

Pair Programming Protokoll Aufgabe 2.1

Driver: Igor Dimitrov

Navigator: Malte Herzog

1. Ziel: Kodierung der fundamentalen Komponenten der Klasse Trainer

Die fundamentalen Member-variablen der Klasse Trainer, sowie der Konstruktor wurden deklariert/definiert:

```
private String firstName;  
private String lastName;  
  
Trainer(String first, String last) {  
    this.firstName = first;  
    this.lastName = last;  
}
```

Driver: Malte Herzog

Navigator: Igor Dimitrov

2. Ziel: Definition von toString() um Trainer in der Konsole anzuzeigen

Die toString() methode wurde wie folgend definiert:

```
public String toString() {  
    return "Name: " + firstName + " " + lastName;  
}
```

Driver: Malte Herzog

Navigator: Igor Dimitrov

3. Ziel: Linking eines Pokemons zu einem Trainer

Da, ein Trainer beliebig viele Pokemons besitzen kann wurde die Klasse **Trainer** zuerst um eine Attribute pokemons des Typs ArrayList<Pokemon> erweitert. Hingegen kann ein Pokemon hoechstens zu einem Trainer gehoeren. Um diese Multiplizitaet-bedingung zu realisieren wurde die Klasse Pokemon um eine Member-variable erweitert, die eine Referenz zu einem Trainer ist.

Diese hat den Default-wert von `null`. D.h. ein frisch erzeugtes Pokemon hat zuerst keinen Trainer:

```
private Trainer trainer = null;
```

Die eigentliche Linkung eines Pokemons zu einem Trainer wurde durch eine Methode `addPokemon(Pokemon p)` der Klasse `Trainer` realisiert. Diese Methode greift auf der `add()` Methode der Liste `pokemons` zu, fuegt zu der Liste den als Parameter uebergebene-n Pokemon hinzu und setzt den Trainer dieses Pokemons als dieses Trainer ueber der `setTrainer(Trainer t)` Methode der Klasse `Pokemon`:

```
public void addPokemon(Pokemon p) {  
    this.pokemons.add(p);  
    p.setTrainer(this);  
}
```

Um die Multiplizitaetsbedingungen zu erhalten wurde diese Methode um eine `if`-Bedingung erweitert, die sicherstellt, dass ein Pokemon, der bereits zu einem Trainer gehoert nicht erneut addiert werden kann:

```
public void addPokemon(Pokemon p) {  
    if (p.trainer == null) {  
        this.pokemons.add(p);  
        p.setTrainer(this);  
    }  
    else {  
        System.out.println("Pokemon wurde bereits einem Trainer zugewiesen");  
    }  
}
```

Driver: Igor Dimitrov

Navigator: Malte Herzog

4. Ziel: Anzeigen von Pokemons eines Trainers im Terminal

Die Pokemons werden anhand einer `for`-Schleife auf der Konsole ausgegeben, die ueber der Elemente der Liste `pokemons` Iteriert. Um die Lesbarkeit zu verbessern wurde die eine `for-each` Schleife bevorzugt, statt eine einfache Schleife.

```
public void showPokemons() {  
    for (Pokemon p : this.pokemons) {  
        System.out.println(p);  
    }  
}
```

Driver: Igor Dimitrov

Navigator: Malte Herzog

5. Ziel: Anzeigen von Pokemons bestimmtes Typs eines Trainers im Terminal

die `showPokemons()` Methode wurde einfach um eine `if`-Bedingung erweitert, die bei der Ausgabe den als Parameter uebergebenen Typ ausfiltert:

```
public void showPokemonByType(Type type) {  
    for (Pokemon p : this.pokemons) {  
        if (p.getType() == type) {  
            System.out.println(p);  
        }  
    }  
}
```

Driver: Malte Herzog

Navigator: Igor Dimitrov

6. Ziel: Anzeigen von i-ten Pokemon von Trainer in Terminal

Diese Methode greift auf der `get()` Methode der Liste Pokemons zu und gibt das Pokemon auf der Konsole aus:

```
public void showIthPokemon(int i) {  
    System.out.println(pokemons.get(i));  
}
```