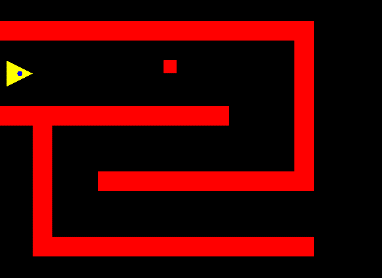
**Praxistest 10d am 16.2.23 Gruppe A Name \_\_\_\_\_\_\_\_\_\_\_**

Vorbereitung:

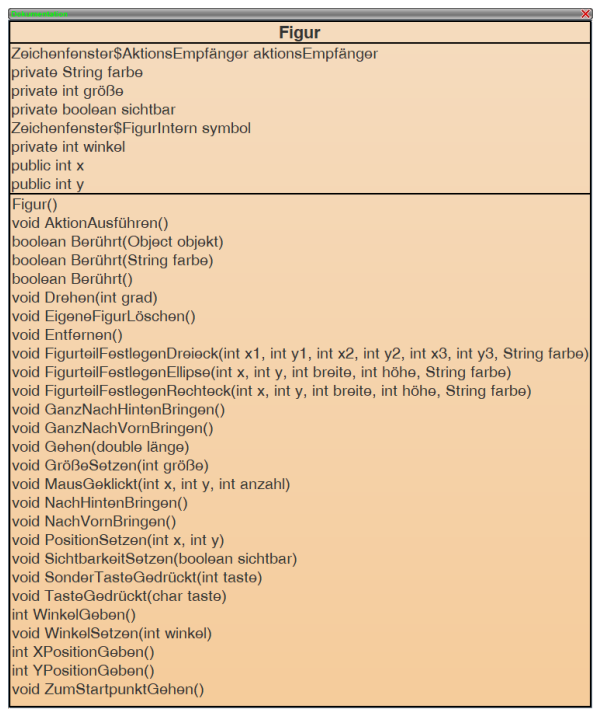
* Logge dich am Computer und in Mebis ein. Lade die Vorlage herunter, entpacke sie und öffne das Projekt in BlueJ. Speichere das Projekt **SOFORT** unter einem Namen, der mit deinem Nachnamen beginnt, z.B. Meier\_Praxistest.

Hinweise:

* Als Hilfsmittel ist ausschließlich das Fehlerlexikon entsprechend der Vorgaben erlaubt.
* Die Teilaufgaben a) bis f) beziehen sich auf die Klasse RAUMSCHIFF. **In anderen Klassen darf nichts geändert werden.**
* Für jeden Test muss zuerst ein Objekt der Klasse SPIEL und dann ein Objekt der Klasse RAUMSCHIFF erstellt werden.
* Als Unterstützung erhältst du das Klassendiagramm von Figur (s. u.).

**Gefährliche Fahrt**

Ziel der Aufgabe ist es ein Spiel zu programmieren, in welchem der Spieler das Raumschiff durch ein kleines Labyrinth steuern muss. Für jede Berüh­rung mit einer „Feuerwand“ nutzt sich der Schutzschild ab, bis das Raumschiff nicht mehr navigierbar ist ...

1. Deklariere ein Attribut navigator, welches einen Namen und ein Attribut valueShield, welches den Zustand des Schutzschildes als Zahl speichert.
2. Initialisiere das Attribut navigator auf deinen Vornamen und das andere Attribut auf den Startwert 10. Positioniere das Raumschiff am Startpunkt (grüne Markierung).
3. Implementiere zum Attribut navigator eine Setzen-Methode, damit ein „Wechsel“ des Steuermanns möglich ist.
4. Überschreibe die Methode AktionAusführen so, dass sich bei jedem Taktschlag das Raumschiff um 10 Pixel bewegt. Weiterhin soll überprüft werden, ob das Raumschiff eine Feuer­wand berührt . Ist das der Fall, so reduziert sich die Stärke des Schutzschildes um 1.
5. Implementiere für die Bewegung des Raumschiffs eine Tastatursteuerung. Details kannst du selbst festlegen, du solltest aber auf eine Nutzer­freundlich­keit achten.

Falls für dich nötig: Die Nummern der Cursortasten (linke Pfeiltaste ist 37, Pfeiltaste nach oben 38, Pfeiltaste nach unten 40, rechte Pfeiltaste 39). Letzteres kann, aber muss nicht eingesetzt werden.

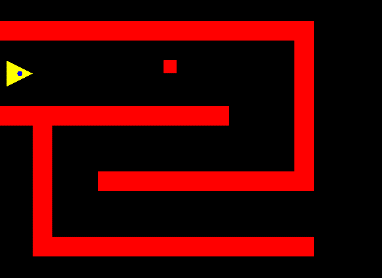
1. Nach jeder Berührung mit der Feuerwand soll der Status des Schutzschildes als Text ausgegeben werden: Einerseits in Form von Sternen, wobei die Sternanzahl dem Wert des Attributs valueShield entspricht, andererseits auch als Zahl, z. B. \*\*\*\*\*5 falls der entsprechende Attributwert 5 ist. Implementiere dies. Hinweis: Man kann System.out.print("…") verwenden, damit ein Ausdruck ohne Zeilenumbruch erfolgt.
2. Ergänze eine eigene (, kleine) Erweiterung. Notiere deine Idee in einem Satz auf der Rückseite des Angabenblatts, damit ich Sie schnell finden kann. Bewertet werden nur die Implementierung im Projekt Brz

**Praxistest 10d am 16.2.23 Gruppe B Name \_\_\_\_\_\_\_\_\_\_\_**

Vorbereitung:

* Logge dich am Computer und in Mebis ein. Lade die Vorlage herunter, entpacke sie und öffne das Projekt in BlueJ. Speichere das Projekt **SOFORT** unter einem Namen, der mit deinem Nachnamen beginnt, z.B. Meier\_Praxistest.

Hinweise:

* Als Hilfsmittel ist ausschließlich das Fehlerlexikon entsprechend der Vorgaben erlaubt.
* Die Teilaufgaben a) bis f) beziehen sich auf die Klasse RAUMSCHIFF. **In anderen Klassen darf nichts geändert werden.**
* Für jeden Test muss zuerst ein Objekt der Klasse SPIEL und dann ein Objekt der Klasse RAUMSCHIFF erstellt werden.
* Als Unterstützung erhältst du das Klassendiagramm von Figur (s. u.).

**Gefährliche Fahrt**

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungZiel der Aufgabe ist es ein Spiel zu programmieren, in welchem der Spieler das Raumschiff durch ein kleines Labyrinth steuern muss. Für jede Berüh­rung mit einer „Feuerwand“ nutzt sich der Schutzschild ab, bis das Raumschiff nicht mehr navigierbar ist ...

1. Deklariere ein Attribut driver, welches einen Namen und ein Attribut stärkeSchutzschild, welches auf den Zustand des Schutzschildes als Zahl speichert.
2. Initialisiere das Attribut driver auf deinen Vornamen und das andere Attribut auf den Startwert 12. Positioniere das Raumschiff am Startpunkt (grüne Markierung).
3. Implementiere zum Attribut driver eine Setzen-Methode, damit ein „Wechsel“ des Fahrers möglich ist.
4. Überschreibe die Methode AktionAusführen so, dass sich bei jedem Taktschlag das Raumschiff um 10 Pixel bewegt. Weiterhin soll überprüft werden, ob das Raumschiff eine Feuer­wand berührt . Ist das der Fall, so reduziert sich die Stärke des Schutzschildes um 1.
5. Implementiere für die Bewegung des Raumschiffs eine Tastatursteuerung. Details kannst du selbst festlegen, du solltest aber auf eine Nutzer­freundlich­keit achten.

Falls für dich nötig: Die Nummern der Cursortasten (linke Pfeiltaste ist 37, Pfeiltaste nach oben 38, Pfeiltaste nach unten 40, rechte Pfeiltaste 39). Letzteres kann, aber muss nicht eingesetzt werden.

1. Nach jeder Berührung mit der Feuerwand soll der Status des Schutzschildes als Text ausgegeben ausgegeben werden: Einerseits in Form von Sternen, wobei die Sternanzahl dem Attributwert entspricht, andererseits auch als Zahl, z. B. \*\*\*\*\*5 falls der entsprechende Attributwert 5 ist. Implementiere dies. Hinweis: Man kann System.out.print("…") verwenden, damit ein Ausdruck ohne Zeilenumbruch erfolgt.
2. Ergänze eine eigene (, kleine) Erweiterung. Notiere deine Idee in einem Satz auf der Rückseite des Angabenblatts, damit ich Sie schnell finden kann. Bewertet werden nur die Implementierung im Projekt Brz

**Praxistest 10d am 15.02.2023**

Name: Muster Marina Note: \_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| **Geisterjagd – Gruppe A** |  |  |
| a) Attributdeklaration  navigator 0,5  valueShield 0,5 | \_/ 1 | Kein Abzug bei zusätzlichen Attributen |
| b) Konstruktor  Methodenkopf 1  Initialisierung Attr. 2  Startposition (30/150) 1  super() 1 | \_/ 5 | Kein Abzug bei Parameter |
| c) Setter NavigatorSetzen  Methodenkopf 1,5  Methodenrumpf 1,5 | \_/ 3 | Kein Abzug, aber .Anmerkung bei Methodenname ohne Geben  Schlüsselwort + richtiges Attribut |
| d) AktionAusführen  Überschreiben 1  Methodenkopf 1  Gehen(10) 1  bed. Anweisung 2 Erniedringung   valueShield 1 | \_/ 6 | Falls Gehen() nur in einem Fall ausgeführt wird -1 BE |
| e) Tastatursteuerung  Überschreiben 1  Methodenkopf 1  Mehrfachauswahl 2  Drehung 2 | \_/ 6 | Bed. Anweisung bzw. Mehrfachauswahl 2BE  Sinnvolle Drehung 2BE -0,5 BE falls Bewegung um 20 Pixel |
| f) Text anzeigen  Bedingte Wdh. mit  Ausgabe der Sterne 3  Ausgabe der Zahl 1  Abzug falls nicht in   AktionAusführen -0,5 | \_\_/4 | Zuweisung statt Vergleich -> -1BE |
| g) Erweiterung 4 | \_/4 | Je nach Umsetzungsgrad und Idee reine Idee auf Papier 0BE |

Summe: \_\_\_/29

Ggf. Abzüge, wenn in falscher Klasse programmiert wurde oder die Struktur der Klasse grob nicht in Ordnung ist

**Praxistest 10d am 15.02.2023**

Name: Muster Marina Note: \_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| **Geisterjagd – Gruppe B** |  |  |
| a) Attributdeklaration  driver 0,5  stärkeSchutzschild 0,5 | \_/ 1 |  |
| b) Konstruktor  Methodenkopf 1  Initialisierung Attr. 2  Startposition (30/150) 1  super() 1 | \_/ 5 |  |
| c) Setter DriverSetzen  Methodenkopf 1,5  Methodenrumpf 1,5 | \_/ 3 |  |
| d) AktionAusführen  Überschreiben 1  Methodenkopf 1  Gehen(10) 1  bed. Anweisung 2 Erniedringung   valueShield 1 | \_/ 6 |  |
| e) Tastatursteuerung  Überschreiben 1  Methodenkopf 1  Mehrfachauswahl 2  Drehung 2 | \_/ 6 |  |
| f) Text anzeigen  Bedingte Wdh. mit  Ausgabe der Sterne 3  Ausgabe der Zahl 1  Abzug falls nicht in   AktionAusführen -0,5 | \_\_/4 |  |
| g) Erweiterung 4 | \_/4 |  |

Summe: \_\_\_/29