

Vorbereitung auf das Programmierdiktat

Michael Strassberger
michael.strassberger@uni-hamburg.de
3strassb@informatik.uni-hamburg.de

April 3, 2019

- 1 Syntax einer Klasse
- 2 Datenstrukturen
- 3 Kontrollstrukturen
- 4 Fragen

- Begriffe: Kopf, Rumpf

Klassen und Interfaces definieren

- Begriffe: Kopf, Rumpf

```
class Buch // KlassenKopf  
{  
    // KlassenRumpf  
}
```

Klassen und Interfaces definieren

- Begriffe: Kopf, Rumpf

```
● class Buch // KlassenKopf  
{  
    // KlassenRumpf  
}
```

```
● interface Medium // InterfaceKopf  
{  
    // InterfaceRumpf  
}
```

Implementationsbeziehungen zwischen einer Klasse und einem Interface definieren

- Deklariere die Klasse Buch welche das Interface Medium implementiert.

Implementationsbeziehungen zwischen einer Klasse und einem Interface definieren

- Deklariere die Klasse Buch welche das Interface Medium implementiert.

```
class Buch implements Medium  
{  
  
}
```

Variablen deklarieren, initialisieren, zuweisen

- Deklariere eine Variable mit dem Bezeichner "nummer" und dem primitiven Typen Integer.

Variablen deklarieren, initialisieren, zuweisen

- Deklariere eine Variable mit dem Bezeichner "nummer" und dem primitiven Typen Integer.

```
int nummer;
```

Variablen deklarieren, initialisieren, zuweisen

- Deklariere eine Variable mit dem Bezeichner "nummer" und dem primitiven Typen Integer.

```
int nummer;
```

```
int nummer = 42;
```

```
int nr;
```

```
{
```

```
    int nr = 42;
```

```
    ...  
}
```

```
...
```

Variablen deklarieren, initialisieren, zuweisen

- Deklariere eine Variable mit dem Bezeichner "nummer" und dem primitiven Typen Integer.

```
int nummer;
```

```
int nummer = 42;
```

```
    nummer = 43;
```

(Zustands-) Felder (alias Exemplarvariablen) deklarieren

primitiv

- Deklariere ein Feld vom Typ Integer mit dem Bezeichner `_seiten`

```
class Buch implements Medium  
{
```

(Zustands-) Felder (alias Exemplarvariablen) deklarieren

- Deklariere ein Feld vom Typ Integer mit dem Bezeichner `_seiten`

```
class Buch implements Medium
```

```
{  
  priv int _seiten;  
}
```

| Klassentumpf

Zugriffsmodifikatoren verwenden können (public/private)

- Deklariere ein Feld vom Typ Integer mit dem Bezeichner `_seiten` und der privaten Sichtbarkeit.

```
class Buch implements Medium  
{
```

Zugriffsmodifikatoren verwenden können (public/private)

- Deklariere ein Feld vom Typ Integer mit dem Bezeichner `_seiten` und der privaten Sichtbarkeit.

```
class Buch implements Medium
{
    private int _seiten;
}
```

Konstruktor definieren

- Schreibe einen Konstruktor, welcher das Feld `_seiten` auf den Wert 42 initialisiert.

```
class Buch implements Medium  
{  
    private int _seiten;
```

Buch()

{

_seiten=42;

}

Konstruktor definieren

- Schreibe einen Konstruktor, welcher das Feld `_seiten` auf den Wert 42 initialisiert.

```
class Buch implements Medium
{
    private int _seiten;

    public Buch()
    {
        _seiten = 42;
    }
}
```


Werte von Methoden zurückgeben lassen (Schlüsselwort "return")

- Schreibe eine sondierende Methode, welche den Wert des Feldes _seiten zurückgibt.

```
class Buch implements Medium  
{
```

```
    private int _seiten;  
    (public) int gibSeiten()  
    getSeiten()  
    {  
        return _seiten;  
    }
```

Werte von Methoden zurückgeben lassen (Schlüsselwort "return")

 Schreibe eine sondierende Methode, welche den Wert des Feldes `_seiten` zurückgibt.

```
class Buch implements Medium
{
    private int _seiten;

    public int gibSeitenAnzahl()
    {
        return _seiten;
    }
}
```

Klassenvariablen deklarieren, initialisieren

primitiven

- Deklariere ein Klassenfeld vom Typ Integer mit dem Bezeichner anzahlBuecher und der privaten Sichtbarkeit.
- Schreibe einen statischen Konstruktor, der das Klassenfeld initialisiert.

```
class Buch implements Medium
```

```
{
```

```
    private static int anzahl(Buecher;
```

```
    static
```

```
{
```

```
    anzahl(Buecher = 0;
```

```
}
```

Klassenvariablen deklarieren, initialisieren

- Deklariere ein Klassenfeld vom Typ Integer mit dem Bezeichner `anzahlBuecher` und der privaten Sichtbarkeit.
- Schreibe einen statischen Konstruktor, der das Klassenfeld initialisiert.

```
class Buch implements Medium
{

    private static int anzahlBuecher;
```

Klassenvariablen deklarieren, initialisieren

- Deklariere ein Klassenfeld vom Typ Integer mit dem Bezeichner `anzahlBuecher` und der privaten Sichtbarkeit.
- Schreibe einen statischen Konstruktor, der das Klassenfeld initialisiert.

```
class Buch implements Medium
{
    private static int anzahlBuecher;

    static
    {
        anzahlBuecher = 0;
    }
}
```

Private und öffentliche Methoden/Operationen definieren,
sowohl als Exemplar- als auch als Klassenmethoden
(Begriffe "Rumpf" und "Kopf" sollten auch bekannt sein)

mit Parameter bla vom Typ Blub

```
public int gibSeitenAnzahl()  
{  
    return _seiten;  
}
```

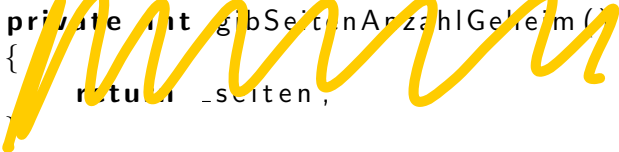
Kopf
Rumpf

Blub blub

Private und öffentliche Methoden/Operationen definieren,
sowohl als Exemplar- als auch als Klassenmethoden
(Begriffe "Rumpf" und "Kopf" sollten auch bekannt sein)

```
public int gibSeitenAnzahl()  
{  
    return _seiten;  
}
```

```
private int gibSeitenAnzahlGeheim()  
{  
    return _seiten;  
}
```



Klassenmethoden

- Schreibe eine statische sondierende Methode, die die Anzahl der Bücher zurückgibt.

```
class Buch implements Medium
```

```
{
```

```
    private static int anzahlBuecher;  
    static
```

```
{
```

```
        anzahlBuecher = 0;
```

```
}
```

```
    static int gibAnzahlBuecher()  
    {  
        return anzahlBuecher;  
    }
```

Klassenmethoden

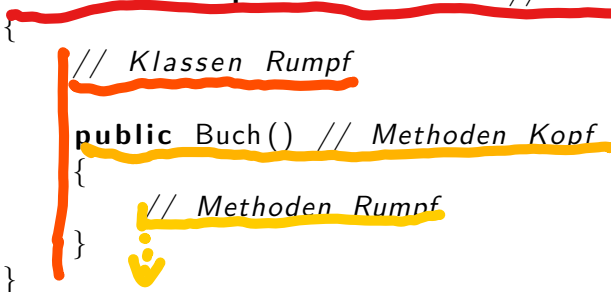
- Schreibe eine statische sondierende Methode, die die Anzahl der Bücher zurückgibt.

```
class Buch implements Medium
{
    private static int anzahlBuecher;
    static
    {
        anzahlBuecher = 0;
    }

    public static int gibAnzahlBuecher()
    {
        return anzahlBuecher;
    }
}
```

Recap: Rumpf von Klassen und Methoden

```
class Buch implements Medium // Klassen Kopf
{
    // Klassen Rumpf
    public Buch() // Methoden Kopf
    {
        // Methoden Rumpf
    }
}
```



Parameter deklarieren und verwenden

```
class Buch implements Medium
{
    private int _seiten;
```

Parameter deklarieren und verwenden

```
class Buch implements Medium
{
    private int _seiten;

    public void setzeSeitenAnzahl(int seiten)
    {
```

Parameter deklarieren und verwenden

```
class Buch implements Medium
{
    private int _seiten;

    public void setzeSeitenAnzahl(int seiten)
    {
        | _seiten = seiten;
    }
}
```

```
public class SE2Tutorium  
{  
    public
```

Main Methode

```
public class SE2Tutorium
{
    public static void main(String[] args)
    {
    }
}
```


Objekte erzeugen

```
public class SE2Tutorium
{
    public static void main(String[] args)
    {
```

interface Medium
class Buch implements Medium

```
public class SE2Tutorium
{
    public static void main(String[] args)
    {
        Medium
        Buch bluej = new Buch();
    }
}
```



Umgang mit Referenzvariablen und der Punktnotation

```
public class SE2Tutorium
{
    public static void main(String[] args)
    {
        Buch bluej = new Buch();
        bluej.gibSeiten(.....);
```

Umgang mit Referenzvariablen und der Punktnotation

```
public class SE2Tutorium
{
    public static void main(String[] args)
    {
        Buch bluej = new Buch();

        int bluejSeiten = bluej.getSeitenAnzahl();
    }
}
```

Umgang mit Referenzvariablen und der Punktnotation

```
public class SE2Tutorium
{
    public static void main(String[] args)
    {
        Buch bluej = new Buch();

        int bluejSeiten = bluej.getSeitenAnzahl();

        bluej.setzeSeitenAnzahl(bluejSeiten + 1);
    }
}
```

Noch Fragen zu Klassen und Methoden Rümpfen

Umgang mit Basisdatentypen

- (1) byte
- (2) short
- (3) int
- (4) long

Umgang mit Basisdatentypen

- (1) byte
- (2) short
- (3) int
- (4) long
- (5) float
- (6) double

- (1) byte
- (2) short
- (3) int
- (4) long
- (5) float
- (6) double
- (7) char

Byte
short
Integer

Umgang mit Arrays (deklarieren, initialisieren, schreibender und lesender Zugriff)

Umgang mit Arrays (deklarieren, initialisieren, schreibender und lesender Zugriff)

```
int [] zahlenkette
```

Umgang mit Arrays (deklarieren, initialisieren, schreibender und lesender Zugriff)

```
int[] zahlenkette
```

, Kapazität

```
zahlenkette = new int[42]
```

zahlenkette[2] = 42;

int b = zahlenkette[3];
Index

Umgang mit Listen und Mengen

Elementtyp Integer
`List<Integer> zahlenListe;`

```
List<Integer> zahlenListe;  
zahlenListe = new ArrayList();
```

<Integer> !

Umgang mit Listen und Mengen

```
List<Integer> zahlenListe;  
  
zahlenListe = new ArrayList();  
  
zahlenListe.add(100);
```


Umgang mit Listen und Mengen

```
List<Integer> zahlenListe;  
  
zahlenListe = new ArrayList();  
  
zahlenListe.add(100);  
  
zahlenListe.remove(0);
```

zahlenListe.add(100);

Switch-Anweisungen schreiben

Switch-Anweisungen schreiben

```
switch (zahl) {  
    case 10:  
        // do things  
        break;  
    default:  
        break i  
}
```

Gibt es noch Fragen?

Gibt es noch Fragen?
Dann viel Erfolg beim Diktat!