Übungen zur Vorlesung Programmieren 2
FHWS – Fakultät für Informatik und Wirtschaftsinformatik

Übungsblatt 13

Aufgabe 1: (Wiederholung Schleifen, mehrdimensionale Arrays)
(Verwenden Sie die vorgegebenen Klassen aus dem schach-Package von der e-learning Plattform!)

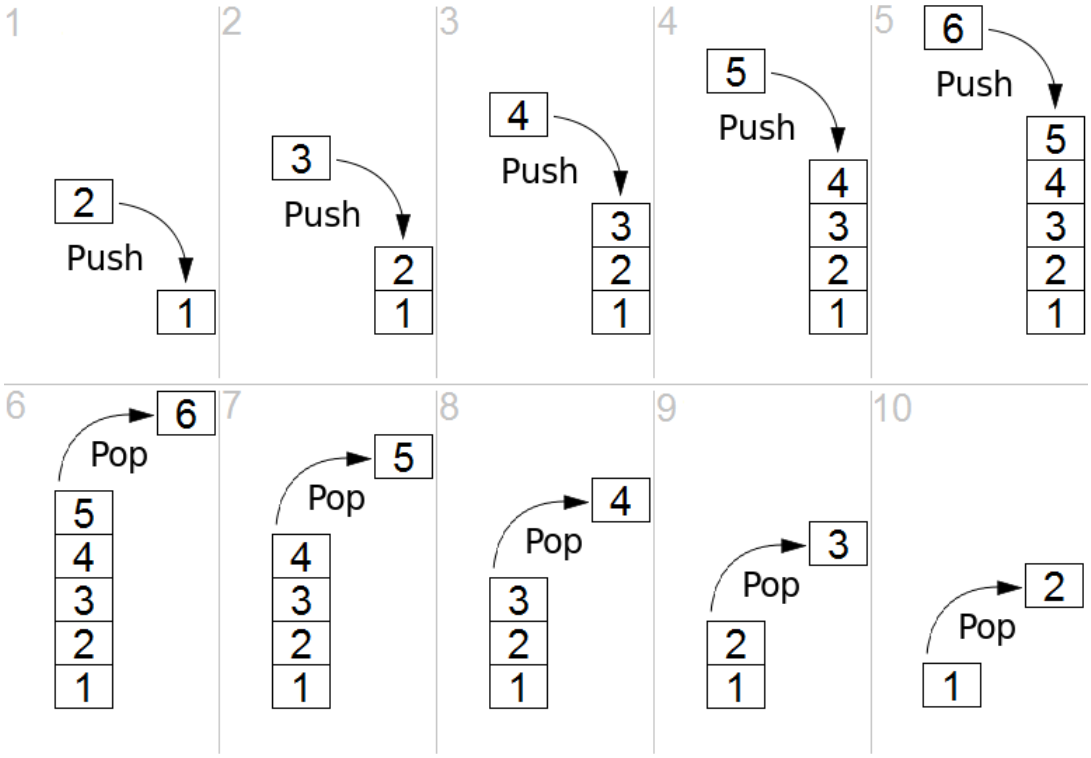
a) Ergänzen Sie in der Klasse `LaeuferImpl` den Algorithmus zur Festlegung der erlaubten Felder.
b) Ergänzen Sie die Klasse `Brett` um eine Methode `kombiniere`. Die Methode soll ein `Brett` entgegennehmen und das aktuelle `Brett` mit dem übergebenen `Brett` zu einem neuen `Brett` verbinden und dieses zurückgibt. Auf dem neuen Brett sollen alle Felder markiert sein, die auf einem der beiden oder beiden Brettern markiert waren.

Aufgabe 2: (Vererbung für Code-Wiederverwendung, Object als Oberklasse)
(Verwenden Sie die vorgegebene Klasse aus dem stack-Package von der e-learning Plattform!)

Ein Stack ist eine Datenstruktur, die zwei Primitive zur Verfügung stellt:

- push: legt ein Element oben auf den Stack
- pop: nimmt ein Element von oben vom Stack

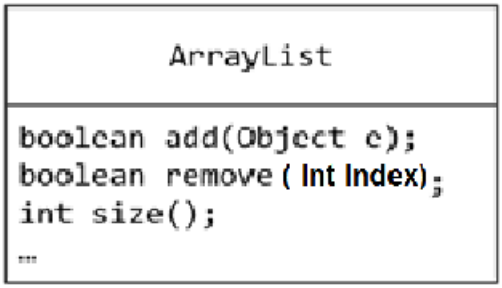
Folgende Abbildung beschreibt die Funktionsweise des Stacks:



Quelle: https://en.wikipedia.org/wiki/Stack_(abstract_data_type)#/media/File:Lifo_stack.png

a) Implementieren Sie eine Klasse `Stack`. Die Klasse `Stack` soll von der Klasse `ArrayList` erben und mit Hilfe der Methoden von `ArrayList`, die Methoden `pop` und `push` umsetzen. Auf den `Stack` sollen beliebige Objekte gelegt werden können.

   Die Klasse `ArrayList` stellt folgende Methoden zur Verfügung:



   Aus dem Javadoc:
   - `add`: Appends the specified element to the end of this list.
   - `remove`: Removes the element at the specified position in this list. Shifts any subsequent elements to the left (subtracts one from their indices).
   - `size`: Returns the number of elements in this list.

b) Welche Nachteile hat die Verwendung der Vererbung in dieser Aufgabe?

Aufgabe 3: (Komposition für Code-Wiederverwendung, Interfaces, Abstrakte Klasse)

a) Implementieren Sie den `Stack` aus Aufgabe 2 so, dass er nicht von `ArrayList` erbt, sondern `ArrayList` als Attribut verwendet wird!
b) Für die Klasse `Stack` sind also verschiedene Implementierungen denkbar. Definieren Sie ein sinnvolles Interface `Stack` und lassen Sie Ihre Klasse das Interface `Stack` implementieren.
c) Ist es sinnvoll, eine abstrakte Klasse `AbstractStack` zu implementieren, die bestimmte Teile der Implementierung für mögliche weitere Implementierungen vorgibt?

Aufgabe 4: (Mehrfachvererbung, Interfaces, Komposition)

a) Ein Mensch lässt sich modellieren, indem seine üblichen Tätigkeiten abgebildet werden. Darunter fallen essen, schlafen, arbeiten und Autofahren. Ein Roboter mit einer künstlichen Intelligenz hat einen ähnlichen Satz Tätigkeiten: aufladen, warten, arbeiten und neuerdings – durch den Trend zu selbst fahrenden Autos – auch Autofahren.

   Sowohl Mensch als auch Roboter sollen eine Methode entscheide() haben, in der sie auf eine gegebene Gefahrensituation reagieren. Die Situation soll ein enum mit drei Werten sein: GEFAHR_LINKS, GEFAHR_RECHTS, GEFAHR_VORNE.

   Sowohl Mensch als auch Roboter reagieren gleich auf die Gefahren:

   - Bei einer GEFAHR_LINKS wird nach rechts ausgewichen
   - Bei einer GEFAHR_RECHTS wird nach links ausgewichen
   - Bei einer GEFAHR_VORNE wird gebremst.

   Der Mensch schätzt die Situation allerdings in 25% der Fälle nicht genau ein und ist UNENTSCHIEDEN.

Die Entscheidungen sollen auch durch ein enum mit den Werten: RECHTS, LINKS, BREMSEN, UNENTSCHIEDEN abgebildet werden.

b) Ein Cyborg ist sowohl ein Mensch als auch ein Roboter. Trotz des Stresses den das Aufladen, Essen, Warten, Schlafen, etc. mit sich bringt, fährt auch ein Cyborg gerne Auto und wird dort Gefahrensituationen ausgesetzt.
Wenn sich der Menschanteil und der Roboteranteil in ihrer Entscheidung einig sind, trifft der Cyborg die gleiche Entscheidung. Wenn der Menschanteil und der Roboteranteil unterschiedlicher Ansicht sind, dann trifft der Cyborg zufällig eine der beiden Entscheidungen.