

## Übungsblatt 2

### Aufgabe 5 Eine Waffenfabrik errichten

Ein Spiel soll durch die Möglichkeiten, Charaktere mit Waffen auszustatten, erweitert werden. Dazu wird eine abstrakte Oberklasse *Weapon* erstellt.

- Als Eigenschaften sind *Cooldown* und *Ammo* enthalten.
- Es sind abstrakte Operationen *Fire()* und *Update()* beinhaltet.

Zwei Klassen erben direkt von *Weapon*: *LaserWeapon* und *ParticleWeapon*. Um das Spiel flexibel zu halten, wird eine Klasse *WeaponFactory* angelegt:

- Die Klasse ist nur als Singleton ansprechbar.
- Sie repräsentiert eine Fabrik nach dem Factory Pattern.
- Sie besitzt eine Methode *Produce()* zum Erstellen neuer Waffen.
- Bauen Sie eine Möglichkeit ein um neue Waffen zu registrieren.

Überlegen Sie sich, welche Parameter die Methode *Produce()* erwartet. Der fertige Bauplan soll in einem UML Klassendiagramm dargestellt werden.

### Aufgabe 6 Die Waffenfabrik implementieren

Implementieren Sie den Entwurf aus Aufgabe 5 in der Programmiersprache Ihrer Wahl (Java, C# oder C++). Gehen Sie dabei so vor:

1. Implementieren Sie das Klassendiagramm.
2. Schreiben Sie ein Beispielprogramm, das die Fabrik anspricht.
3. Testen Sie verschiedene Parameter, um sicherzustellen, dass die Fabrik funktioniert.

Zeichnen Sie abschließend ein Sequenzdiagramm, das den Aufruf der *Produce()* Methode widerspiegelt.

### Aufgabe 7 Charaktere bauen

In dem Spiel sollen auch Charaktere vorkommen, welche aus verschiedenen Teilen bestehen. Solche Teile können z.B. durch

- eine Waffe,
- eine Spezialfähigkeit,

- eine Rüstung,
- ...

gegeben sein.

Implementieren Sie das Builder Pattern um solche Charaktere zu erschaffen. Für die Charaktere gilt:

- Erstellen Sie eine abstrakte Basisklasse *Character*.
- Geben Sie dieser sinnvolle Eigenschaften (z.B. eine für *Weapon*).
- Implementieren Sie zwei bis drei konkrete Spezialisierungen der Basisklasse.

Integrieren Sie die Fabrik aus Aufgabe 5 ins Builder Pattern, um den Charakteren Waffen zu geben. Wie kann diese Integration aussehen?

**! Wichtig**

Alle Diagramme können über Anwendungen (z.B. PlantUML, yUML, Visual Studio, ...) oder auch per Hand gezeichnet werden.