Übungsblatt 1: Refactoring, JUnit, Entwurfskonzepte

1. Literaturstudium Refactoring

a. Machen Sie sich mit dem Thema Refactoring vertraut und erklären den Begriff in eigenen Worten (ggf. Literatur suchen!!).
b. Unter http://www.refactoring.com/catalog/index.html finden Sie eine ganze Sammlung von Beispielen zur Strukturverbesserung. Suchen Sie sich ein Beispiel aus dieser Liste aus und fertigen eine stichpunktartige Erklärung dieses Beispiels an.

2. Refactoring und Testbarkeit der Klasse von VierGewinnt

a. Arbeiten Sie sich in `JUnit` ein (`JUnit` Tutorial) und programmieren ein beliebiges Beispiel mit Hilfe einer `JUnit` Testklasse. Achten Sie darauf, dass Sie mit der (neusten) **JUnit Version 5.x https://junit.org/junit5/** arbeiten!

b. Gegeben ist ein Java-Programm zur Realisierung des Spiels **`VierGewinnt`** (s.zip Datei). Zunächst diskutieren Sie mit Ihrem Nachbarn oder Dozenten folgende Frage: Welches sind die größten Probleme im Hinblick auf Unit-Testbarkeit?

Danach müssen Sie die `JUnit` Testklasse `VierGewinntTest` zum Laufen bringen. Diese dort programmierten Tests dürfen nicht verändert werden. Sie müssen aber zusätzliche Tests einbauen, z.B. wenn Sie auf Bugs im Code stoßen.

Sinnvoll ist folgende Vorgehensweise (hier nicht zu viel ändern, sonst kommen Sie nicht ans Ziel!):

1. Den als fehlend angemeckerten Konstruktor einbauen (in Eclipse aus der Testklasse heraus mit Quickfix: Strg+1)
2. Die im Konstruktor übergebenen Streams `InputStream` und `PrintStream` statt `System.in`, `System.out` und `System.err` verwenden. Vorsicht: Der `Scanner` darf nur genau ein Mal intanziiert werden!
3. Aus `main` wird eine parameterlose Methode `play`.
4. `System.exit` beseitigen. Dazu muss `setzeFeld` einen return-Wert vom Typ `Character` liefern (das `zeichn` im Erfolgsfall, sonst `null`). Dieser Wert wird beim Aufruf in einer lokalen Variablen `winner` gespeichert . Ferner ist die dort vorhandene if-Abfrage zu ergänzen: `while (zaehler < columns*rows && winner == null)` Die beiden Ausgaben nach dieser Schleife müssen mit der Bedingung `if(winner == null)` versehen werden.

   Nach den Änderungen sollten Sie prüfen:

   1. Cross-Check: es darf kein `System....` mehr vorkommen.
   2. `play` wird nur aus den Tests aufgerufen. Ausnahme: sollten Sie das Spiel spielen wollen, so bauen Sie eine `main` -Methode ein mit: `new VierGewinnt(System.in, System.out).play();`

3. Refactoring und neuer Entwurf

Verändern Sie die Software schrittweise so, dass eine „bessere“ **objektorientierte Lösung von `VierGewinnt`** entsteht. Dokumentieren Sie Ihre Änderungen und alle Refactoring Maßnahmen.

4. Entwurfskonzepte

Reviewen Sie Ihre Architektur und erklären Sie die Begriffe Kohäsion, Kopplung und Separation of concern an Ihrem Beispiel.