

Anmerkungen

- **Abgabe:** Quellcode zu 5-1 auf Papier und über ILIAS. Lösung zu 5-2 auf Papier. Vorgegebene Dateien **nicht** ausdrucken!
- Jede Quellcode-Datei enthält **Name(n)** und **Matrikelnummer(n)**.

Aufgabe 5-1

Sie sollen ein “Vier gewinnt“ Spiel programmieren, bei dem man wahlweise gegen einen menschlichen Gegner oder den Computer spielen kann.

Laden Sie von ILIAS die Dateien `VierGewinnt.java`, `HumanPlayer.java`, `ComputerPlayer.java`, `Token.java` und `IPlayer.java` herunter. Die Klasse `VierGewinnt` enthält bereits Methoden `play()` (definiert den Spielablauf), `main` (startet das Spiel) und `displayField()` (graphische Darstellung des Spielfelds):

```

Player X choose a column between 1 and 7: 2
+---+---+---+---+---+---+
|   |   |   |   |   |   |   |
+---+---+---+---+---+---+
|   |   |   |   |   |   |   |
+---+---+---+---+---+---+
| X |   |   |   |   |   |   |
+---+---+---+---+---+---+
| 0 | X | 0 |   |   |   |   |
+---+---+---+---+---+---+
| X | 0 | X |   |   |   | X |
+---+---+---+---+---+---+
| 0 | 0 | 0 | X | X |   | 0 |
+---+---+---+---+---+---+
      1       2       3       4       5       6       7
Player X wins!

```

Um das Spiel zum Laufen zu bekommen, müssen Sie in der Klasse **VierGewinnt** die folgenden Methoden implementieren (die anderen gegebenen Methoden dürfen Sie *nicht* verändern):

1. **insertToken**: Der übergebene Stein (Token-Objekt) soll in die gewählte Spalte (**column**) des Spielfelds (Array **board**) gefüllt werden. Falls eine nicht existierende oder bereits bis oben gefüllte Spalte gewählt wurde, soll das Programm mit einer Fehlermeldung abbrechen. Verwenden Sie dazu **System.exit(1)**.
2. **isBoardFull**: gibt genau dann **true** zurück, wenn alle Felder durch einen Stein besetzt sind.
3. **checkVierGewinnt**: überprüft – ausgehend vom durch **col** und **row** gegebenen Feld – ob es in einer der vier Richtungen (d.h. **-**, **|**, **/**, ****) mindestens vier gleiche Steine gibt. In diesem Fall wird **true** zurückgegeben, andernfalls **false**. Tipp: Schreiben Sie für jede der vier Richtungen eine Hilfsmethode.

Verbessern Sie anschliessend die Klasse `ComputerPlayer` derart, dass der Computer zumindest ein wenig intelligenter spielt. Stellen Sie sicher, dass "er" keine ungünstigen Züge macht, also insbesondere keine Spalte auswählt, die bereits voll ist.

Aufgabe 5-2

1. Welches Problem tritt beim Ausführen des folgenden Programmcodes auf? Wie kann man es beheben?

```
int[] numbers = {1,2,3,5,8,13,21};
for(int i=2; i<=numbers.length; i++){
    System.out.println(numbers[i]);
}
```

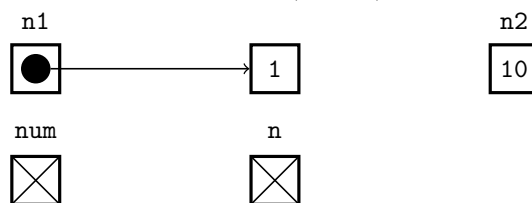
2. Gegeben sei die Klasse Num aus der Vorlesung:

```
public class Num{
    private int value;
    public Num(int update){ value = update; }
    public void setValue(int update){ value = update; }
    public String toString(){ return value + ""; }
}
```

Zeichnen Sie ein Diagramm, das die Wertzuordnungen für das folgende Programm illustriert.

```
1 public class Increment {
2     public static void main(String[] args) {
3         Num n1 = new Num(1);
4         int n2 = 10;
5         modify(n1);
6         modify(n2);
7     }
8     public static void modify(Num num){ num.setValue(100); }
9     public static void modify(int n) { n=100; }
10 }
```

Nach der Initialisierung (Zeile 4):



Bei Aufruf von `modify(Num)`:

...