



Java für Fortgeschrittene

Teil 1



Kleiner Warm-UP

➤ <http://www.youtube.com/watch?v=RnqAXuLZlaE>





Vorausblick

- Grundlagen objektorientierter Programmierung in Java
- Praxisnaher Bezug durch das Erstellen eines Schach-Spiels
- Swing als GUI (für das Schachspiel)
- Begleitende Hausaufgabe: Programm erstellen: für Android oder Swing oder einer anderen Plattform
- Anbindung an eine MySQL Datenbank



Verwendete Programme



- Eclipse:

<http://www.eclipse.org/downloads/>

- Swing-Editor (Plugin für Eclipse):

<https://www.eclipse.org/windowbuilder/>

- Subversion / TortoiseSVN:

<http://tortoisesvn.net/downloads.html>

- NotePad++ (n++)

http://www.chip.de/downloads/Notepad_12996935.html



Eclipse



- **Anlegen eines neuen Projektes:** Package Explorer -> Context-Menu (rechts-Klick) -> New -> Java Project
- **Plug-Ins installieren:** Menu -> Help -> Install New Software...
- **Weitere Fenster einblenden:** Menu -> Window -> Show View -> ...
- **Wichtige Short-Cuts:**
 - Strg + F = in Datei Suchen
 - Strg + H = in Projekt oder in Workbench suchen
 - Strg + Shift + / = Aus- und Einkommentieren von markierten Zeilen
 - Strg + Leerzeichen = Liefert Vorschläge im Editor



Wichtige Links



- Gutes Online-Java-Buch – Java ist auch eine Insel:
<http://openbook.galileocomputing.de/javainsel9/>
- Forum für Fragen zum Thema Programmierung:
<http://stackoverflow.com/>
- Bibliothek für Open-Source Bilder:
<http://www.clker.com/>
- Java Code-Snipptes:
<http://www.java2s.com/Code/Java/CatalogJava.htm>

Primitive Datentypen

Datentyp	Größe¹	Wrapper-Klasse	Wertebereich	Beschreibung
boolean	JVM-Spezifisch	java.lang.Boolean	true / false	Boolescher Wahrheitswert
char	16 bit	java.lang.Character	0 ... 65.535 (z. B. 'A')	Unicode-Zeichen (UTF-16)
byte	8 bit	java.lang.Byte	-128 ... 127	Zweierkomplement-Wert
short	16 bit	java.lang.Short	-32.768 ... 32.767	Zweierkomplement-Wert
int	32 bit	java.lang.Integer	-2.147.483.648 ... 2.147.483.647	Zweierkomplement-Wert
long	64 bit	java.lang.Long	-9.223.372.036.854.775.808 ... 9.223.372.036.854.775.807	Zweierkomplement-Wert
float	32 bit	java.lang.Float	+/-1,4E-45 ... +/-3,4E+38	Gleitkommazahl (IEEE 754)
double	64 bit	java.lang.Double	+/-4,9E-324 ... +/-1,7E+308	Gleitkommazahl doppelter Genauigkeit (IEEE 754)

Quelle: http://de.wikibooks.org/wiki/Java_Standard:_Primitive_Datentypen

Anweisungen

If-Else-Anweisung:

```
if(i < 8){  
...  
}  
else if(i > 12){  
...  
}  
else{  
...  
}
```

For-Schleife:

```
for(int i = 0; i < 3; i++){  
...  
}
```

While Schleife:

```
while(i < 6){  
...  
}
```

```
do{  
...  
}while(x > 6);
```

Switch-Case-Anweisung:

```
switch (status) {  
case NEXTTURN_WHITE:  
...  
break;  
case NEXTTURN_BLACK:  
...  
break;  
case WHITE_WINS:  
...  
break;  
case BLACK_WINS:  
...  
break;  
default:  
...  
break;  
}
```

Break: Beendet die Schleife

Continue: Beendet den aktuellen Schleifendurchlauf und beginnt den Nächsten

Enums und Arrays

- Erstellung eines Enums:

- `public enum Status {NEXTTURN_WHITE, NEXTTURN_BLACK, WHITE_WINS, BLACK_WINS, NO_GAME}`

- Verwendung eines Enums:

- `Status status = Status.BLACK_WINS;`

http://openbook.galileocomputing.de/javainsel/javainsel_05_004.html#dodtp4a00920c-2407-4586-a0c5-c21d67dc887f

- Arrays:

```
int[] array = new int[10];
```

```
array[0] = 1;
```

```
int[][] chessboard = new int[8][8];
```

```
chessboard[0] = 1;
```

http://openbook.galileocomputing.de/javainsel/javainsel_03_008.html#dodtp9894e88c-16e7-4052-8cc7-1f947e09e784



Packages

Java Quell-Code wird in Packages organisiert. Jedes File wird einen bestimmten Package hinzugefügt. Darum beginnt jedes Java-File mit dem Signal-Wort Package: `package <Name des Package>`

Packages können auch ineinander verschachtelt werden:
`<Package-Name>.<untergeordnetes Package>`

Packages können mit dem Signalwort Import in ein File eingebunden werden.
`Import <Package-Name>(. <Package-Name>).<Klassen-Namen>`

Klassen

```
public abstract class Figure {  
  
    protected Player player;  
    ...  
  
    public Figure(Player player, Position position, IChessboard chessboard){  
        this.player = player;  
        ...  
    }  
  
    protected void setImage(String whiteImage, String blackImage){  
        ...  
    }  
  
    public void setReachablePositions(){  
        //get Overridden in childclass  
    }  
}
```

http://openbook.galileocomputing.de/javainsel/javainsel_03_001.html#dodtp394ff2f8-9134-4eb3-a6b4-469ed4572a74

Instanz einer Klasse

- Eine Instanz:


```
Pawn pawn = new Pawn(Player.WHITE, new Position(x, 1), this);
```

- Aufruf einer Methode:

```
pawn.getReachablePositions();
```

- Aufruf einer Variable:

```
pawn.variable;
```



Die Signalwörter private, protected und public

- **Private:** Methoden und Variablen die mit Private deklarierte wurden sind nur in der Klasse verwendbar/aufrufbar, in der sie deklariert wurden.
- **Protected:** Im Gegensatz zu privaten Methoden und Variablen können die Erben (Kinderklassen) die Parameter verwenden, nicht jedoch externe Klassen.
- **Public:** In diesen Fall können sowohl Erben als auch externe Klassen, und Methoden und Variablen verwenden.

Statische Methoden und Variablen

- Variablen und Methoden die mit Hilfe von „static“ deklarierten wurden beziehen sich auf die Klasse und nicht auf die Instanz einer Klasse. Darum werden statische Variablen bereits beim Ausführen des Programms im Speicher angelegt. In Statischen Methoden können daher keine dynamischen Variablen verwendet werden da diese sich auf das Objekt beziehen.
- Der Aufruf einer statischen Methode lautet wie folgt:
 <Klassen-Name>.<Methoden-Name>
- Der Aufruf einer statischen variable lautet wie folgt:
 <Klassen-Name>.<Variablen-Name>

Das Schlüsselwort final

- Finale Variablen die mit final deklariert wurden können nur ein einziges mal beschrieben werden, danach können Sie nur gelesen werden.
- Bei einer Deklaration einer Variable mit „static“ und mit final handelt es sich um eine Konstante.
- Finale Methoden können von Erben nicht überschrieben werden.

http://openbook.galileocomputing.de/javainsel/javainsel_02_007.html#dodtp81f49256-ccc6-4e96-9a2c-166ba4d2d711

http://openbook.galileocomputing.de/javainsel/javainsel_05_010.html#dodtpcd7c734d-1ac9-4bc6-8a0a-eab14c6606af



Singleton

Bei einem Singleton handelt es sich um ein Objekt, das im Programm nur einmal vorkommen darf. Am Beispiel eines Schachspiels ist das Schachbrett ein Singleton.

Aufgabe:

Erstellen Sie ein Singleton!

Tipp: Verwenden Sie eine statische Methode, die die Instanz der Singleton-Klasse zurück gibt. Bedenken Sie, dass es sich hierbei immer um die selbe Klasse handeln muss.

Vererbung

```
public class Pawn extends Figure { // Bauer

    public Pawn(Player player, Position position, IChessboard chessboard) {

        public Pawn(Player player, Position position, IChessboard chessboard) {
            super(player, position, chessboard); // Aufruf der Basis-Klassen-Konstruktor
            ...
        }

        @Override
        public void setReachablePositions() {
            ...
        }

        private void add(Chessfield field) {
            ...
        }
    }
}
```

Schlüsselwörter super und this:

Super: Verwendung der Basis-Klasse

This: Verwendung der eigenen Klasse (kann auch weg gelassen werden)



Hausaufgabe:

- Überlegung welches eigenes Programm man machen möchte
 - Architektur des Schachspieles entwerfen
 - Entwurf der Figuren
- 