

# Uebung 1 Repetition

OOP1

# Theorie

OOP1

# Welche der folgenden Aussagen sind wahr

---

## Jedes Java Programm

- braucht eine Klasse mit Namen Main
- braucht eine Methode mit Namen main.
- muss mindestens eine Variable deklarieren.
- muss eine Methode mit einem Rückgabewert haben.
- muss in einem oder mehreren Files mit der Endung .java gespeichert sein.

Ergänzen Sie den Code so dass das Wort «drin» 5 mal ausgegeben wird:

---

```
____ (int x = ____; ____ < 5; ____)  
{  
    System.out.println("drin");  
}
```

Ergänzen Sie den Code, so dass das Wort «drin» 6 mal ausgegeben wird:

---

```
int x = 1;
while (x <= __)
{
    System.out.println(____);
    __++;
}
```

Ergänzen Sie den Code so dass das Wort «wahr» ausgegeben wird, falls a grösser als 15 und b kleiner gleich 72 ist:

---

```
if (a ____ b ____ )  
{  
    System.out.println("wahr");  
}
```

# Welche Aussagen sind wahr?

---

- $a \ \&\& \ b$  ist false, falls  $a$  und  $b$  true sind.
- $(a \ || \ b) \ \&\& \ c$  ist true, falls  $c$  true ist und  $a$  oder  $b$  true ist.
- $a \ || \ b$  ist true falls  $a$  oder  $b$  true ist
- $a \ \&\& \ b$  ist true falls  $a$  oder  $b$  true ist
- $(a \ \&\& \ b) \ || \ c$  ist true, falls  $c$  true ist oder  $a$  oder  $b$  true ist.
- $a \ \&\& \ (b \ || \ c)$  ist true, falls  $b$  oder  $c$  true ist oder  $a$  true ist.

# Was ist die Ausgabe vom folgenden Code? Warum entsteht dieses Ergebnis?

---

```
int x1 = 1000000;  
System.out.println(x1 * x1);  
long x2 = 1000000;  
System.out.println(x2 * x2);  
float x3 = 1000000;  
System.out.println(x3 * x3);
```



Was ist die Ausgabe vom folgenden Code?  
Warum entsteht dieses Ergebnis?

```
double x = 0.1;
double y =
    x + x + x + x + x + x + x + x + x + x;
System.out.println(1 - y);
```

Die Methode swap wurde geschrieben, um die Werte a und b zu vertauschen. Was ist die Ausgabe beim Pfeil und warum?

---

```
public static void swap(double a, double b) {  
    double temp = a;  
    a = b;  
    b = temp;  
}
```

```
public static void main(String[] args){  
    double x = 3;  
    double y = 4;  
    swap(x, y);  
    System.out.println(x + " " + y);  
}
```



Ergänzen Sie den Code, so dass er sinnvoll und korrekt ist. Was ist dann die Ausgabe?

---

```
int arr[ ] = _____  
for (int i = 0; i < 3; i++)  
{  
    arr[i] = i;  
}  
System.out.println(arr[0] + arr[2]);
```

# Was ist die Ausgabe von diesem Code?

---

```
int result = 0;
for (int i = 0; i < 5; i++) {
    if (i == 3) {
        result += 10;
    } else {
        result += i;
    }
}
System.out.println(result);
```

Das folgende Programm gibt immer eine Zahl zwischen 20 und maximal 52 zurück. Erklären Sie das Phänomen.

---

```
public static void anzahlSchleifen() {  
    double t = Math.random();  
    int n = 0;  
    while (t != 0) {  
        if (t < 1)  
            t *= 2;  
        else  
        {  
            t -= 1;  
            n++;  
        }  
    }  
    System.out.println(n);  
}
```

# Praxis

OOP1

A1. Schreiben Sie eine Methode `min( )`, welche von drei Zahlen `a`, `b` und `c` die kleinste zurückgibt:

---

```
public static long min(long a, long b, long c)
{

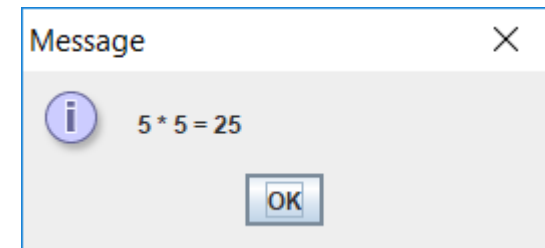
}
}
```

A2. Schreiben Sie eine Methode `square(String s)`, welche den String `s` in eine Zahl umwandelt und dann den quadrierten Wert ausgibt.

---

```
public static void square(String s)
{
```

```
}
```



Geben Sie das Resultat auf dem Bildschirm mit dem Befehl `JOptionPane.showMessageDialog(null, resultString)` aus.

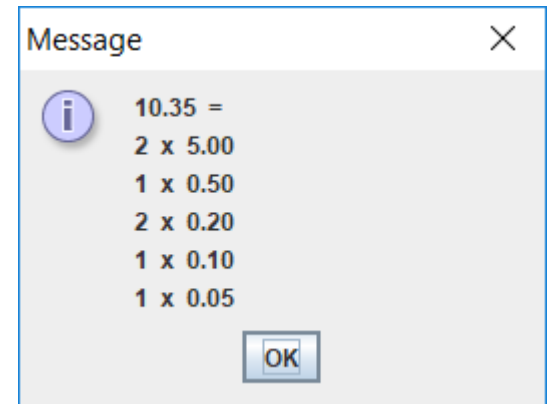


A3. Schreiben Sie eine Methode, welche für einen Geldbetrag  $d$  die optimale Zusammensetzung aus 5.-, 2., 1.-, -.50, -.20, -.10 und -.05 Münzen berechnet.

```
public static void change( double d )
```

 $\{$ 

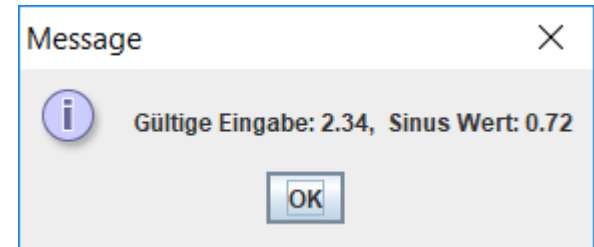
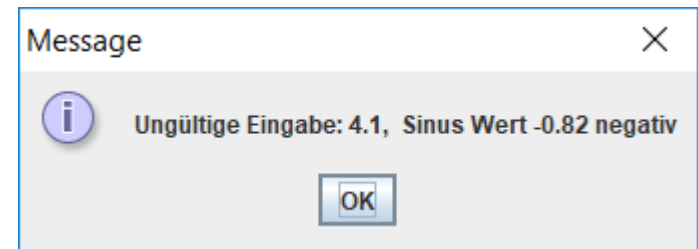
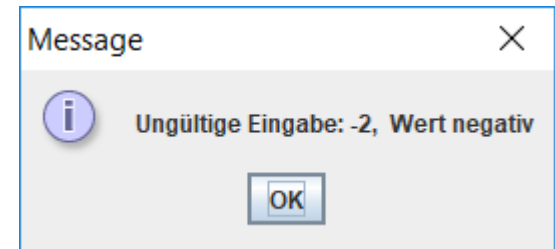
}



Verwenden Sie einen Array, um die verschiedenen Arten von Münzen zu speichern und die Methode `String.format()` um eine schöne Ausgabe zu produzieren.  
Geben Sie einen Fehlerstring aus, falls der Betrag nicht in diesen Münzen ausbezahlt werden kann.

A4. Schreiben Sie eine Methode, welche für eine Zahl deren Sinus Wert berechnet. Die Eingabe soll ungültig sein, falls die eingegebene Zahl oder der Sinus Wert davon negativ sind.

```
public static void sinValue( )
{
}
}
```



A5. Schreiben Sie eine Methode `pyramide(int n)`, welche für eine beliebige ganze Zahl  $n < 10$  die folgende Ausgabe liefert.

---

```
public static void pyramide( int n ) {
```

Pyramide bis 5

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

Pyramide bis 8

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

6 6 6 6 6 6

7 7 7 7 7 7 7

8 8 8 8 8 8 8 8

```
}
```

A6. Implementieren Sie eine Methode `findPair`, welche aus einem Array von Zahlen die ersten zwei findet, welche als Summe den vorgegebenen Wert *target* haben.

[illegible]

findPair({ 2, 3, 4, 6, 8 }, 9) soll zum Beispiel {3, 6} zurückgeben.

Das Resultat soll aus dem Array  $\{-1, -1\}$  bestehen, falls kein solches Paar existiert.

A7. Implementieren Sie eine Methode `reverse`, welche einen String `s` umdreht und am Ende des Strings `t` anhängt.

---

```
public static String reverse(String s, String t)
{

}

}
```

`reverse("abc", "xyz")` soll den String `"xyzcba"` zurück geben.