Die Hausaufgaben müssen von jedem Studierenden einzeln bearbeitet und abgegeben werden. Für die Hausaufgabe sind die aktuellen Informationen vom Blog https://seblog.cs.uni-kassel.de/ws2324/programming-and-modelling/ zu berücksichtigen.

Abgabefrist ist der 09.11.2023 - 23:59 Uhr

Abgabe

Wir benutzen für die Abgabe der Hausaufgaben Git. Jedes Repository ist nur für den Studierenden selbst sowie für die Betreuer und Korrektoren sichtbar.

Für die Hausaufgabe benötigst du **ein neues** Repository.

Dieses kann über folgenden Link erstellt werden, falls nicht bereits geschehen:

https://classroom.github.com/a/unUTIqF8

Nicht oder zu spät gepushte (Teil-)Abgaben werden mit 0 Punkten bewertet!

Zur Erstellung der geforderten Diagramme kann die Webanwendung "Diagrams.net" verwendet werden:

https://www.diagrams.net

Handschriftliche Abgaben werden nicht gewertet.

Die Hausaufgaben müssen als PDF-Datei (.pdf) abgegeben werden. Jedes Diagramm ist in einer eigenen Datei abzulegen. Jede PDF-Datei soll genau eine Seite haben.

Anmeldung

Auf dem letzten Hausaufgabenblatt ist uns ein Fehler unterlaufen. Der Link zur Anmeldung war dort falsch.

Falls du dich noch nicht in dem Google Form (https://forms.gle/QJVbzjwigJ46T6qw7) angemeldet hast, hole dies jetzt unbedingt nach. Deine Hausaufgaben können ohne Anmeldung nicht bewertet werden!

Aufgabe 1 - Objektdiagramme von Szenarien herleiten (55P)

Leite für die drei folgenden textuellen Szenarien zum Spiel „Risiko" passende Objektdiagramme ab. Erstelle dazu jeweils ein Objektdiagramm zur Start- und Endsituation. Für jedes der 3 Szenarien müssen somit zwei Diagramme entstehen. Benenne die Dateien eindeutig (beispielsweise "<Szenariotitel><Start | End>").

Title: Moving Troops

Start: Tom is playing „Risk" against Sandro. Tom has occupied territory Great Britain with two troops. Sandro has occupied territory Scandinavia with one troop. Great Britain and Scandinavia are connected. Great Britain is also connected to territory Western Europe. It's Tom's turn in the moving phase.

Action: Tom moves one troop from Great Britain to Western Europe.

End: Tom has occupied Great Britain with one troop and Western Europe with one troop. Sandro has occupied Scandinavia with one troop. It's Sandro's turn in the reinforcement phase.

Title: Reinforcement

Start: Tom is playing „Risk" against Sandro. Tom has occupied territory Peru with one troop. Sandro has occupied territory Alaska with one troop. Territory Indonesia has no troops yet. It's Sandro's turn in the reinforcement phase.

Action: Sandro is allowed to place one additional troop. He places the troop in territory Indonesia.

End: Tom has occupied Peru with one troop. Sandro has occupied Alaska with one troop and Indonesia with one troop. It's Sandro's turn in the attacking phase.

Title: Winning

Start: Tom is playing „Risk" against Sandro. Tom has occupied territory Alberta with three troops. Sandro has occupied territory Alaska with one troop. Alaska is connected to Alberta. It's Toms turn in the attacking phase.

Action: Tom attacks Sandro's troops in Alaska with his troops in Alberta. They roll the dice. Tom rolled a higher number and wins the attack, so Sandro's troop has been defeated.

End: Sandro has no more troops in any territory. Tom won the game.

Lege die **6** erstellten Dateien in einem Ordner mit dem Namen "task1" in deinem Repository ab. Committe und pushe die Änderung abschließend auf den main-Branch.

Bei der Bewertung wird vor allem auf die vorgestellten Konventionen der Diagrammtypen geachtet.

Achte darauf, das Repository der aktuellen Hausaufgabe zu verwenden.

Aufgabe 2 - Objektdiagramme von Code herleiten (30P)

Leite für den Code des folgenden Test-Szenarios passende Objektdiagramme ab. Erstelle dazu jeweils ein Objektdiagramm zur Start- und Endsituation. Somit müssen zwei Diagramme entstehen. Benenne die Dateien eindeutig (”TicTacToe<Start | End>”).

Hinweis: Das Listing erstreckt sich über zwei Seiten.

```
1   @Test
2   void myTestScenario() {
3
4       // create player one
5       Player playerAlice = new Player()
6               .setName("Alice")
7               .setScore(250);
8
9       // create player two
10      Player playerBob = new Player()
11              .setName("Bob")
12              .setScore(100);
13
14      // create the fields
15      //      (for now just a small part of the game board...)
16      Field f1 = new Field().setCoordinate("f1");
17      Field f2 = new Field().setCoordinate("f2");
18      Field f3 = new Field().setCoordinate("f3");
19
20      // "left" and "right" are bidirectional!
21      f1.setRight(f2);
22      f2.setLeft(f1);
23
24      f2.setRight(f3);
25      f3.setLeft(f2);
26
27      // create current game situation:
28      //      some markers are already on the board
29
30      // "owner" and "markers" are bidirectional!
31      Marker markerB = new Marker().setOwner(playerBob);
32      playerBob.withMarkers(markerB);
33
34      Marker markerB2 = new Marker().setOwner(playerBob);
35      playerBob.withMarkers(markerB2);
36
37      // "marker" and "position" are bidirectional!
38      f1.setMarker(markerB);
39      markerB.setPosition(f1);
40
41      f2.setMarker(markerB2);
42      markerB2.setPosition(f2);
43
44      // TODO: now draw object diagram for start situation
45
46      // action: place a third symbol on the game board
47      Marker markerB3 = new Marker().setOwner(playerBob);
48      playerBob.withMarkers(markerB3);
```

```
49
50          f3.setMarker(markerB3);
51          markerB3.setPosition(f3);
52
53          // result:
54          // bob has three in a row and wins
55          playerBob.setIsWinner(true);
56
57          // he also earns some points
58          playerBob.setScore(playerBob.getScore() + 40);
59
60          // TODO: now draw object diagram for end situation
61      }
```

Listing 1: TicTacToe-Test

Lege die **2** erstellten Dateien in einem Ordner mit dem Namen "task2" in deinem Repository ab. Committe und pushe die Änderung abschließend auf den main-Branch.

Bei der Bewertung wird vor allem auf die vorgestellten Konventionen der Diagrammtypen geachtet.

Achte darauf, das Repository der aktuellen Hausaufgabe zu verwenden.

Anhang

Es folgt eine Auflistung hilfreicher Webseiten und weiterer Erklärungen zu den Themen dieser Hausaufgabe. Die Links sind als Startpunkt zur selbstständigen Recherche angedacht. Das Durcharbeiten der folgenden Quellen ist kein bewerteter Anteil der Hausaufgaben.

UML

- Was ist UML? https://www.uml.org/what-is-uml.htm
- UML (Spezifikation), nur damit ihr mal gesehen habt, wie so etwas aussieht ;) https://www.omg.org/spec/UML/2.5.1/PDF
- Objektdiagramme: https://mbse.se-rwth.de/book1/index.php?c=chapter4

GitHub Desktop

- Download: https://desktop.github.com/

IntelliJ IDEA

Ab Hausaufgabe 3 arbeiten wir mit IntelliJ IDEA. Um euch auf die nächste Übung vorzubereiten, könnt ihr euch das Programm schon vorher installieren. Bei IntelliJ wird zwischen der kostenlosen „Community“ und der kostenpflichtigen „Ultimate“-Version unterschieden. Es ist für Studierende möglich die „Ultimate“-Version kostenlos zu erhalten, dies ist die „Free Educational License“. Für diese Veranstaltung genügt die kostenlose Version.

- Download: https://www.jetbrains.com/idea/download/
- Free Educational Licenses: https://www.jetbrains.com/community/education/#students
- Unterschiede der Versionen: https://www.jetbrains.com/products/compare/?product=idea&product=idea-ce