

Hochschule Harz	FB Automatisierung und Informatik
Design Pattern	Dipl.-Inf., Dipl.-Ing. (FH) M. Wilhelm
	Thema: Abstrakte Fabrik

Versuchsziele

Kenntnisse in der Anwendung von:

- Abstrakten Fabriken

Abstrakte Fabrik:

In dieser Aufgabe soll mit Hilfe einer abstrakten Fabrik ein Rollenspiel mit drei verschiedenen Welten programmiert werden:

- | | | | |
|--------------|-----------|------------------------|-----------------------------------------------------------|
| • Robin Hood | gegen den | Sheriff von Nottingham | im Wald (Sherwood Forest) |
| • Skywalker | | Darth Vader | im Weltall (Stern Galaktika) |
| • Messi | | Ronaldo | in Spanien auf einem Fußballfeld
Madrid oder Barcelona |

Interfaces:

- IPerson
- IWaffe
- IOrt

Abstrakter Generator:

- IPerson getPerson1()
- IPerson getPerson2()
- IWaffe getWaffe1()
- IWaffe getWaffe2()
- IOrt getOrt()

Aufgaben

1. Teilaufgabe: Projekt erstellen und aufbauen:

- Projektname: JediRobinHoodFussball

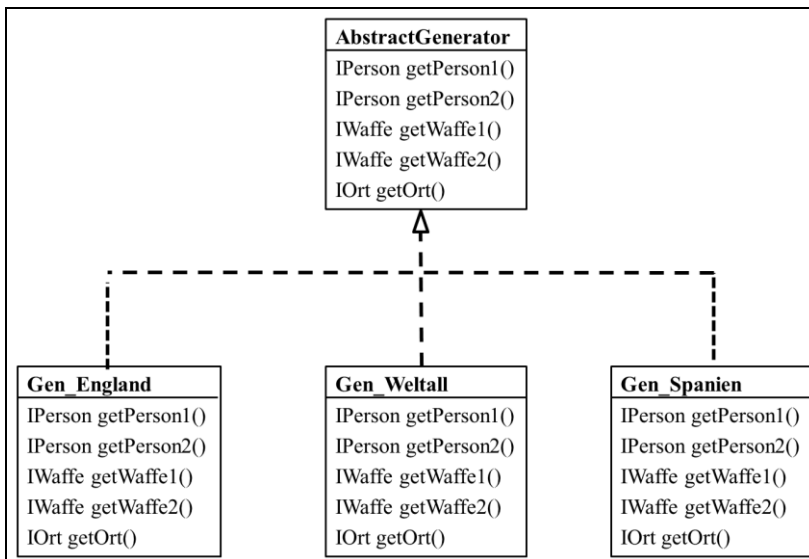


Abbildung 1 Überblick der GUI: JediRobinHoodFussball

- Im Projekt soll der vorgegeben Quellcode eingebaut werden:
 - Homepage,
 - Name: JediRobinHoodFussball.txt.
 - Name: Arena.txt

2. Teilaufgabe: Interfaces erstellen

In dieser Aufgabe sollen die Schnittstellen und der abstrakte Generatoren implementiert werden:



- Erstellen der Schnittstellen:
 - IPerson
 - String getName();
 - IWaffe
 - String getName();
 - int kaempfe(IWaffe waffe); // ruft getValue auf, 2x
 - int getValue(); // gibt den aktuellen „Waffenwert“ zurück
 - IOrt
 - String getName();
 - Abstrakter Generator:
 - Name: Generator
 - Methoden:
 - getPerson1();
 - getPerson2();
 - getWaffe1();
 - getWaffe2();
 - getOrt();

3. Teilaufgabe: Reale Klasse für die Interfaces erstellen

Diese Klassen sind nun wesentlich komplexer:

Hinweis:

- **Implementieren Sie erst mal EINE Variante.**

- **Spiel England:**

- Person1/2:
 - Klasse Pers_England
 - (Unterschied per Parameter)
 - Attribut: String name
- Waffe:
 - Klasse PfeilBogen
 - Hat keine Attribute, muss aber trotzdem als Namen „PfeilBogen“ zurückgeben.
- Waffe:
 - Klasse Schwert_Schild
 - Hat keine Attribute, muss aber trotzdem als Namen „Schwert_Schild“ zurückgeben.
- Ort:
 - Klasse Wald
 - Attribut: String Name
 - Wert: Sherwood Forest

- **Spiel Weltall:**

- Person1/2:
 - Klasse Pers_Weltall
 - (Unterschied per Parameter)
 - Attribut: String name
- Waffe:
 - Klasse Laser
- Ort:
 - Klasse Weltall
 - Attribut: String Name
 - Wert: „Stern Galaktita“

- **Spiel Spanien**

- Person1/2:
 - Klasse Pers_Spanien
 - (Unterschied per Parameter)
 - Attribut: String name
- Waffe:
 - Klasse Ball
 - Hat keine Attribute, muss aber trotzdem als Namen „Fußball“ zurückgeben.
- Ort:
 - Klasse Rasen
 - Attribut: String Name
 - Wert: Madrid oder Barcelona (mit Math.random())

4. Teilaufgabe: Reale Generatoren erstellen

In dieser Aufgabe sollen die realen Generatoren implementiert werden:

- realer Generator:
 - Name: „Gen_England“
 - Methoden (die Rückgabetypen wurde leider gelöscht):
 - `getPerson1();` Pers_England (Robin Hood)
 - `getPerson2();` Pers_England (Sheriff von Nottingham)
 - `getWaffe1();` PfeilBogen
 - `getWaffe2();` Schwert_Schild
 - `getOrt();` Wald (Sherwood Forest)

- realer Generator:
 - Name: „Gen_Starwars“
 - Methoden:
 - `getPerson1();` Pers_Weltall Skywalker
 - `getPerson2();` Pers_Weltall Darth Vader
 - `getWaffe1();` Laser
 - `getWaffe2();` Laser
 - `getOrt();` Weltall („Stern Galaktita“)

- realer Generator:
 - Name: „Gen_England“
 - Methoden:
 - `getPerson1();` Pers_Spanien Messi
 - `getPerson2();` Pers_Spanien Ronaldo
 - `getWaffe1();` Ball
 - Name: Fußball
 - `getWaffe2();` Ball
 - Name: Fußball
 - `getOrt();` Rasen
 - mit Zufallsgenerator entweder Barcelona oder Madrid.

5. Teilaufgabe: Kampf-Methode implementieren:

Um einen „echten“ Kampf zu simulieren, muss man zwei Waffen verknüpfen.

- In der Klasse „Arena“ werden neben anderen Objekten zwei Waffen übergeben.
- **Jede Waffe hat folgende Methoden:**
 - `int kaempfe(IWaffe waffe);`
 - `int getValue();`
- **Methode getValue:**
 - Hier wird mit einer aktuelle Zufallszahl der „Kampfwert“ ermittelt.
 - Der Kampfwert sollte zwischen 0 und 100 liegen
 - `return (int) (Math.random()*2);` // Nur ein Beispielcode
- **Methode kaempfe:**
 - Diese Methode erhält als Parameter die gegnerische Waffe.
 - Es wird die eigene „getValue“- und die gegnerische getValue aufgerufen.
 - Aus diesen beiden Werten wird die Differenz gebildet.
 - Ist die Differenz größer als 60, hat Person1 gewonnen. Dann wird eine +1 zurückgegeben.
 - Ist die Differenz kleiner als -60, hat Person2 gewonnen. Dann wird eine -1 zurückgegeben.
 - Ansonsten wird eine Null zu zurückgegeben.

6. Teilaufgabe: Reale Klasse „Arena“ erstellen

In dieser Klasse sollen das „Spiel“ verknüpft und gestartet werden:

- Einbau der Klasse „Arena“ von meiner Homepage.
- Bitte die Arena-Klasse für die Hausaufgabe benutzen.
- Verknüpfen der drei Events in der Klasse „Aufgabe10.java"
 - Erzeugen der realen Generatoren
 - Weiterleiten an die Arena
- Klasse Arena:
 - In einem Timer-Event „kaempfe_click“ wird die Methode „kaempfe“ der ersten Person aufgerufen. Diese Methode erhält aber als Parameter die Waffe der zweiten Person.
 - Der Rückgabewert „w1.kaempfe(w2)“ wird nun ausgewertet.
 - Wenn der Wert ungleich Null, stoppt der Timer und der Gewinner wird ausgegeben.
 - Der „Kampfwert sollte aber immer ausgegeben werden.

Testbilder:

