

Sie sind hier: **FH Wedel** > **Mitarbeiter** > **Gerit Kaleck** > **Programmierpraktikum (Java)** > **Zwischenstand** > **Beispielprojekt**

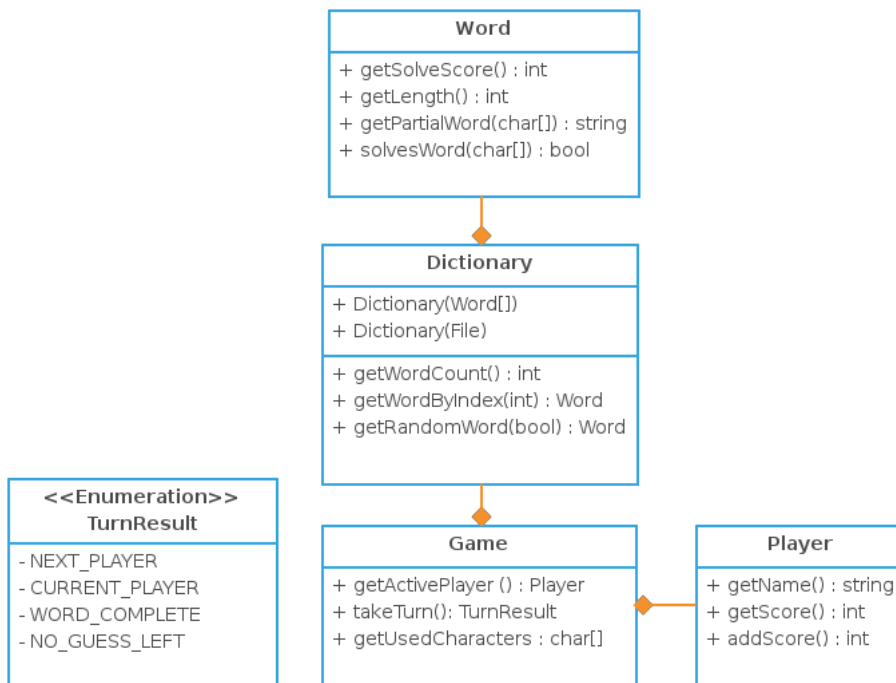
Beispielprojekt für die Abnahme

Das **Beispielprojekt** demonstriert, welche Herangehensweise sich für die Erstellung bis zur Abnahme des Zwischenstands eignet. Die grundlegende Aufgabenstellung, die diesem Beispiel zugrunde liegt, lautet dabei wie folgt:

Erstellt eine Mehrspielervariante von *Hangman*. Die Wörter sollen dabei aus Wörterbuchdateien eingelesen werden und bringen möglicherweise unterschiedlich viele Punkte. Um ein Spiel zu starten, müssen die teilnehmenden Spieler alle ihren Namen angeben, sich dann auf ein Wörterbuch einigen und entscheiden, ob die Wortauswahl rein zufällig oder ohne mehrfaches Auftreten eines Wortes im gleichen Spiel erfolgen soll. Die maximale Anzahl der Versuche hängt von der Länge des Wortes ab: Bei sehr kurzen Worten (≤ 4 Buchstaben) haben die Spieler maximal 7 Fehlversuche, bei sehr langen Worten (≥ 10 Buchstaben) haben sie nur 3 Fehlversuche. In allen anderen Fällen sind 5 Fehlversuche zulässig. Wenn sich in dem Wörterbuch keine neuen Worte mehr finden oder eine vorab festgelegte Rundenzahl abgelaufen ist, endet das Spiel und es gewinnt der Spieler mit der höchsten Punktzahl.

Datenstruktur

Um diese Aufgabenstellung zu erfüllen, wurden im Beispielprojekt vier Klassen angelegt. Die Rümpfe der Methoden bestehen im Regelfall aber lediglich aus dem Werfen einer Exception, um deutlich zu machen, dass die jeweilige Methode (noch) nicht implementiert ist.

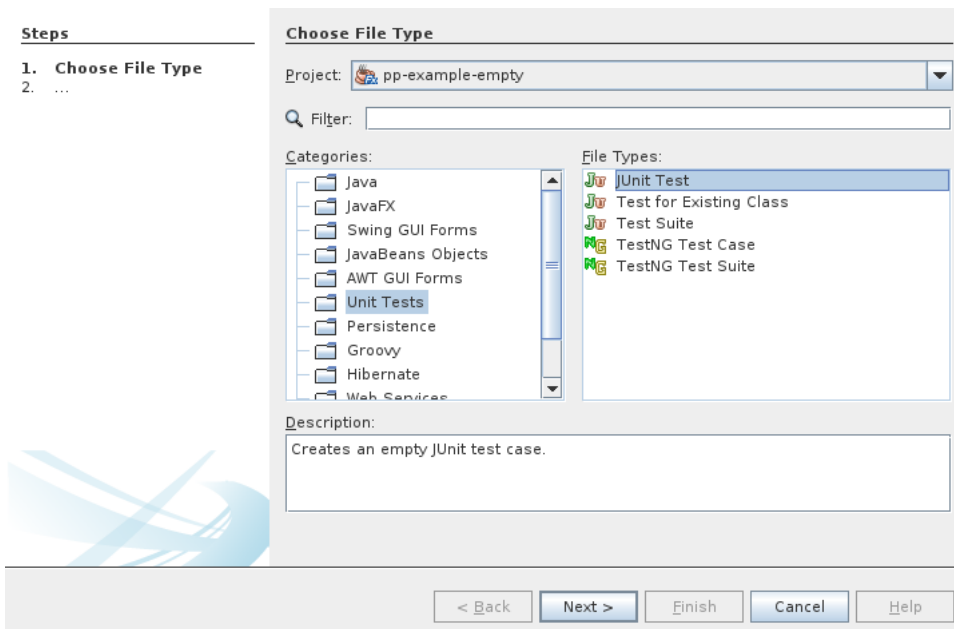


Testfälle

Trotzdem lassen sich für diese Struktur schon Testfälle schreiben, um den Programmfluss des Programms zu demonstrieren. Das Beispiel verwendet die JUnit Bibliothek, um den Umgang mit den Testfällen zu vereinfachen. Die Integration von JUnit ist in Netbeans sehr gelungen: um einen Testfall anzulegen, machen Sie einen Rechtsklick auf die Wurzel Ihres Projektes und fügen einen `JUnit Test` hinzu. Sollte die Option nicht direkt im Kontextmenü angeboten werden, kann sie über den allgemeinen "Neue Datei" Dialog hinzugefügt werden.

Steps

1. **Choose File Type**
2. ...



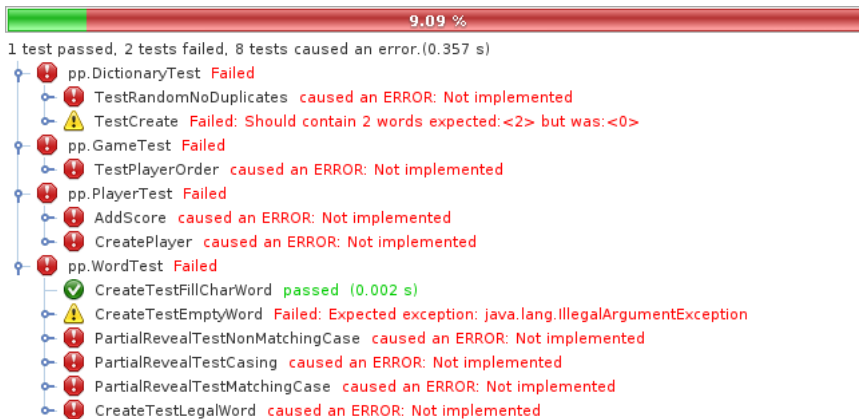
Typischerweise legt man dann einen Unit-Test je zu testender Klasse an und platziert diese im gleichen package. Die Codegenerierung, welche Ihnen von Netbeans angeboten wird, können Sie einfach abschalten, damit würde nur von Ihnen nicht genutzter Code entstehen. Wenn Sie den Test hinzugefügt haben, werden Sie feststellen, dass er von Netbeans in einem speziellen Ordner für Tests platziert worden ist. Auf diese Weise kann man die Testfälle bei der Veröffentlichung ganz einfach aus der endgültigen .jar Datei heraushalten. Sollten Sie nach der zu verwendenden JUnit Version gefragt werden, wählen Sie bitte Version 4.

Sie können nun eigene Testfälle schreiben, dazu müssen Sie eine öffentliche Methode mit den folgenden Eigenschaften erstellen:

- » sie nimmt keine Parameter entgegen,
- » der Rückgabotyp ist `void`,
- » sie ist mit `@Test` annotiert.

Wenn Sie eine solche Methode erstellt haben, können Sie nun mit den `assert*` Methoden aus `org.junit.Assert` arbeiten, um verschiedene Testfälle zu schreiben. Werfen Sie hier am besten einen Blick auf die Testfälle des Beispielprojektes.

Um die Tests auszuführen, können Sie wahlweise `Strg + F6` drücken, um nur die aktuelle Datei zu testen oder `Alt + F6` um alle Tests des aktuellen Projektes auszuführen.



JUnit unterscheidet zwischen zwei Arten von Fehlern: Bei Errors handelt es sich um unbehandelte Ausnahmen, also im Endeffekt einen Absturz des Programmes. Mit Failed werden Testfälle markiert, bei denen eine geforderte Assertion nicht eingehalten werden konnte.

Probleme der Musterlösung

Wie schon eingangs erwähnt, würde diese "Musterlösung" nicht mit einer vollen Punktzahl bewertet werden:

- » Die Aufgabenstellung beschreibt, dass Spieler über einen Namen verfügen, die GUI sieht jedoch keine Möglichkeit vor, einen Namen einzugeben.
- » Außerdem wird in der Aufgabenstellung beschrieben, welche Parameter für ein Spiel gesetzt werden müssen, auch diese Möglichkeit bietet die GUI nicht.
- » Es besteht keine Möglichkeit, die Anzahl der gespielten Worte in z.B. der Game Klasse festzulegen.
- » Es wurde kein GUIController angelegt, die hierfür vorgesehenen Methoden sind also nicht ersichtlich.

Kontakt FH Wedel

Staatlich anerkannte
Fachhochschule Wedel
Gemeinnützige Schul-
gesellschaft mbH
Feldstraße 143
D - 22880 Wedel
Tel.: +49 (0)4103 - 8048-0
Fax: +49 (0)4103 - 8048-39
E-Mail: sekretariat@fh-wedel.de