

Dokumentation zur Projektarbeit Programmieren 2

Für die Projektarbeit im Fach Programmieren 2 wurden zur Herangehensweise der Aufgabe mehrere Lösungsmöglichkeiten angefertigt. Die Lösungen, die auch für die spätere Umsetzung und Erweiterung der Programmierung der Aufgabe dienen, werde ich nun im Detail genauer veranschaulichen bzw. erklären.

Im folgenden möchte ich nun die drei Lösungsmöglichkeiten aufzeigen:

1. Lösung: Erste Idee für das Projekt
2. Lösung: Das programmierte Projekt mit BlueJ
3. Lösung: Mögliche Erweiterungen für das Projekt

Lösung No.1: Erste Idee für das Projekt

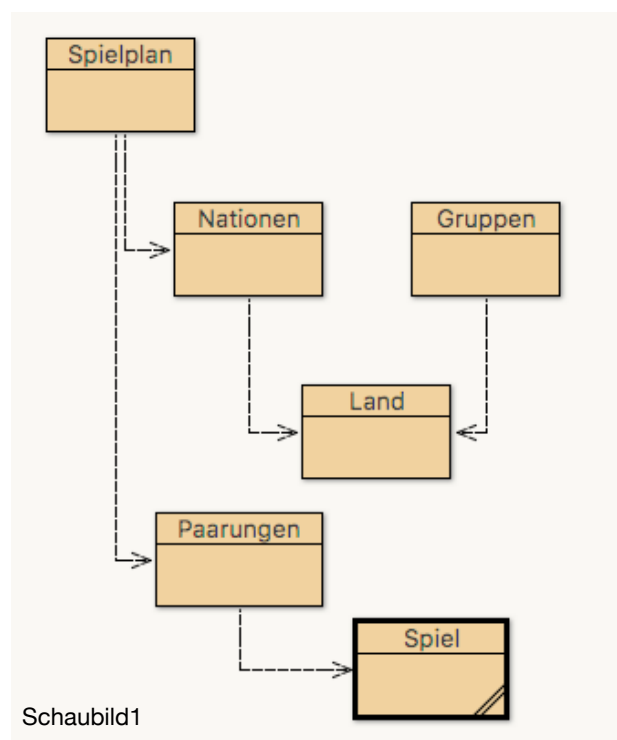
Der erste Ansatz für die Umsetzung der Projektarbeit war das dargestellte Schaubild1 als Klassendiagramm.

Die Idee dahinter war eine Klasse Land zu erstellen, die unendlich viele Länder mit Informationen (Name, Tore, Punkte) als Arrays beinhalten kann. Zudem eine Klasse Nationen, in der die 32 Nationen der diesjährigen WM als ArrayList gespeichert werden. Die Überlegung war die Klasse Gruppe nicht mit den Nationen zu verknüpfen, sondern die Gruppen aus den Ländern in eine separate ArrayList zu speichern. Dies stellte sich im Nachhinein als Problem da, weil zwei unterschiedliche Listen von Ländern geführt wurden.

In der Klasse Paarung sollen alle Spiele der Weltmeisterschaft ermittelt werden. Hintergrund des Konzepts war das in der Klasse Spiele Informationen wie Spielnamen und Spieltore gespeichert werden. Diese Infos werden danach in einer ArrayList der Klasse Paarungen übergeben, in der auch die Berechnung der Paarung statt findet.

Die Ergebnisse der Spiele sollten in der Klasse Spielplan berechnet werden, da es die Hauptklasse ist und die Informationen an die anderen Klassen weiterleiten könnte. Der Gedanke war die Klasse Spielplan könnte die Punkte und Tore aus den Spielen dem Land übergeben damit die Informationen somit immer aktuell sind. Dieser Ansatz stellte sich durch die spätere Codeentwicklung als fehlerhaft da, weil die Main-Klasse immer die darunter liegenden Klassen referenziert.

Aufgrund der oben genannten Punkte wurde von diesem Klassenentwurf etwas abgewichen, um auch später mit den Konzepten wie Kopplung und Kohäsion besser umzugehen



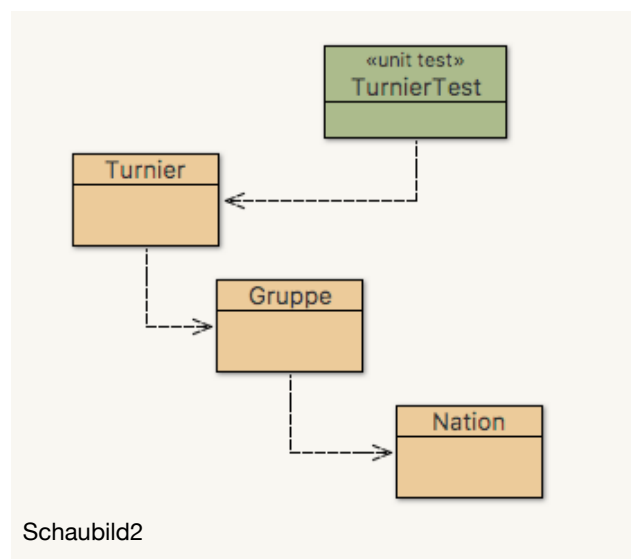
Lösung No.2: Das programmierte Projekt mit BlueJ

Das Schaubild2 zeigt das Klassendiagramm der programmierten Projektarbeit „WM Scheduler“ mit BlueJ. Der Code entstand durch den Ansatz der Idee aus Lösung 1 jedoch mit einem anderen Klassenentwurf. Grundlegend ist der Klassenaufbau so strukturiert das über die Klasse Turnier auf die Klasse Gruppe und Nation verwiesen wird. Dabei wurde stark auf lose Kopplung geachtet damit alle Klassen weitgehend unabhängig bleiben und die Kommunikation zu anderen Klassen über Schnittstellen erfolgt, wie z.B. Objektreferenzen zu Gruppen- und Nationenobjekten.

In der Klasse Nation werden die Variablen für die einzelnen Nationen initialisiert und deklariert. Die Nationen-Objekte bestehen aus einfachen Datentypen wie Integer und Strings. In der Klasse Gruppe werden neue Nationen-Objekte in eine HashMap hinzugefügt. Für die HashMap habe ich mich entschieden, weil ich durch die Schlüssel-Wert-Paare auf die Informationen der Nationen zugreifen kann. Man könnte auch Arrays benutzen, jedoch geht die Suche nach Inhalten wesentlich einfacher und schneller als beim Array. Außerdem muss man dem Array die Länge von vorn herein festlegen während die HashMap flexibel in Größe und Länge ist. Nachteil der HashMap ist die Sortierung der Informationen, die ich mit Hilfe der Benutzung von lokalen ArrayListen bewerkstelligen konnte. Durch die Kombination aus HashMap und ArrayList fand somit eine Möglichkeit der Erweiterbarkeit von Gruppen und Nationen statt.

Der Entwurfsprozess meines Codes besagt, das der Nutzer hauptsächlich über die Klasse Turnier interagiert. Dabei muss er keine Objekte der Klasse Nation oder Gruppe direkt über BlueJ aufrufen. Jede Methode in der Klasse Turnier ist genau für eine Aufgabe zuständig, da auf Kohäsion geachtet wurde. Es wurde beim Klassenentwurf außerdem auf die Kapselung der Daten geachtet und damit die Änderungen der Klassen möglichst lokal gehalten werden. Dies erzielte ich durch die Getter- und Setter-Methoden in den Klassen. Zum Thema Code-Duplizierung habe ich das System auf doppelte Quelltextabschnitte geprüft.

Die Gruppen- und Nationenverwaltung wird durch die Klasse Turnier erzielt. Die Bedienung des „WM Schedulers“ erfolgt ebenfalls über die Klasse Turnier, in dem man die einzelnen Methoden aufruft um den WM Planer mit Daten zu befüllen oder Daten abzurufen. Das Programm kann über die Klasse Turnier Test getestet werden, dazu wurden 12 Test-Methoden angelegt, die erfolgreich verlaufen sind. Werden Fehler bei der Eingabe der Tore und Punkte der Nation gemacht müssen diese durch Methoden überschrieben werden.



Lösung No.3: Mögliche Erweiterungen für das Projekt

Die Möglichen Erweiterungen für das Projekt können im Klassendiagramm des Schaubilds3 betrachtet werden. Dieses zeigt bis auf die Klasse IO keine Unterschiede zur Lösung2. Die Erweiterung des Codes liegt also in der Bedienung für den Benutzer. Dieser kann bis zur Lösung2 nur über das Aufrufen der Klasse Turnier mit dem System interagieren.

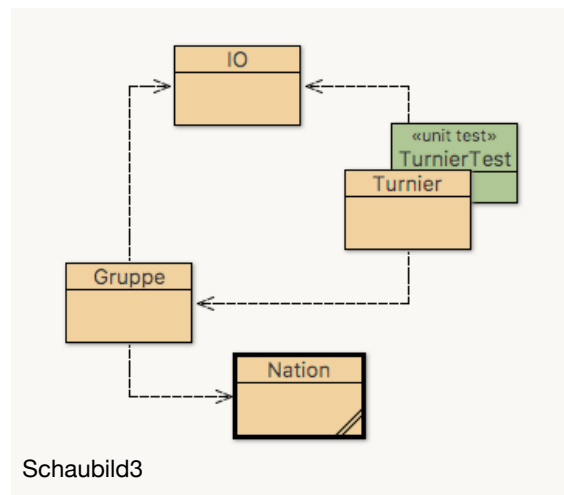
Gedanke ist über Klasse IO die Daten aus einer Textdatei auszulesen bzw. einzulesen. Dazu müsste man zwei Ordner mit Textdateien anlegen. Einen Ordner für die Nationen mit allen darin befindlichen Nationen als Textdateien. Die erste Zeile wäre der Nation-Name, gefolgt von den Zeilen Tore und Punkte. Der zweite Ordner wäre die Gruppen mit der ersten Zeile für die Nationanzahl und den darunter liegenden Gruppennamen.

Damit die Textdateien aber überhaupt erst lesbar oder schreibbar wird muss die Java Klasse „java.io.FileReader“ „java.io.FileWriter“ in das Projekt integriert werden.

Damit die Textdateien auch bearbeitet werden können müssen die Nationen als String Array implementiert werden. Dabei hilft die Methode split um Strings anhand gewisser Kriterien in mehrere Einzelteile aufzusplitten.

Eine andere Möglichkeit wäre die Klasse „java.util.Scanner“, da diese Zeichenketten in Tokens zerlegen und die Dateien zeilenweise einlesen kann.

Aufgrund der Umstrukturierung der Codes und der nicht allzu leichten Umsetzung der Textdateien konnte ich diese leider nicht in das System integrieren.



Fazit

Die Projektarbeit im Fach Programmieren 2 hat mir gezeigt wie ich mit der Aufgabenstellung in der Programmiersprache Java umgehe. Auch wenn ich nicht alle Aufgabenstellungen bis ins Detail lösen konnte, habe ich viel in diesem Projekt für meine Programmierkenntnisse gelernt.

Gerade die Konzepte des Klassenentwurfs wie Kopplung, Kohäsion und Codeduplizierung sowie Fehlervermeidung kann ich nun im Code anwenden. Darüberhinaus konnte ich die Java Konstrukte Array, HashMap und ArrayList für das Projekt anwenden und mein Wissen für diese essentiellen Bibliotheksklassen erweitern.

Die aus dem Projektarbeit erlernten Fähigkeiten und Erkenntnisse helfen mir später in anderen Programmiersprachen oder bei Projekten im Praxissemester.

Durch das Projekt verstehe ich nun die Zusammenhänge der Programmierung in Java besser und ich bin gewillt weiter daran zu arbeiten.

Kaay