Übungsblatt 1

Einführung Java & Objektorientierung

In dieser Übung sollen die grundlegenden Konzepte der Objektorientierung angewendet werden. Darüber hinaus werden die bereits aus GPT bekannten Konzepte wiederholt und im Rahmen eines kleinen Projektes umgesetzt. Das Ziel dieser Übung ist, dass ein kleines textbasiertes Spiel programmiert wird, in dem rundenbasierte Kämpfe im Stil von Final Fanatsy ausgetragen werden können. Es wird immer nur gegen ein einzelnes Monster gekämpft. Ein Monster besitzt Lebenspunkte und hat einen minimalen und einen maximalen Schaden. Der Schaden, den das Monster austeilt, ist immer zwischen diesen beiden Werten. Der Spieler besitzt Lebenspunkte und Mana. Er kann sowohl Angriffe mit Waffen als auch mit Magischen Sprüchen ausführen, wobei die Sprüche eine bestimmte Menge an Mana verbrauchen. Die verschiedenen Angriffe sollen im Quelltext einfach erstellt und modifiziert werden können.

Möglicher Spielablauf

```
Welcome to the Turn-Based-Battle-System App
Do you wanna fight? (y/n)
y
############# New Round #############
Your Life: 100
Your Mana: 35
Monster Life: 100
_____________________________________
It is your turn. Which attack do you want to perform?
Melee Attack with Sword. Deals 7 to 9 Damage (1)
Magic Spell called Fire Ball. Costs 10 Mana and deals 1 to 18 Damage (2)
1
Monster got 7 Damage.
Monster Attacks!
You  got 9 Damage.
############# New Round #############
Your Life: 91
Your Mana: 35
Monster Life: 93
_____________________________________
It is your turn. Which attack do you want to perform?
Melee Attack with Sword. Deals 7 to 9 Damage (1)
Magic Spell called Fire Ball. Costs 10 Mana and deals 1 to 18 Damage (2)
2
Monster got 8 Damage.
Monster Attacks!
You  got 8 Damage.
############# New Round #############
Your Life: 83
Your Mana: 25
Monster Life: 85
_____________________________________
It is your turn. Which attack do you want to perform?
Melee Attack with Sword. Deals 7 to 9 Damage (1)
Magic Spell called Fire Ball. Costs 10 Mana and deals 1 to 18 Damage (2)

[...]
```

Lösungshinweise

Die Lösungshinweise sind nur als mögliche Ansätze gedacht und andere Herangehensweisen sind ebenso erwünscht. In der Programmierung gibt es nie nur eine Lösung!

Tipp 1

Verwende die Klasse **Java.util.Scanner** aus der Standardbibliothek. Diese bietet unter anderem die Methoden **nextInt()** und **nextLine()** an, die die Verarbeitung der Benutzereingaben vereinfacht.

Tipp 2

Verwende die Methode **Math.random()** aus der Standardbibliothek, um die zufälligen Schadenswerte zu berechnen.

Tipp 3

Es könnte folgende Klassen oder Interfaces geben: **Player, Monster, Attack, WeaponAttack, MagicAttack, BattleArena ...**

Tipp 4

Die **BattleArena**-Klasse ist für das Austragen der Kämpfe verantwortlich.

```
public class BattleArena {
  public void fight(Player player, Monster monster) {
    [...]
    boolean done = false;
    while (!done) {
      [...]
      if (monster.getLife() <= 0) {
        System.out.println("You won!");
        return;
      }
      if (player.getLife() <= 0) {
        System.out.println("You lost!");
        return;
      }
      [...]
    }
  }
}
```

Tipp 5

Es könnte ein Interface **Attack** geben, das mit den Klassen **WeaponAttack** und **MagicAttack** verwendet wird.

```
public interface Attack {
  String getDescription();
  int getDamage();
}
```

Tipp 6

Die Erstellung eines Spielers könnte folgendermaßen aussehen:

```
var player = new Player();
player.setLife(100);
player.setMana(35);
player.addAttack(new WeaponAttack("Sword", 7, 9));
var spellAtack = new SpellAttack();
spellAtack.setSpellName("Fire Ball");
spellAtack.setMinDamage(1);
spellAtack.setMaxDamage(18);
spellAtack.setManaCost(10);
player.addAttack(spellAtack);
```

Zusätzliche Aufgaben

Um eure Fähigkeiten noch weiter zu verbessern, kann das Spiel um folgende Features erweitert werden:

- Kämpfe gegen mehrere Monster sind möglich.
- Die Monster können auch aus verschiedenen Angriffen wählen.
- Es gibt ein Belohnungssystem mit Erfahrungspunkten.
- Verändere das Programm derart, sodass Fließkommawerte für die Lebenspunkte, Schaden und mana verwendet werden. Die Ausgabe soll dabei aber auf eine Nachkommastelle begrenzt sein.
- Es gibt ein Itemsystem (z.B. Heil- oder Manatränke)