

**Projekt: Würfelpoker**

Gert Vasak  
Matrikelnummer 9150666

**PROBLEMSTELLUNG**

**Ziel**

Ziel dieses Projektes ist es, eine einfache Version des Brettspiels ‚Würfelpoker‘ in Java zu programmieren.

Es soll dem User möglich sein, anhand einer vorgegebenen Scoreliste so viele Punkte wie möglich zu erreichen; das Spiel (Computer) füllt die Scoreliste nach den Wünschen des Users aus und aktualisiert seinen jeweiligen Punktestand. Am Ende wird die Gesamtpunktzahl angezeigt.

**Darstellung**

Für das Spiel werden fünf (virtuelle) Würfel erzeugt.

Ein Würfel besteht aus sechs Würfelseiten, die die Symbole 9, 10, Bube, Dame, König und Ass darstellen.

In der ersten Version (Prototyp) wird das Spiel über die Windows-Eingabeaufforderung gestartet, das Spielen ist über eine textbasierte Eingabe möglich.

In der Endfassung soll das Spiel als Applet implementiert werden; dabei sollen die Würfel zweidimensional – Ansicht von oben – dargestellt werden.

**Spielanleitung**

Für das Spiel gelten die allgemeinen Regeln für (Würfel-)Poker:

Der Spieler hat die Möglichkeit, drei Mal pro Runde zu würfeln, wobei er vor Wurf Nr. 2 und 3 jene Würfel, die schon einen für ihn nützlichen Wert haben, halten kann.

Insgesamt stehen zehn solcher Wurfserien zur Verfügung, um die jeweils bestmögliche Punktzahl für folgende vorgegebene Kombinationen zu erreichen:

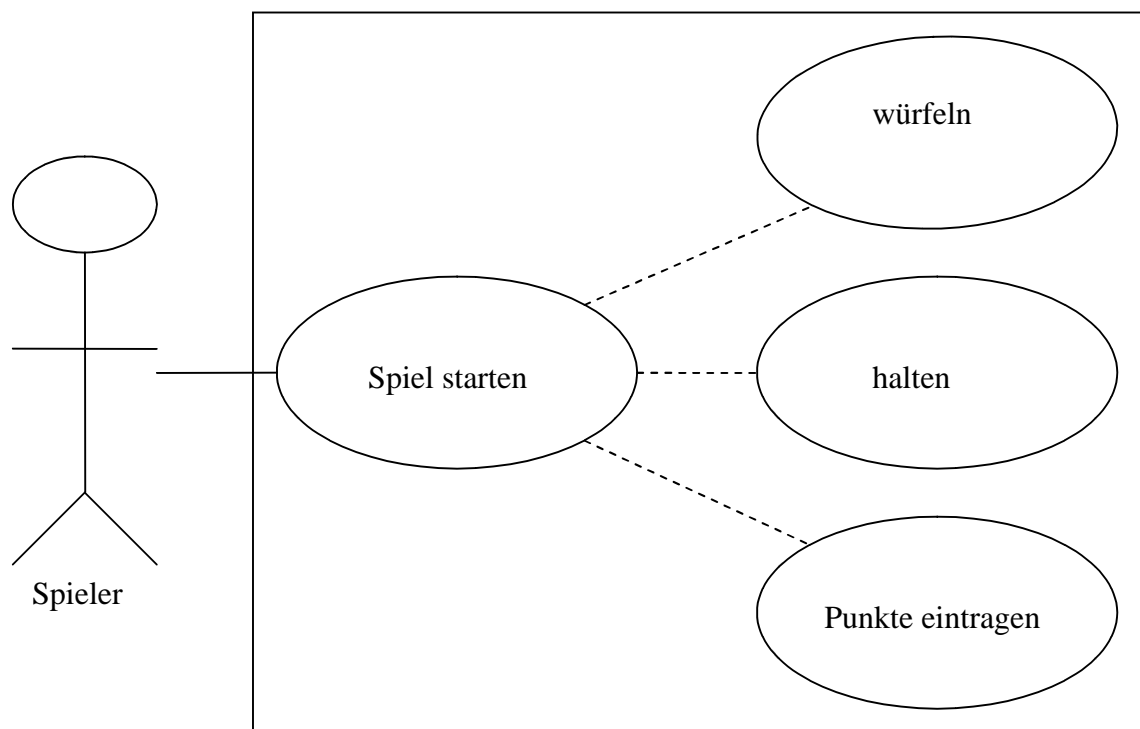
- Höchste Anzahl an Würfeln mit dem jeweiligen Symbol (9 – Ass, Symbole zählen in aufsteigender Reihenfolge ein bis sechs Punkte)
- Strasse (Würfelsymbole in ansteigender Reihenfolge, 20 Punkte)
- Full House (drei Würfel eines Wertes und zwei Würfel eines anderen Wertes, 30 Punkte)
- Poker (vier gleiche Würfelsymbole, 40 Punkte)
- Grande (fünf gleiche Würfelsymbole, 50 Punkte)

Jede Kombination darf genau ein Mal eingetragen werden. Der Spieler kann sich aussuchen, welches Ergebnis er wo eintragen lässt bzw. welche Kombination er streichen lässt, weil er sie nicht erreicht.

Am Ende des Spiels wird die Gesamtpunktezahl angezeigt, der Spieler wird gefragt, ob er noch einmal spielen möchte.

Eine Highscore-Wertung ist vorläufig noch nicht geplant.

## USE CASE DIAGRAMM



### Use Case ‚Spiel starten‘

Der User öffnet das Spiel über die Windows-Eingabeaufforderung. Das Spiel beginnt sofort.

### Use Case ‚würfeln‘

Nach Spielbeginn wird automatisch die erste Runde gewürfelt. Anschließend wird der User gefragt, welche der Würfel er halten möchte.

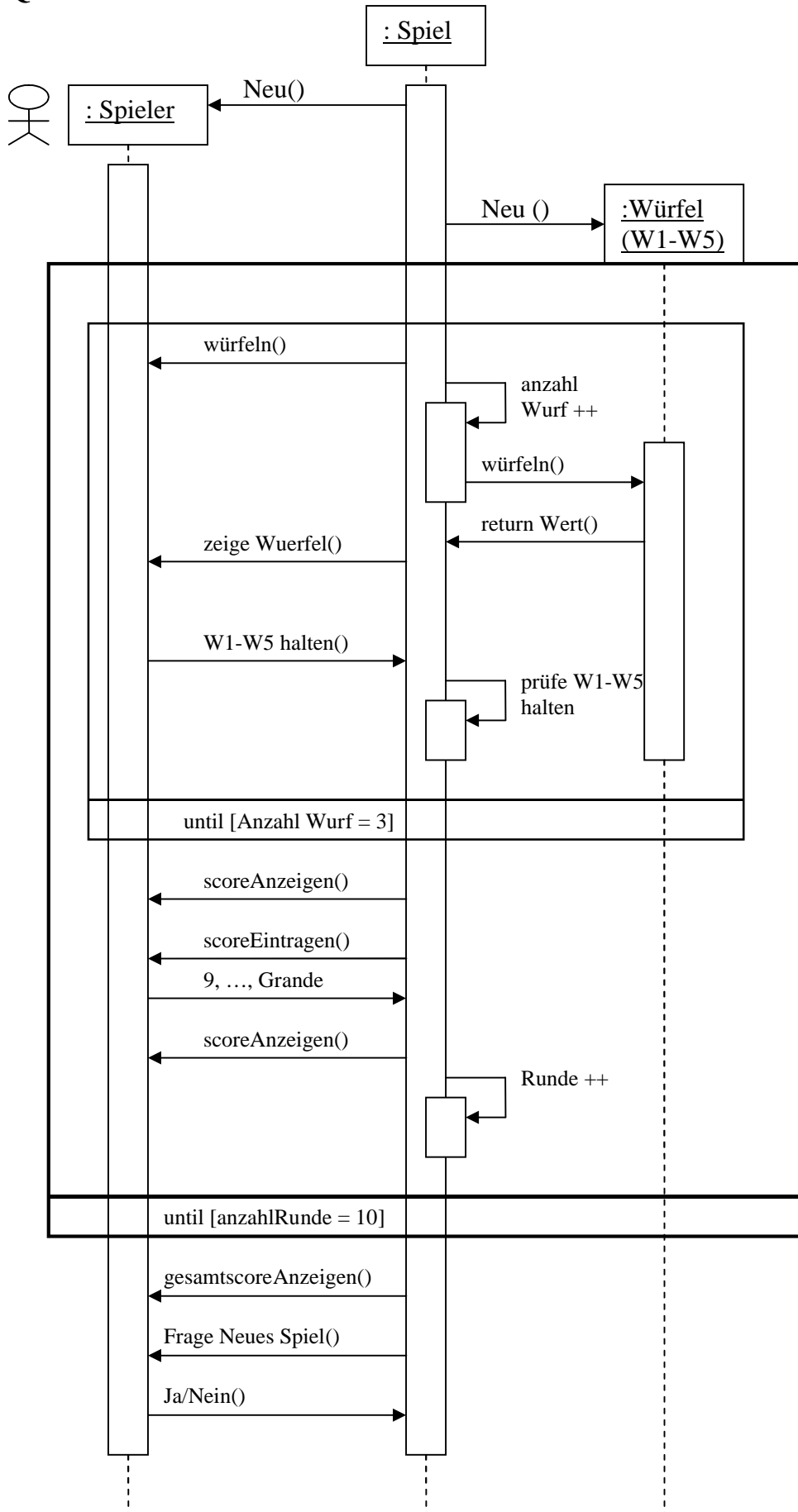
### Use Case ‚halten‘

Mit Eingabe der Tasten ‚j‘ und ‚Enter‘ bestimmt der User, ob der jeweilige Würfel auf ‚halten‘ gesetzt werden soll.

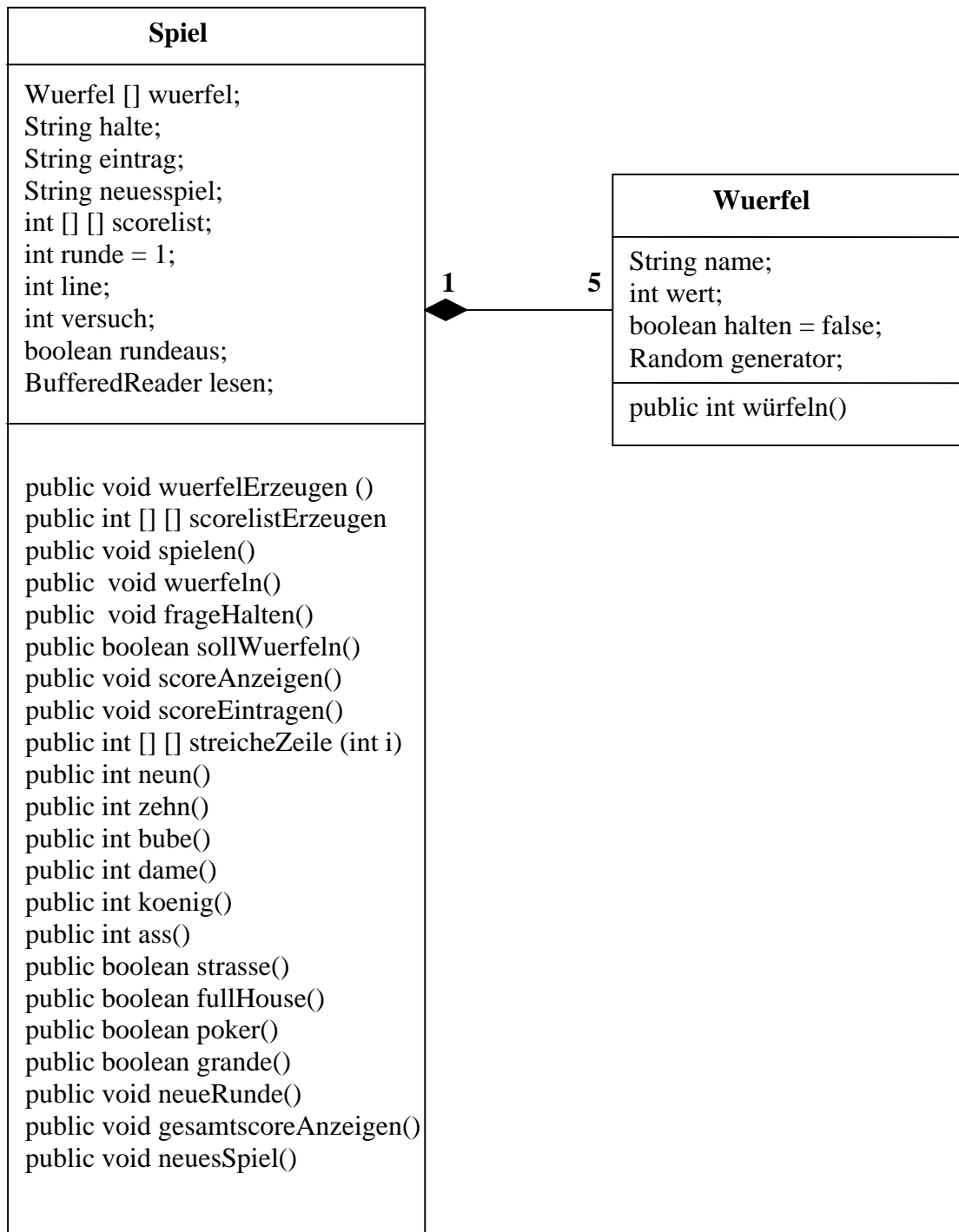
### Use Case ‚Punkte eintragen‘

Nach Abschluss einer Runde (d.h. wenn drei Mal gewürfelt wurde, oder wenn alle Würfel auf ‚halten‘ gesetzt sind) wird der User gefragt, in welche Zeile er sein Ergebnis eintragen möchte.

# SEQUENZDIAGRAMM



# KLASSENDIAGRAMM



# BESCHREIBUNG DER ATTRIBUTE UND METHODEN

## Klasse Spiel

### Attribute:

Diese Klasse Spiel enthält die Attribute *Wuerfel* (Array der Klasse Wuerfel), *halte* (String), *eintrag* (String), *neuesSpiel* (String), *scorelist* (mehrdimensionales Array), *line* (int), *runde* (int), *versuch* (int), *rundeaus* (boolean) und *lesen* (BufferedReader).

### Methoden:

*public void wuerfelErzeugen()*

Erzeugt 5 Würfel und weist jedem den Namen ‚Wuerfel‘ und die ansteigende Nummer (von 1 - 5) zu.

*public int [][] scorelistErzeugen()*

Erzeugt ein mehrdimensionales Array mit 10 Zeilen und 2 Spalten und setzt alle Werte auf ‚0‘. Die Methode gibt eine Scoreliste zurück.

*public void spielen()*

Ist so eine Art ‚Hauptmethode‘; wird am Anfang des Spieles aufgerufen und löst mit einer if-Bedingung die Methoden *sollWuerfeln()* bzw. *wuerfeln()* aus. In weiterer Folge werden hier auch die Methoden *scoreAnzeigen()*, *scoreEintragen()* und *neueRunde()* aufgerufen.

*public void wuerfeln()*

Ruft für jeden Würfel die Methode *wuerfeln()* der Klasse Wuerfel auf. Nach dem 1. und 2. Würfelversuch innerhalb einer Runde wird auch die Methode *frageHalten()* aufgerufen.

*public void frageHalten()*

Fragt für jeden Würfel, der innerhalb einer Runde noch nicht auf ‚halten‘ gesetzt wurde, ob man ihn ‚halten‘ möchte.

*public boolean sollWuerfeln()*

Überprüft, ob schon vor Ende einer Runde alle Würfel auf ‚halten‘ gesetzt sind. Gibt die Variable ‚rundeaus‘ als ‚true‘ zurück, wenn das der Fall ist.

*public void scoreAnzeigen()*

Zeigt den aktuellen Stand der Scorelist an.

*public void scoreEintragen()*

Liest die Zeile ein, in der die Punkte eingetragen werden sollen. Überprüft, ob die eingegebene Zahl zwischen 0 und 11 liegt, ob die eingegebene Zahl eine bereits ‚gestrichene Zeile‘ angibt, oder ob schon etwas in die Zeile eingetragen wurde.

Überprüft durch aufrufen der Wert-Methoden (z.B.: ‚neun()‘), ob die Werte der Würfel für die eingegebene Zeile als Scorepunkte zählen. Ruft am Ende die Methode scoreAnzeigen() auf.

*public int [] [] streicheZeile (int i)*

Setzt die zweite Spalte der jeweiligen Zeile des Scorelist-Arrays auf ‚1‘ und löst so in der Methode scoreAnzeigen() die Meldung ‚Zeile gestrichen!‘ aus.

*public int neun()*

Überprüft, wie viele Würfel dem Wert ‚Neun‘ entsprechen (d.h. dem errechneten Wert ‚1‘ der Variable ‚wert‘), berechnet die Punkte für die Scoreliste und gibt diese zurück.

*public int zehn()*

Überprüft, wie viele Würfel dem Wert ‚zehn‘ entsprechen (d.h. dem errechneten Wert ‚2‘ der Variable ‚wert‘), berechnet die Punkte die Scoreliste und gibt diese zurück.

*public int bube()*

Überprüft, wie viele Würfel dem Wert ‚Bube‘ entsprechen (d.h. dem errechneten Wert ‚3‘ der Variable ‚wert‘), berechnet die Punkte für die Scoreliste und gibt diese zurück

*public int dame()*

Überprüft, wie viele Würfel dem Wert ‚Dame‘ entsprechen (d.h. dem errechneten Wert ‚4‘ der Variable ‚wert‘), berechnet die Punkte für die Scoreliste und gibt diese zurück

*public int koenig()*

Überprüft, wie viele Würfel dem Wert ‚Koenig‘ entsprechen (d.h. dem errechneten Wert ‚5‘ der Variable ‚wert‘), berechnet die Punkte für die Scoreliste und gibt diese zurück

*public int ass()*

Überprüft, wie viele Würfel dem Wert ‚Ass‘ entsprechen (d.h. dem errechneten Wert ‚6‘ der Variable ‚wert‘), berechnet die Punkte für die Scoreliste und gibt diese zurück

*public boolean strasse()*

Überprüft, ob die Würfel dem Wert ‚Strasse‘ entsprechen und gibt bei Richtigkeit den Wert ‚true‘ zurück.

*public boolean fullHouse()*

Überprüft, ob die Würfel dem Wert ‚Full House‘ entsprechen und gibt bei Richtigkeit den Wert ‚true‘ zurück.

*public boolean poker()*

Überprüft, ob die Würfel dem Wert ‚Poker‘ entsprechen und gibt bei Richtigkeit den Wert ‚true‘ zurück.

*public boolean grande()*

Überprüft, ob die Würfel dem Wert ‚Grande‘ entsprechen und gibt bei Richtigkeit den Wert ‚true‘ zurück.

*public void neueRunde()*

Setzt für jeden Würfel die Variable ‚halten‘ auf ‚false‘, erhöht die Variable runde um 1, setzt die Variable rundeaus auf false und ruft die Methode spielen() auf.

*public void gesamtscoreAnzeigen()*

Berechnet nach 10 gespielten Runden die erreichte Gesamtpunkteanzahl und zeigt sie an.

*public void neuesSpiel()*

Fragt nach Ende der 10. Spielrunde, ob der User noch einmal spielen möchte. Setzt nach Eingabe von ‚j‘ die Variable ‚runde‘ auf ‚1‘ und ruft die Methoden wuerfelErzeugen(), scorelistErzeugen() und spielen() auf.

## **Klasse Wuerfel**

### Attribute:

Die Klasse Wuerfel enthält die Attribute *wert* (int), *name* (String), *halten* (boolean, auf ‚false‘ gesetzt) und *static Random generator*.

## Methoden:

*public int wuerfeln (String x)*

Überprüft für alle 5 Wuerfel, ob die Variable ‚halten‘ auf ‚false‘ gesetzt ist; falls ja, wird für den jeweiligen Wuerfel ein neuer Wert zwischen 1 und 6 berechnet, falls nicht wird der Wuerfel als ‚gehalten‘ dargestellt.

## **WARTUNG**

Eine Wartung im eigentlichen Sinn ist für das Programm grundsätzlich nicht notwendig.

Während des Testbetriebes wurden jedoch noch Beeinträchtigungen festgestellt, die im Zuge einer Weiterentwicklung behoben werden können:

So kommt es bei der Eingabe in die Scorelist zu einer IOException, wenn nicht eine Zahl, sondern ein Buchstabe oder anderes Symbol eingegeben wird. Dieses Problem muss im Zuge der nächsten Wartung bearbeitet werden.

Ebenso ist es denkbar, das Programm um eine Highscore-Liste zu erweitern.

Zusätzlich ist bei reibungslosem Programmdurchlauf in der Endversion der Ausbau als Applet angedacht.

## **HINWEISE FÜR DEN USER**

Zum Spielen Quellcode (Wuerfelpoker.java) downloaden und über die Windows-Eingabeaufforderung starten –

→ *java Wuerfelpoker*

### Spielanleitung:

- Halten des jeweiligen Würfels durch Eingabe von ‚j‘ und ‚Enter‘
- Score eintragen in die gewünschte Zeile erfolgt durch Eingabe der ‚Zeilennummer‘ (1,..., 10) und ‚Enter‘
- Neues Spiel starten am Ende des Spiels durch Eingabe von ‚j‘ und ‚Enter‘