deklaratsioonist ja lõpeb teda sisaldava bloki lõpus

```
public static int leiaSumma(int num1, int num2) {  
    int result = 0;  
    for (int i = num1; i <= num2; i++){  
        result += i;  
    }  
    return result;  
}
```

result {  
    i {  
        for (int i = num1; i <= num2; i++){  
            result += i;  
        }  
    }  
}

## Muutujate ulatus (*scope*) meetodis - II

- Eri blokkides võib muutujat uuesti defineerida

```
for (int i = 1; i <= 5; i++) {  
  }  
...  
for (int i = 2; i <= 100; i = i + 2) {  
  }
```

- Väljaspool blokki ei saa blokis defineeritud muutujat kasutada

```
for (int i = 1; i <= 5; i++) {  
  }  
//System.out.println(i);
```

Viga



# Sõnetöötlus

- Sõne on Javas objekt;
- Javas põhineb sõnetöötlus sisseehitatud klasside kasutamisel

```
java.lang.String  
java.lang.Character  
java.lang.StringBuffer  
java.lang.StringBuilder
```

<http://docs.oracle.com/javase/7/docs/api/>

## Sõne loomine:

```
String s1 = new String("Hello");  
String s2 = new String("Hello");
```

## Klass **String**

### Lihtsustatud variant:

```
String t1 = "Hello";  
String t2 = "Hello";
```

Neid vaadeldakse sõnekonstandina

### Erinevus võrdlemisel:

```
System.out.println(s1 == s2);
```

false

```
System.out.println(t1 == t2);
```

true

Objekti viida võrdlemine

Puhvri sisu võrdlemine

Segaduse vältimiseks kasutada sisu võrdlemiseks alati meetodeid  
(`equals`, `compareTo`)

```
System.out.println(s1.equals(s2));
```

true

```
System.out.println(t1.equals(t2));
```

true

# Klass **String**

java.lang.String
+String() +String(value: String) +String(value: char [])
+charAt(index: int) : char +compareTo(s:String) : int +equals(s: String): boolean +indexOf(ch:int): int +length() : int +replace(oc:char, nc char):String +toLowerCase():String +toUpperCase():String +toString() : String ...

## Konstruktorid

```
char [] c1 = {'H', 'e', 'l', 'l', 'o'};  
String s3 = new String(c1);
```

## Meetodid

### Sõnede ühendamine:

```
String m1 = "Hello, " ;  
String m2 = "World!";  
m1 = m1 + m2; // "Hello, World!"
```

### Märkide arv sõnes:

```
int len = m2.length(); //6
```

### Märgi eraldamine:

```
char c2 = m1.charAt(4); // 'o'
```

"Hello, "

"Hello, World!"

# Klass **String**

## Registri vahetamine:

```
String m4 = "Hei, kus on MINU piip?";  
System.out.println(m4.toLowerCase()); // "hei, kus on minu piip?"  
System.out.println(m4.toUpperCase()); // "HEI, KUS ON MINU PIIP?"
```

## Veel võrdlemisest:

```
double tax = 0;  
if (riik.equals("Eesti") || riik.equals("eesti")) {  
    tax = 0.20;  
}
```

## Selle asemel:

```
if (riik.equalsIgnoreCase("Eesti")) { ... }
```



# Klass `String`

## Leksikograafiline võrdlemine:

Meetodi päis:

```
public int compareTo(String s)
```

```
String s1 = "kass";  
String s2 = "koer";  
int j = s1.compareTo(s2); //-14
```

- Kui sõne `s1` alfabeetiliselt eelneb sõnele `s2`, siis tagastatakse negatiivne täisarv
- Kui sõne `s1` alfabeetiliselt järgneb sõnele `s2`, siis tagastatakse positiivne täisarv
- Kui sõne `s1` alfabeetiliselt on võrdne sõnega `s2`, siis tagastatakse täisarv null

Character ASCII		Character ASCII		Character ASCII	
0	48	A	65	a	97
1	49	B	66	b	98
2	50	C	67	c	99
...	...	...	...	...	...
8	56	Y	89	y	121
9	57	Z	90	z	122

Väljavõte ASCII  
kooditabelist

## Tekstianalüüs

```
String s3 = "sometimes";
```



## Meetod

`indexOf()` – neli eri varianti (kõik tagastavad ebaõnnestumise korral -1)

1.

```
int j = s3.indexOf('e');
```

```
System.out.println(j); //3
```

2.

```
int j = s3.indexOf('e', 4);
```

```
System.out.println(j); //7
```

3.

```
int j = s3.indexOf("me");
```

```
System.out.println(j); //2
```

4.

```
int j = s3.indexOf("me", 4);
```

```
System.out.println(j); //6
```

Teine argument:

indeks, millest alustada otsimist

## Tekstianalüüs

```
String s3 = "sometimes";
```

### Meetod

`replace()` – märgi (sõne) asendamine

1.

```
String s4 = s3.replace('m', 'n');  
System.out.println(s4); // "sonetines"
```

2.

```
String s4 = s3.replace("me", "ni");  
System.out.println(s4); // "sonitinis"
```

### Meetod

`trim()` – eemaldab tühikud jt "*whitespace*" märgid mõlemast otsast

```
String s5 = " \t \n some times \r\n \t ";  
System.out.println(s5.trim()); // "some times"
```

Palju teisi meetodeid veel ...

java.lang.Character

```
+Character(value: char)
+charValue(): char
+compareTo(ch Character) : int
+equals(ch Character): boolean
+isDigit(ch: char) : boolean
+isLetter(ch: char) : boolean
+isLetterOrDigit(ch: char) : boolean
+isLowerCase(ch: char) : boolean
+isUpperCase(ch: char) : boolean
+ toLowerCase(ch: char):char
+ toUpperCase(ch: char):char
...
```

## Klass **Character**

Mähisklass (*wrapper class*) tüübile char

- Kõik peab olema objektides!

Teisi mähisklasse:

Boolean, Byte, Short, Integer, Long,  
Float, Double

Mähisklassid võimaldavad primitiivtüüpi andmeid käsitleda objektidena.

```
Character co = new Character('b');
System.out.println(co.compareTo(new Character('a'))); // 1
System.out.println(co.compareTo(new Character('b'))); // 0
System.out.println(co.compareTo(new Character('c'))); // -1
System.out.println(co.compareTo(new Character('d'))); // -2
System.out.println(co.equals(new Character('b'))); // true
System.out.println(co.equals(new Character('d'))); // false
char c1 = co.charValue(); // c1 = 'b'
```

# Klass `Character`

```
char c = 'a';  
System.out.println(Character.isLetter(c)); // true  
System.out.println(Character.isDigit(c)); // false  
System.out.println(Character.isLetterOrDigit(c)); // true
```



Need on klassimeetodid

Näide erinevate tähemärkide loendamisest sõnes

# Klass **Character**

```
import java.util.Scanner;

public class CountEachLetter {

    public static int[] countLetters(String s) {
        int[] counts = new int[26];
        for (int i = 0; i < s.length(); i++) {
            if (Character.isLetter(s.charAt(i)))
                counts[s.charAt(i) - 'a']++;
        }
        return counts;
    }
}
```

```
int[] counts = new int[26]; //nullidega täidetud täisarvujärjend
counts[s.charAt(i) - 'a']++;
```

# Klass Character

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter a string:");
    String s = sc.nextLine();
    int[] counts = countLetters(s.toLowerCase());
    String output = "";
    for (int i = 0; i < counts.length; i++) {
        if (counts[i] != 0)
            output += (char)('a' + i) + " appears " +
                counts[i] + ((counts[i] == 1) ? " time\n" : " times\n");
    }
    System.out.println(output);
} //main
} //class
```

## java.lang.StringBuilder

```
+StringBuilder()
+StringBuilder(suurus: int)
+StringBuilder(str: String)

+append(str: String): StringBuffer
+charAt(ind: int) :char
+substring(beg:int, end:int): String
+replace(o: String, n:String): StringBuffer
+delete(int start, int end): StringBuffer
+indexOf(s: String): int
+insert(i: int, s: String): StringBuffer
+capacity(): int
+length(): int
```

## Klass `StringBuilder`

- Klassi `String` kasutamine ebaefektiivne;
- Klassi `String` isendit ei saa muuta, selle asemel luuakse uus isend

- Vaikekonstruktor loob klassi `StringBuilder` isendi 16 märgi jaoks;

```
int length()
int capacity()
```

- Meetodid lisamiseks, asendamiseks, kustutamiseks
- Objekt (isend) jääb samaks!



# Klass `StringBuilder`

```
StringBuilder sb = new StringBuilder("Welcome to Java!");
```

Käsk

Isendi sb uus sisu

---

```
sb.delete(8, 11) // "Welcome Java"  
sb.deleteCharAt(8) // "Welcome o Java"  
sb.reverse() // "avaJ ot emocleW"  
sb.replace(11, 15, "HTML") // "Welcome to HTML"  
sb.setCharAt(0, 'w') // "welcome to Java"
```

# Klass `StringBuilder`

```
StringBuilder sb = new StringBuilder();  
for (int i = 1; i <= 8; i++) {  
    for (int k = 1; k <= i; k++) {  
        sb.append("X");  
    }  
    sb.append('\n');  
}  
System.out.println(sb);
```

```
X  
XX  
XXX  
XXXX  
XXXXX  
XXXXXX  
XXXXXXX  
XXXXXXXX
```

# Infovahetus tekstifailidega

Kirjutamiseks `java.io.File`  
`java.io.PrintWriter`

PrintWriter konstruktor võib tekitada erindi

```
public class WriteData{
    public static void main(String[] args) throws Exception {
        java.io.File file = new java.io.File("tulemused.txt");
        if (file.exists()){
            System.out.println("Fail on juba olemas");
            System.exit(0);
        }
        java.io.PrintWriter pw = new java.io.PrintWriter(file);
        pw.print("Charlotte Kalla SWE ");
        pw.println("24.58,4");
        pw.print("Kristina Smigun-Vahi EST ");
        pw.println("+6,6");
        pw.close();
    }
}
```

```
tulemused.txt
Charlotte Kalla SWE 24.58,4
Kristina Smigun-Vahi EST +6,6
```

# Infovahetus tekstifailidega

**Lugemiseks:** `java.util.Scanner`

```
public class ReadData {
    public static void main(String[] args) throws Exception {
        java.io.File file = new java.io.File("tulemused.txt");
        java.util.Scanner sc = new java.util.Scanner(file);
        while (sc.hasNext()) {
            String firstName = sc.next();
            String lastName = sc.next();
            String country = sc.next();
            String result = sc.next();
            System.out.println(firstName + " " + lastName +
                " " + country + " "+ result);
        }
    }
}
```

```
Charlotte Kalla SWE 24.58,4
Kristina Smigun-Vahi EST +6,6
```