

Inhaltsverzeichnis

Inhaltsverzeichnis	1
1. Problemdefinition.....	2
2. Problembeschreibung	3
3. Analyse	3
3.1. Use Case Diagramm 1	4
3.2. Use Case Diagramm 2	5
3.3. Klassendiagramm.....	6
3.4. Sequenzdiagramm	6
4. Klassenbeschreibung.....	7
4.1. public class Punkt.....	7
4.2. public class Schiff.....	7
4.3. public class Feld.....	7
4.4. public class Hilfe	8
4.5. public abstract class Teilnehmer	8
4.6. public class Spieler extends Teilnehmer.....	8
4.7. public class Rechner extends Teilnehmer	9
4.8. public class Spiel.....	9
4.9. public class Toggle extends JFrame implements ActionListener	9
4.10. Klasse „In“ und Klasse „Main“	10
5. Beschreibung für den Anwender	10
6. Wartung.....	11

1. Problemdefinition

Datum: 11.Oktober 2005

Projektbezeichnung: Softwareversion vom Spiel „Schiffe Versenken“

Autor: Martin Vodenicharov

Matrikelnummer: 0253795

E-mail: 0253795@wu-wien.ac.at

Kurzbeschreibung:

In diesem Projekt wird das Spiel „Schiffe Versenken“ in eine Softwareversion gebracht. In der Grundversion wird die Ein- und Ausgabe textbasiert sein. In einer Upgradeversion soll es möglich sein, das Spiel über ein komfortables User-Interface bedienen zu können.

Projektziele:

Erstellung einer funktionsfähigen, leicht zu bedienenden Software. Des Weiteren soll Know-how im Bereich Design und Implementierung von Java aufgebaut werden.

Projektumfang:

Das Projekt läuft im Rahmen der Lehrveranstaltung Rechnerpraktikum aus Programmierung, das ein Semester dauern wird. Somit wird der Zeitaufwand in etwa 15 Wochen betragen.

Das Spiel „Schiffe Versenken“:

Ziel des Spiels „Schiffe Versenken“ besteht darin, die Schiffe von dem Gegner, in diesem Fall der Rechner, durch Eingabe von zwei Zahlen, die einen Punkt auf dem Feld darstellen, zu versenken. Jeder Spieler hat nur einmal das Recht zu fragen, dann fragt der andere Spieler. Fragen heißt in diesem Fall, dass zwei Zahlen entweder eingegeben werden(durch Spieler) oder durch einen Zufallsgenerator erzeugt werden(im Falle des Rechners)

2. Problembeschreibung

Ziel dieses Projekt ist ein lauffähiges Programm mit Hilfe der Programmiersprache „Java“ zu erstellen, das das bekannte Spiel „Schiffchen versenken“ darstellen soll. Das Spiel wird von einem Spieler gespielt (gegen den Rechner). Das Spiel soll zwei Felder (mit der Größe 6x6 zum Beispiel) generieren können. In dem einen sind die Schiffe, durch Punkte dargestellt, des Spielers eingetragen, in dem anderen sind keine Schiffe. Durch Eingabe von zwei Ziffern, die einen Punkt auf dem leeren Feld darstellen, „sucht“ der Spieler ein Schiffchen des Rechners. Danach soll dieser Punkt auf dem leeren Feld entweder als „gefragt“ oder als „getroffen“ markiert werden. Dasselbe gilt auch für den Rechner, wobei bei dem die Zahlen durch Zufallsgenerator erzeugt werden. Der Einfachheit halber soll in der ersten Version dieses Spieles nur je ein einziges Schiff für beide Seite zur Verfügung stehen. Ein Schiff ist in dem Fall eine Reihe von vier Punkten. Wenn alle vier Punkte als „getroffen“ gekennzeichnet sind ist das Schiff versenkt und das Spiel vorbei. Aufbauend auf dieser ersten Version soll weitere Versionen erstellt werden, bei denen dann schrittweise die Anzahl der Schiffe und die Größe der Felder erhöht werden.

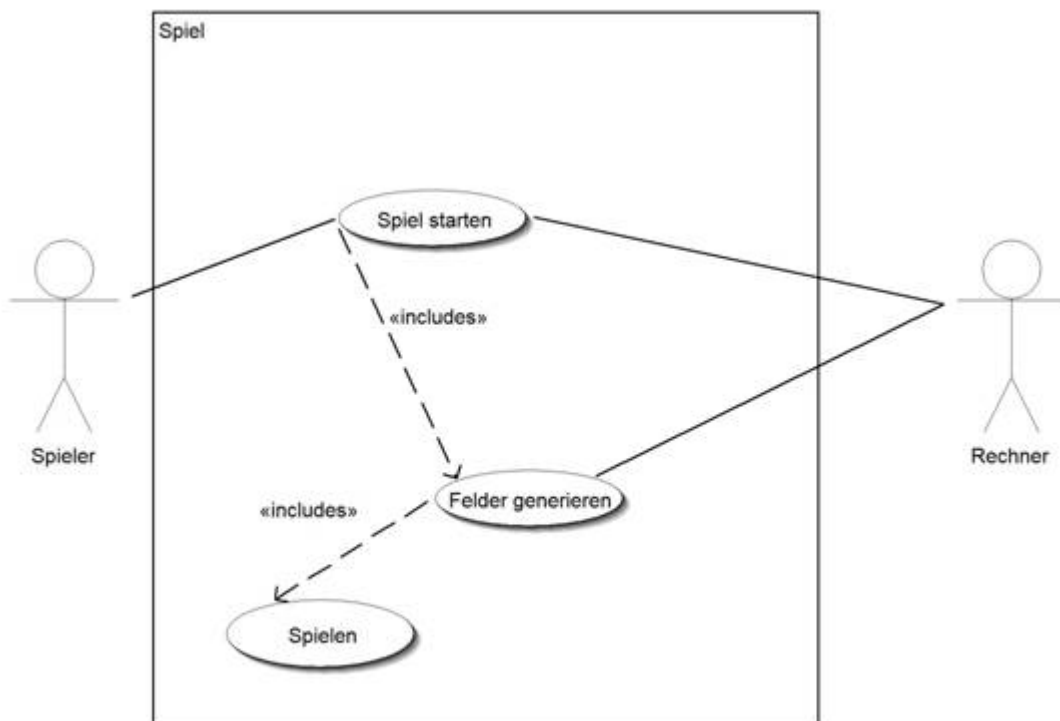
3. Analyse

Spieler startet das Spiel. Es werden zwei Felder generiert. Durch Klicken des Start-Buttons werden die Schiffe des Spielers in dem einen Feld erzeugt. Das andere Feld ist leer. Dort befinden sich die Schiffe des Rechners, die aber für den Spieler nicht sichtbar sind. Spieler fängt an, in dem leeren Feld Feldchen anzuklicken. Dann ist der Rechner dran. Sollte der Spieler auf einem Feldchen klicken, wo sich ein Teil einer der Schiffe des Rechners befindet, wird dieses Feldchen als getroffen markiert. Wird der letzte Teil eines Schiffes getroffen, dann ist das Schiff als versenkt zu markieren. Dasselbe gilt auch für den Rechner. Der macht seine Züge durch einen Zufallsgenerator, der die Koordinaten eines Punktes vorgibt. Sollte der Rechner ein Feldchen treffen, wo sich ein Teil eines Schiffes des Spielers befindet, wird keine Zufallszahl erzeugt, sondern der Rechner fängt an, um das getroffene Feld zu „suchen“. Sobald ein Schiff, das sich um das getroffene Feldchen befunden hat, versenkt ist, wird für das weitere Suchen wieder eine Zufallszahl erzeugt. Gewinner des Spieles ist derjenige, der die Schiffe des Gegners zuerst versenkt. Das Spiel ist vorbei. Das bedeutet, dass weder dem Spieler noch dem Rechner möglich sein kann, weitere Feldchen anzuklicken bzw. weitere Zufallszahlen zu erzeugen.

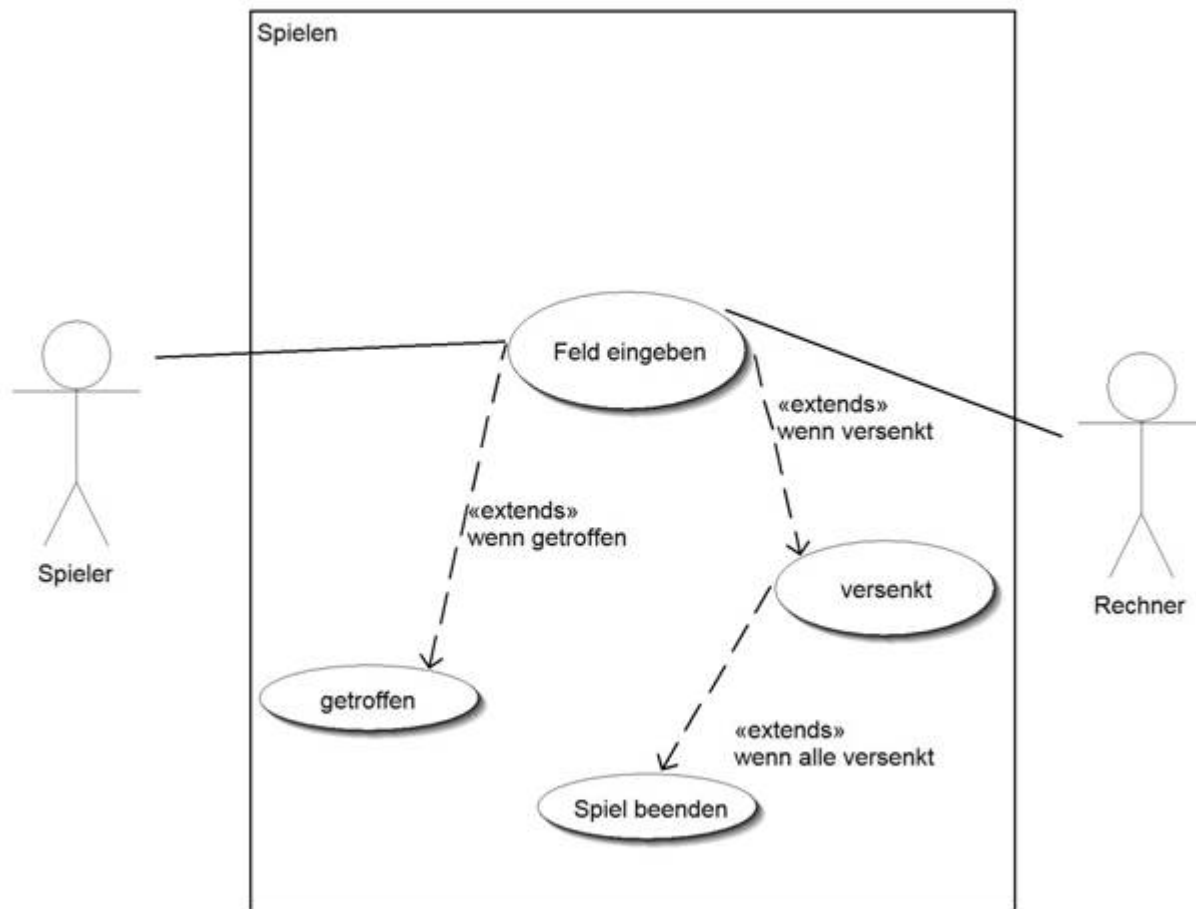
Ausgehend vom obigen Sachverhalt und mit Hilfe von der Modellierungssprache UML habe folgende Use Case Diagramme sowie UML Diagramme entworfen. Aufbauend auf diese habe ich begonnen, das Spiel schrittweise zu implementieren. Mein Vorhaben war folgendes. Zuerst wollte ich das Spiel ohne

graphische Oberfläche implementieren, wobei die Schiffe nur durch ein einziges Feld dargestellt waren. In einem weiteren Schritt habe ich auch die graphische Oberfläche implementiert, wobei dafür einige kleine Veränderungen an den Code vorzunehmen waren. Wenn das Spiel auch mit Graphische Oberfläche ungefähr fertig gestellt worden war, habe ich andere Änderungen vorgenommen, um mehrdimensionale Schiffe einzuführen.

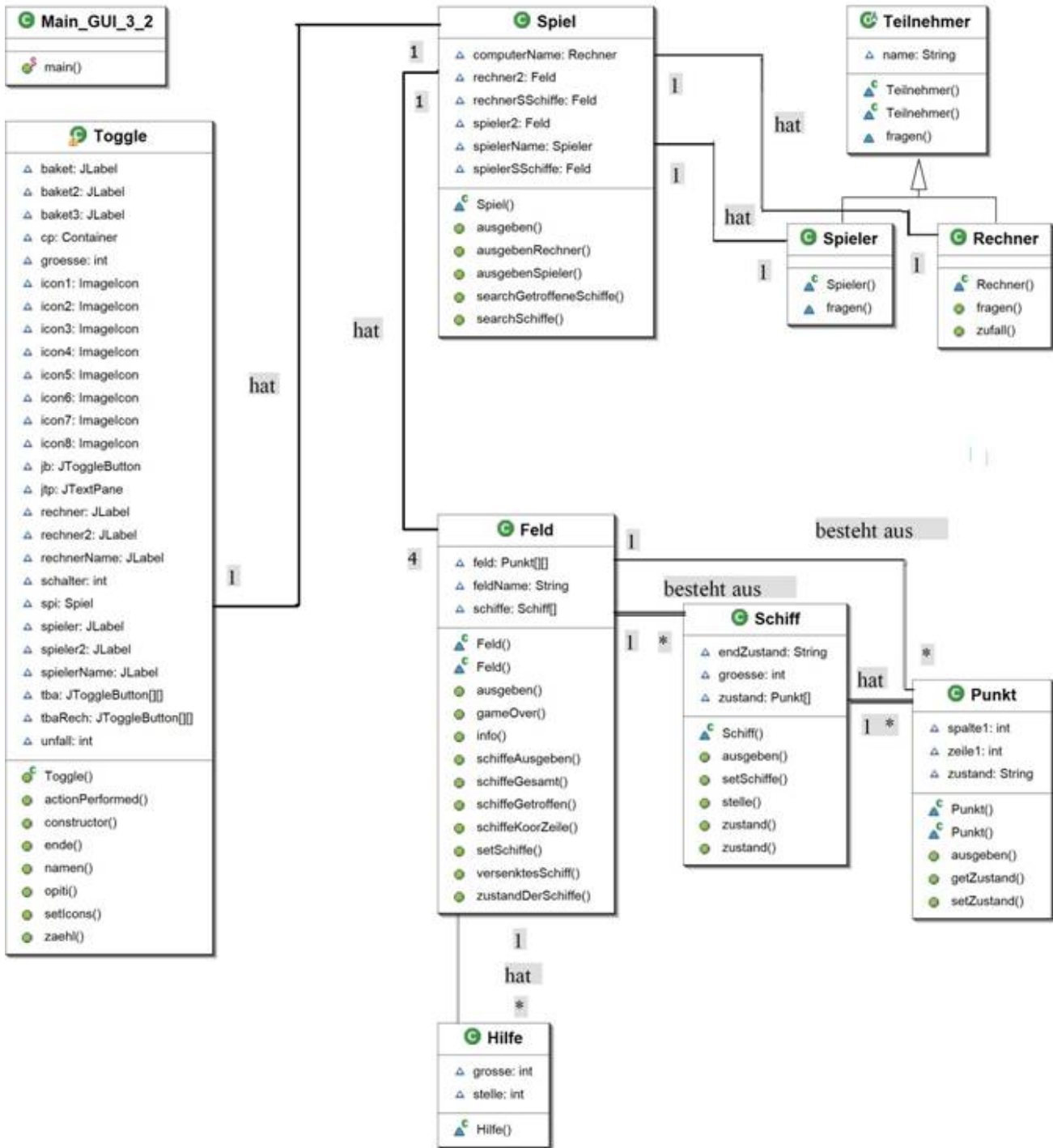
3.1. Use Case Diagramm 1



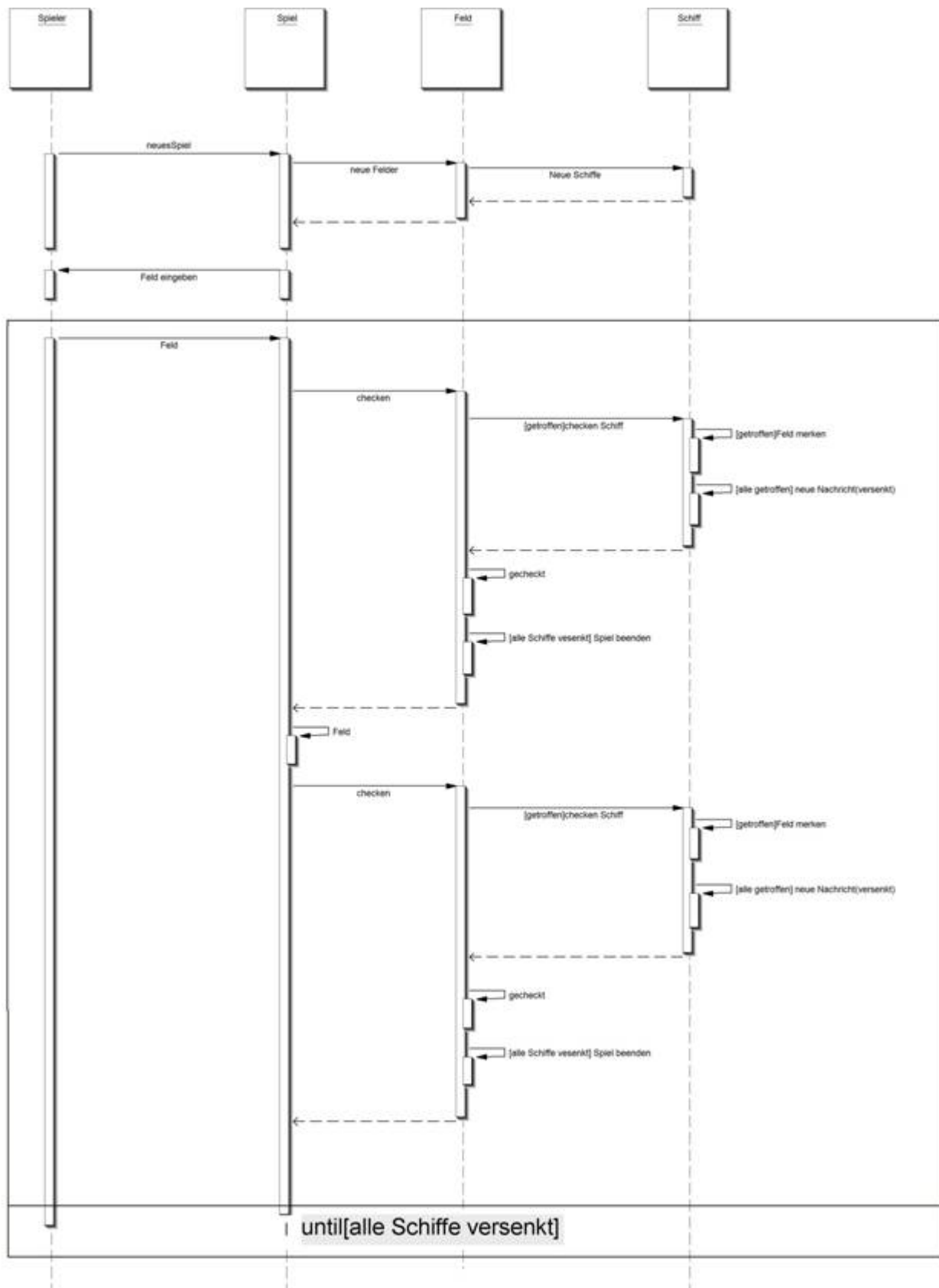
3.2. Use Case Diagramm 2



3.3. Klassendiagramm



3.4. Sequenzdiagramm



4. Klassenbeschreibung

4.1. public class Punkt

Die Klasse "Punkt" ist die Grundlage für die Klasse "Schiff". Sie hat drei Variable. Diese sind zwei int - Variable und eine String-Variable. Die int - Werte stellen einen Punkt auf dem Feld dar und die String-Variable beinhaltet den Zustand dieses Punkts. Methoden dieser Klasse sind:

- Punkt()
- Punkt(String zustand)
- public String getZustand()
- public void setZustand(String zustand)
- public void ausgeben()

4.2. public class Schiff

Diese Klasse implementiert das Objekt Schiff. Sie besteht aus folgenden Variablen:

- int groesse
- Punkt[] zustand
- String endZustand

Die Variable „groesse“ steht für die Größe eines Schiffes. Das Array „zustand“ beinhaltet Objekte der Klasse „Punkt“. Die Länge des Array ist gleich der Größe eines Schiffes. Die String – Variable „endZustand“ steht für die Beschreibung des gesamten Zustands eines Schiffes. Methoden dieser Klasse sind:

- Schiff(int groesse)
- public void ausgeben()
- public void setSchiffe(Feld feld)
- public void zustand(Feld feld)
- public int stelle(int zeile, int spalte)
- public int zustand()

4.3. public class Feld

Diese Klasse implementiert das Objekt „Feld“. Sie besteht aus folgenden Variablen:

- Punkt[][] feld
- String feldName
- Schiff[] schiffe

Die Variable „feld“ stellt ein zweidimensionales Array, auf dem sich Schiffe „befinden“. Dies ist durch das Array „schiffe“ implementiert. Die Klasse verfügt über folgende Methoden:

- Feld()
- Feld(String fieldName)
- public void info(int l, int b)
- public void ausgeben()
- public void setSchiffe()
- public void schiffeAusgeben()
- public void schiffeKoorZeile(int zeile, int spalte)
- public void zustandDerSchiffe()
- public int gameOver()
- public int schiffeGesamt
- public int schiffeGetroffen()
- public Hilfe
versenktesSchiff(int zeile, int spalte)

4.4. public class Hilfe

Da es sich herausgestellt hat, dass ein Objekt benötigt wird, das zwei int – Werte als Variablen hat, wurde die Klasse „Hilfe“ eingeführt. Diese wird nur durch die Klasse „Feld“ verwendet. Variablen dieser Klasse sind:

- int stelle
- int grosse

Methoden dieser Klasse sind:

- Hilfe()

4.5. public abstract class Teilnehmer

Diese Klasse ist Superklasse der Klassen „Spieler“ und „Rechner“. Sie hat folgende Methoden und Variablen:

- Teilnehmer(String name)
- Teilnehmer()
- void fragen()
- String name

4.6. public class Spieler extends Teilnehmer

Diese Klasse erbt von der Klasse „Teilnehmer“. Sie verfügt über Methoden:

- Spieler(String name)
- void fragen(Spiel spi, int l, int b)

4.7. `public class Rechner extends Teilnehmer`

Diese Klasse erbt von der Klasse „Teilnehmer“. Sie verfügt über folgende Methoden:

- `Rechner(String name)`
- `public void fragen(Spiel spi)`
- `public int zufall()`

4.8. `public class Spiel`

Diese Klasse hat als Variablen vier Instanzen der Klasse „Feld“ sowie eine der Klasse „Rechner“ und eine der Klasse „Spieler“. Ausgehend von der Größe eines Feldes wird eine „for-Schleife“ solange durchlaufen, bis alle Schiffe, entweder die vom Rechner oder die vom Spieler, versenkt worden sind. Sie hat diese Methoden:

- `Spiel(String name)`
- `public void ausgeben()`
- `public void ausgebenSpieler()`
- `public void ausgebenRechner()`
- `public int searchGetroffeneSchiffe(Feld feld)`
- `public int searchSchiffe(Spiel spi)`

4.9. `public class Toggle extends JFrame implements ActionListener`

Diese Klasse wurde in „Game“ umbenannt. Hier wird die graphische Oberfläche erzeugt. Zentrale Variablen dieser Klasse sind „`JToggleButton[][] tba`“ und „`JToggleButton[][] tbaRech`“. Diese stellen zweidimensionale Arrays vom Typ „`JToggleButton`“, das sind Zwei-Zustandsbuttons. Weitere Variable dieser Klasse sind:

- `ImageIcon icon1`
- `ImageIcon icon2`
- `ImageIcon icon3`
- `ImageIcon icon4`
- `ImageIcon icon5`
- `ImageIcon icon6`
- `ImageIcon icon7`
- `ImageIcon icon8`
- `JToggleButton[][] tba`
- `JToggleButton[][] tbaRech`
- `JTextPane jtp`
- `JLabel spieler`
- `JLabel rechner`
- `JLabel spieler2`
- `JLabel rechner2`
- `JLabel spielerName`
- `JLabel rechnerName`
- `int schalter`
- `int groesse`
- `int unfall`
- `JLabel baket`
- `JLabel baket2`
- `JLabel baket3`
- `Spiel spi`
- `JToggleButton jb`
- `Container cp`

Methoden dieser Klasse sind:

- public Toggle()
- public void actionPerformed (ActionEvent ae)
- public void opiti ()
- public int zaehl ()
- public void ende(String name)
- public void setIcons(Schiff schiff, JToggleButton[][] tbaRech)
- public JLabel namen()
- public void constructor()

4.10. Klasse „In“ und Klasse „Main“

Klasse „In“ diente der ersten Version des Spieles, bei der Eingabe von Zahlen durch den Spieler notwendig war. Die Klasse „Main“ startet das Spiel.

5. Beschreibung für den Anwender

Alle Klassen und Abbildungen sollen sich in der gleichen Mappe befinden. Das Spiel „Schiffe versenken“ wird in der Klasse „Main_GUI_3_3“ gestartet. Diese muss kompiliert werden. Das geschieht durch Eingabe der Anweisung javac Main_GUI_3_3 in dem DOS-Fenster. Dann wird gestartet durch java Main_GUI_3_3. Zuerst muss aber bis zur Mappe navigiert werden, wo sich die Klassen befinden.

Als erstes muss man den Namen eingeben und dann auf „Weiter“ drücken. Danach kann/soll man Felder anklicken. Es sind fünf Schiffe zu versenken. Die Größe der Schiffe ist wie folgt:

Ein Schiff der Größe eins(um versenkt zu werden muss dieses Schiff ein Mal getroffen werden)

Ein Schiff der Größe zwei(um versenkt zu werden muss dieses Schiff zwei Mal getroffen werden)

Ein Schiff der Größe drei(um versenkt zu werden muss dieses Schiff drei Mal getroffen werden)

Ein Schiff der Größe vier(um versenkt zu werden muss dieses Schiff vier Mal getroffen werden)

Ein Schiff der Größe fünf(um versenkt zu werden muss dieses Schiff fünf Mal getroffen werden)

Die eigenen Schiffe befinden sich auf dem rechten Feld. Dort „fragt“ der Rechner nach. Diese sind für den Spieler sichtbar. Wenn ein Schiff, dessen Größe größer als eins ist, getroffen wird, wird dies durch eine Flammenabbildung sichtbar. Sollten alle Felder eines Schiffes getroffen sein, werden alle Felder dieses Schiffes durch eine Abbildung, die ein versenktes Schiff darstellt, gekennzeichnet. Wer zuerst die Schiffe des Gegners versenkt gewinnt das Spiel.

Ist das der Fall wird der Name des Gewinners im Textfeld links unten sichtbar, wobei auch die Anzahl der dafür notwendigen Züge angezeigt wird.
Folgende Abbildungen sollen veranschaulichen, in welchem Zustand sich ein Feld gerade befindet:



- Anfangszustand



- Feld, das angeklickt bzw vom Rechner gefragt wurde und dort kein Schiff war.



- Feld, das angeklickt wurde und dort ein Schiff ist(ein Teil von dem) oder eigene Schiffe in dem linken Feld



- Feld, wo ein Teil eines der Schiffe des Spielers getroffen wurde.



- Wenn ein Schiff zur Gänze versenkt wurde, werden alle seine Felder durch diese Abbildung dargestellt.

6. Wartung

Grundsätzlich ist keine Wartung des Programms vorgesehen. Treten jedoch unerwartet Fehler auf, kann man mir gerne eine e-Mail schicken, in der eine genaue Fehlerbeschreibung des aufgetreten Problems mitgeschickt wird. Ich werde dann versuchen, den Fehler in einer neueren Version zu beheben und dem Sender die überarbeitete Version zurückzuschicken. Des Weiteren kann die Schwierigkeitsgrad in einer zukünftigen Version erhöht werden, indem die Art wie der Rechner „fragt“ (dh. Schiffe sucht) verbessert wird. Eine weitere Änderung kann auch vorgenommen werden, damit der Spieler die Plätze seiner Schiffe selbst bestimmen kann.