Aufgabenblatt #5

Lösen Sie die hier gestellten Aufgaben und kommen Sie mit diesen Lösungen in die Übungsgruppe, in die Sie sich angemeldet haben. Bereits **am Tag zuvor** müssen Sie allerdings bereits in TUWEL angegeben haben, welche Aufgaben Sie tatsächlich gelöst haben. Geben Sie nur die Beispiele (=Aufgaben) als gelöst an, welche vollständig und wie verlangt durchgeführt wurden.

Wenn Sie es nicht bereits gemacht haben, lesen Sie bitte die relevanten Teile des Skriptums und erarbeiten Sie gemeinsam die Fragen am Ende des dritten Kapitels.

Alles bis inklusive drittes Kapitel.

Ladybird

In den Programmierbeispielen befassen Sie sich mit Marienkäfern. Coccinellidae (engl. Ladybird, deutsch Marienkäfer) ist eine bekannte Familie von Käfern. Aber zuvor einige **allgemeine Hinweise** zum Programmieren:

- Vermeiden Sie Getter- und Setter-Methoden wenn möglich.
- Achten Sie auf korrekte Sichtbarkeit.
- Für alle Klassen ist immer `toString` und `equals` zu implementieren.
- Vermeiden Sie bitte langen und unnötigen Programmcode. Bedenken Sie, dass der Code auf die Tafel geschrieben werden muss.
- Lesen Sie sorgfältig den Eintrag zu `equals` in der Java-API und behandeln Sie die dort angeführten Sonderfälle (`null`) wie dort angegeben.

Die Methode darf *keine* NullPointerException oder ClassCastException werfen, unabhängig davon, welcher Parameter übergeben wird.

Listing 1: Marienkäfer

```
public class Ladybird {
    private int weight; // Gewicht in Gramm
    private int dots; // Anzahl der Punkte

    // Erhöhe das Gewicht wenn das Essen
    // essbar ist (bestimmbar über foodName)
    // um einen Wert der mit nutritionalValue
    // zusammenhängt.
    void eat(String foodName,
         double nutritionalValue) {/*...*/}
}
```

Listing 2: Locomotion

```
public interface Locomotion {
    // Wechselt das Objekt in den
    // nächsten Bewegungszustand:
    // 1.) langsames Bewegen
    // 2.) schnelles Bewegen
    // 3.) spezielles Bewegen
    // und führt diese Bewegung durch.
    // Von Zustand 3 wird wieder
    // zum ersten Zustand
    // zurückgesprungen.
    public double doTheLocomotion();

    // Setzt das Objekt in einen
    // unbewegten Zustand.
    // doTheLocomotion fängt danach
    // wieder mit den ersten Zustand
    // an.
    public void stopTheLocomotion();
}
```

Aufgabe 1: Bewegen

Verbessern Sie die gegebene Klasse `Ladybird`, indem Sie die Methode und einen geeigneten Konstruktor implementieren. Die Klasse soll dabei das auch Interface `Locomotion` implementieren. „Spezielles Bewegen“ soll dabei ein Richtungswechsel beinhalten. Schreiben Sie ein kleines Hilfsprogramm welches `doTheLocomotion` und `stopTheLocomotion` mehrmals aufruft. `toString` soll dabei auch immer die Position und die Richtung ausgeben.

Erfüllen Sie auch alle anderen Anforderungen der allgemeinen Hinweisen.

Aufgabe 2: Distanzen

Ergänzen Sie die Klasse um eine Methode `double dist(Ladybird p)`. Diese soll den Abstand zwischen dem aktuellen Objekt und dem Objekt im Parameter zurückliefern. Mathematische Berechnungen wie die Quadratwurzel finden Sie in der Java-Klasse `Math`.

Aufgabe 3: Fliegen

Marienkäfer können natürlich auch fliegen. Verändern Sie in `doTheLocomotion` den dritten Zustand, sodass Marienkäfer ihre Richtung auch nach oben und

unten verändern können.

Passen sie auch `dist`, `toString` und `equals` entsprechend an. Geben Sie Beispiele wie sich das Verhalten von `doTheLocomotion` jetzt verändert hat.

Aufgabe 4: Essen

Verbessern Sie die Methode `void eat(String, double)` mit Hilfe von einem neuen Interface `Food`. Achten Sie dabei auf die Funktionalität der Methode `void eat(String, double)`. Beachten Sie, dass Marienkäfer nicht beliebig viel essen können und ihr Gewicht sich nur grammweise verändern kann.

Schreiben Sie zwei Klassen welche `Food` implementieren. Geben Sie Code-Beispiele wie Marienkäfer Instanzen dieser Klassen essen.

Aufgabe 5: Untertypen

Es gibt über 5000 Arten in der Familie der Marienkäfer. Implementieren Sie drei davon und nehmen Sie auf spezifische Gegebenheiten Rücksicht. So kann Spiladelpha barovskii nicht fliegen, Vierzehnpunkt-Marienkäfer haben keine einheitlichen Farb- bzw. Mustervarianten oder Zweiundzwanzigpunkt-Marienkäfer können sich tot stellen.

Überschreiben Sie die Methoden `doTheLocomotion`, `toString` oder `eat` mit den speziellen Verhaltensweisen. Geben Sie Code-Beispiele für die Verwendung der verschiedenen Arten mit ihren Eigenheiten.

Theoriefragen

Beantworten Sie die folgenden Fragen.

Aufgabe 6: Paradigmen

Wodurch unterscheidet sich der objektorientierte von anderen Programmierstilen? Für welche Arten von Programmen eignet sich die objektorientierte Programmierung gut, für welche nicht?

Aufgabe 7: Kapselung

Was sind und wozu verwendet man Klassen? Was ist und wozu dient Kapselung, Data-Hiding und Datenabstraktion? Was haben Getter- und Setter-Methoden mit Data-Hiding zu tun?

Aufgabe 8: Sichtbarkeit

Wie kann man in Java die Sichtbarkeit beeinflussen? Wo sollen die meisten Objektvariablen sichtbar sein? Für welche Instanzen sind in Java auch private Details zugänglich?

Aufgabe 9: Dynamisches Binden

Wozu benötigt man dynamisches Binden? Inwiefern hängt dynamisches Binden mit Mehrfachverzweigungen zusammen? Warum ist dynamisches Binden gegenüber `switch`-Anweisungen zu bevorzugen?

Aufgabe 10: Untertypen

Was besagt das Ersetzbarkeitsprinzip? Unter welchen Bedingungen ist ein Typ U Untertyp eines Typs T? Was versteht man unter Vererbung?

Das sollten Sie unbedingt kennen.