



1. Zulassungsklausur zur Veranstaltung
Objektorientierte Programmierung

im Sommersemester 2023

Prüfer: Prof. Dr. Matthias Tichy

Fakultät Ingenieurwissenschaften, Informatik, Psychologie

24.06.2023, 11-14 Uhr

Bearbeitungszeit: 45 min

Nachname:	Vorname:	Matrikelnummer:
Studiengang und Abschluss:		Account:
<p>Hiermit erkläre ich, dass ich prüfungsfähig bin. Sollte ich nicht auf der Liste der angemeldeten Studierenden aufgeführt sein, dann nehme ich hiermit zur Kenntnis, dass diese Prüfung nicht gewertet werden wird.</p> <p>_____</p> <p>Datum, Unterschrift des Prüfungsteilnehmers</p>		

Zur allgemeinen Beachtung:

- Füllen Sie das Deckblatt vollständig und korrekt aus.
- Lesen Sie sich zunächst die Klausur sorgfältig durch (sie besteht aus 3 Seiten).
- Zum Erreichen der Vorleistung benötigen Sie **15** der insgesamt **31** Punkte.
- Im Ordner `~/materials` finden Sie das zu bearbeitende Projekt sowohl als ZIP als auch entpackt.
- Sie dürfen im Projekt beliebig Hilfsmethoden und Klassen hinzufügen. Die Signaturen der vorgegebenen Methoden und Interfaces dürfen nicht verändert werden. Die schon implementierten Methoden dürfen nicht geändert werden.
- **Vor Beginn der Bearbeitungszeit** haben Sie 5 Minuten Zeit, um einen Editor/eine IDE Ihrer Wahl zu starten und sich in den Materialien zurecht zu finden.
- **Nach Ablauf der Bearbeitungszeit** haben Sie 5 Minuten Zeit, um Ihre Lösung im Ordner `~/export` abzulegen. **Bleiben Sie sitzen!**
- Abgaben sind als Projektexport in Zip-Form abzugeben. Achten Sie darauf Ihr **gesamtes** Projekt, mit allen zugehörigen Ordnern, zu exportieren.
- Abgaben die Compilerfehler produzieren werden nicht bewertet.

Punkteverteilung				
1	2	3	Σ	Note
von 7	von 9	von 15	von 31	
				Korrektur

Das Projekt

Das Ihnen zur Verfügung gestellte Projekt soll das Spiel Tic-Tac-Toe implementieren. Bei diesem Spiel treten zwei Spieler gegen einander an. Die Spieler setzen abwechselnd durch Eingabe von x- und y-Koordinaten ihre Markierung auf ein 3x3 Spielfeld. Der Spieler, der es zuerst schafft, eine horizontale, vertikale oder diagonale Line mit seinen Symbolen zu bilden, hat eine Runde gewonnen. Es werden maximal fünf Runden gespielt. Hat ein Spieler bereits dreimal gewonnen, gilt das Spiel als beendet.

Das Spielfeld ist ein 3x3 Array, das in (x,y) Koordinaten aufgeteilt ist, wie in der folgenden Abbildung zu sehen:

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(3,2)	(3,3)

Während eines Spiels werden die Koordinaten durch die einzelnen Spieler in die Konsole eingegeben und haben dabei folgendes Format: x-y.

Die folgende Abbildung zeigt ein Spiel zwischen Bob und Alice. Bob hat das Symbol X und Alice hat das Symbol O. Alice ist am Zug und gibt 3-1 als Koordinaten in die Konsole ein.

O	X	O
	O	X
		X

Dadurch ergibt sich der folgende neue Zustand für das Spielfeld:

O	X	O
	O	X
O		X

Da Alice nun über die Diagonale mit den Koordinaten (1,3), (2,2) und (3,1) ihr Symbol platzieren konnte, hat sie diese Runde gewonnen.

Das Projekt besteht aus den folgenden 4 Klassen. Im Folgenden finden Sie eine kurze Übersicht über die einzelnen Klassen:

oop.sose2023.admission_exam.Main Diese Klasse initialisiert ein neues Spiel zwischen den Spielern Bob und Alice, und enthält die Logik, dass ein Spiel maximal 5 Runden dauert, abgesehen davon, wenn ein Spieler bereits 3 mal gewonnen hat.

oop.sose2023.admission_exam.group01.Game Diese Klasse implementiert den Spielzustand und die Spiellogik.

oop.sose2023.admission_exam.group01.Player Diese Klasse implementiert den Zustand der einzelnen Spieler.

oop.sose2023.admission_exam.group01.util.InputManagement Diese Klasse implementiert die Logik, um Eingaben über die Konsole als Koordinaten zu interpretieren.

Aufgabe 1 - Spielerinformationen

7 Punkte

Implementieren Sie die Klasse `Player`, die den Zustand eines Spielers verwaltet. Abbildung 1 zeigt das Klassendiagramm für die zu implementierende Klasse `Player`. Ihre Implementierung soll nur die dargestellten Attribute und Methoden implementieren. Achten Sie auf die Schreibweise der Namen, die angegebenen Sichtbarkeiten und die festgelegten Typen. Übernehmen Sie auch die im Klassendiagramm vorgegebenen Sichtbarkeiten: **Private** Attribute oder Methoden werden mit einem vorangestellten `-` gekennzeichnet. **Öffentliche** Attribute oder Methoden werden mit einem vorangestellten `+` gekennzeichnet. Die Klasse soll nur einen Konstruktor zur Verfügung stellen. Dieser soll den Namen (`name`) eines Spielers und dessen Symbol (`mark`) als Parameter übergeben bekommen und die gleichnamigen Attribute in der Klasse setzen. Zudem soll der Konstruktor die Anzahl der gewonnen Runden (`wins`) auf 0 setzen. Die Methode `incWins()` soll den Wert im Attribut `wins` um 1 erhöhen. Methoden, die mit dem Präfix `get` beginnen, sollen jeweils den Wert des gleichnamigen Attributes zurückgeben. Methoden, die mit dem Präfix `set` beginnen, sollen jeweils das gleichnamige Attribut auf den übergebenen Parameterwert setzen.

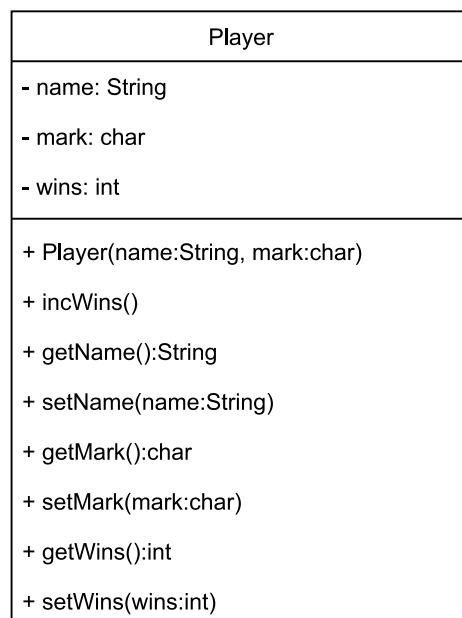
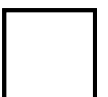


Abb. 1: Klassendiagramm zur Verwaltung von Spielerinformationen

Aufgabe 2 - Spiellogik

9 Punkte

Implementieren Sie in der Klasse `Main` die Methode `runGame(Game game)`. Diese Methode realisiert den Prozess, der hinter den einzelnen Spielzügen steckt. Dies bedeutet, dass solange kein Gewinner feststeht weiterhin die Spieler neue Koordinaten für ihre Züge eingeben, das entsprechende Symbol auf dem Spielfeld gesetzt wird und anschließend der Spielzustand evaluiert wird. Verwenden Sie die Methoden der Klasse `Game` und beachten Sie die JavaDoc Kommentare in der Klasse `Game`. Beachten Sie, dass solange das Attribut `winner`, eines `Game`-Objekts, den Wert `-1` besitzt, gibt es noch freie Felder auf dem Spielfeld und noch keinen Gewinner.



Aufgabe 3 - Gewinner prüfen**15 Punkte**

Implementieren Sie in der Klasse `Game` die Methode `checkWinner()`. Diese Methode prüft, ob der Spieler des aktuellen Spielzugs gewonnen hat oder nicht. Beachten Sie, dass ein Spieler gewonnen hat, wenn 3 seiner Symbole eine horizontale, vertikale oder diagonale Linie auf dem Spielfeld bilden. Zum Beispiel, zeigt die folgende Abbildung den Zustand des Spielfelds, wenn der Spieler mit dem Symbol X gewonnen hat:

O		X
	X	
X		O

Hat der aktuelle Spieler gewonnen, soll die Methode `true` zurück liefern, sonst `false`. Beachten Sie, dass die Spieler in dem Attribut `players`, der Klasse `Game`, verwaltet werden und der Index des Spielers des aktuellen Spielzugs im Attribut `currentPlayer` verwaltet wird.

