


Zu erledigen: Eine Bewertung erhalten

Zu erledigen: Bestehensnote erreichen

 Beschreibung

 [Bearbeiten](#)

 [Abgabesicht](#)

 **Abgabetermin:** Sonntag, 14. Januar 2024, 23:59

 **Maximale Anzahl an Dateien:** 1

Arbeitstyp:  Einzelarbeit

Lernziele

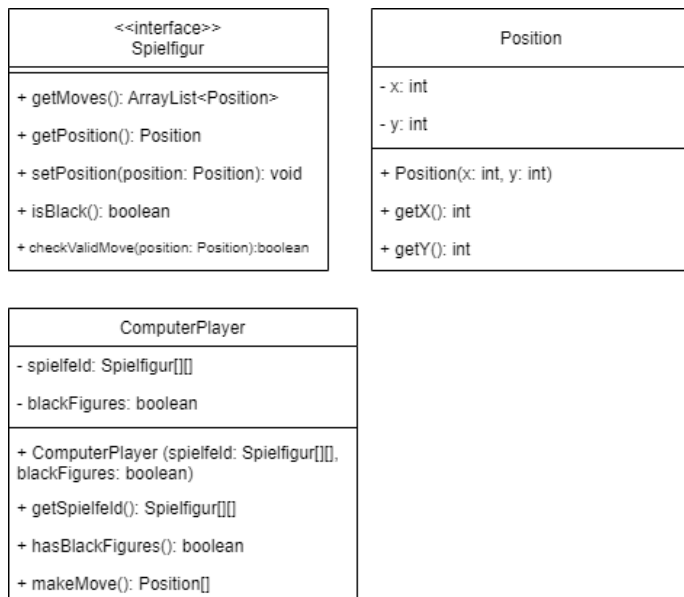
Mit dieser Aufgabe lernen Sie

- Den Umgang mit UML-Diagrammen
- Das Nutzen von Interfaces

Aufgabenstellung

In dieser Aufgabe sollen Sie einen Computer-Spieler erstellen, der automatisiert gültige Spielzüge auswählen kann. Dazu stehen Ihnen sämtliche Spielfiguren zur Verfügung, welche nach dem Schema aus dem UML Diagramm aufgebaut sind (wie in Aufgabe 5.1).

Sie müssen nur die Klasse `ComputerPlayer` erstellen.



Implementieren Sie die Klasse `ComputerPlayer` nach den Vorgaben aus dem UML-Diagramm.



Die Methoden sollen dabei folgende Funktionalitäten haben:

- Die Getter geben den entsprechenden Wert zurück. (Die Methode `hasBlackFigures()` stellt einen Getter für `blackFigures` dar.)
- `makeMove()` wählt einen gültigen Spielzug aus. Dabei soll, wenn möglich, eine möglichst wertvolle Figur geschlagen werden (siehe Ende Aufgabenstellung). Wenn keine Figur geschlagen werden kann, darf ein beliebiger gültiger Spielzug ausgewählt werden. Um einen Spielzug auszuwählen, geben Sie ein `Position[]` der Länge 2 zurück. Schreiben Sie an die erste Stelle des Arrays die Ausgangsposition und in die zweite Stelle die Endposition des Zuges. Verschieben Sie keine Figuren auf dem Spielfeld!

Figuren sollen nach folgender Priorisierung geschlagen werden:

Koenig > Dame > Turm > Springer > Laeufer > Bauer. Wenn mehrere Figuren der gleichen Priorität geschlagen werden können, darf eine beliebige geschlagen werden.

Bearbeitungshinweise

- Schreiben Sie die Klassen, welche das beschriebene Verhalten aufweisen. Sie müssen keine main-Methode schreiben.
- Denken Sie daran, Ihre Lösung zu  speichern und  abzugeben, damit Ihre Lösung gewertet wird.

?

- Die Methoden der Klassen, welche das `Spielfigur` Interface implementiert haben, sollten Ihnen bereits aus der letzten Aufgabe vertraut sein.
Tipp: Die Methode `getMoves()` liefert Ihnen eine `ArrayList` mit allen gültigen Zugpositionen einer Figur.
- Der `instanceof` Operator kann bei dieser Aufgabe sehr nützlich sein.
Der boolsche Ausdruck `(x instanceof Bauer)` liefert `true` zurück, wenn es sich bei `x` um eine Instanz der Klasse `Bauer` handelt, `false` wenn nicht.
- Sollten Sie Fragen zu Schach im Umfang der Vorbereitungsaufgaben haben, lesen Sie im [Schach-Guide](#) des Kurses nach oder Fragen Sie Ihre Tutoren.