

## Übungsblatt #2

**Neu:** Um Ihnen die Bearbeitung der Aufgaben zu erleichtern, haben wir die Verzeichnissstruktur der Vorgabe etwas abgeändert. In der Vorgabe finden Sie nun zu jeder Aufgabe einen Unterordner mit einer eigenen Testumgebung. Somit sind die Aufgaben inklusive der Tests auch voneinander vollständig entkoppelt. Die Tests werden immernoch mit dem enthaltenen **gradlew** Skript gestartet, jedoch mit einem anderen Parameter: **./gradlew testen** Dies ist für alle Teilaufgaben identisch.

**Wichtig: Überprüfung fehlerhafter Eingaben.** Ab diesem Blatt wird vorausgesetzt, dass Sie bei jedem Ihrer Programme die Gültigkeit der von der Kommandozeile übergebenen Parameter prüfen (unabhängig von der Aufgabe und Erwähnung im Text). Hierzu muss auch die Anzahl der übergebenen Argumente, sowie deren Wertebereiche überprüft werden. Da sich das Konzept der Exceptions noch nicht in der Vorlesung kennen gelernt haben, müssen Sie diese im Falle einer fehlerhaften Eingabe nicht abfangen. Fehlende Überprüfungen der Eingaben führen zu Punktabzügen. Gibt es eine falsche Eingabe, geben Sie **ERROR** aus. Mithilfe von `args.length` können Sie zum Beispiel überprüfen, ob auch Argumente vorhanden sind: `if (args.length > 0) { ... }`

1. **Fehler finden.**

10 Punkte

**Abzugebene Datei(en):** Bughunt02.java

In der zip Datei finden Sie ein Programm *Bughunt02.java*. Korrigieren Sie das Programm, so dass es kompiliert und funktioniert und geben Sie die korrigierte Version ab.

2. **Fehler finden (2).**

10 Punkte

**Abzugebene Datei(en):** Bughunt03.txt

In der zip Datei finden Sie ein Programm *Bughunt03.java*, das zwei nicht terminierende Schleifen hat. Beschreiben Sie in der Datei *Bughunt03.txt* in wenigen (3-4) Sätzen was die jeweilige Fehlerursache war und was Sie aus den Beispielen über Zählvariablen gelernt haben.

3. **Schachbrett.**

20 Punkte

**Abzugebene Datei(en):** Schachbrett.java

Schreiben Sie ein Programm *Schachbrett*, das eine positive, ganze Zahl  $n$  übergeben bekommt und ein Schachbrett der Größe  $n \times n$  ausgibt. Schwarze Felder werden durch einen Stern, weiße Felder durch ein Leerzeichen repräsentiert. Das Brett beginnt oben links mit einem schwarzen Feld. Wenn die eingegebene Zahl nicht positiv ist, soll das Programm **ERROR** ausgeben.

4. **Größter gemeinsamer Teiler.**

20 Punkte

**Abzugebene Datei(en):** Gcd.java

Schreiben Sie ein Programm *Gcd*, das den größten gemeinsamen Teiler zweier nicht-negativer ganzer Zahlen mit Hilfe des Euklidischen Algorithmus berechnet und ausgibt. Der Algorithmus, den Sie verwenden sollen, ist unter [https://de.wikipedia.org/wiki/Euklidischer\\_Algorithmus#Iterative\\_Variante](https://de.wikipedia.org/wiki/Euklidischer_Algorithmus#Iterative_Variante) beschrieben.

## 5. Eulersche Vermutung.

40 Punkte

**Abzugebene Datei(en):** CountEuler.java

Im Jahr 1769 formulierte Leonhard Euler eine verallgemeinerte Version des Großen Fermatschen Satzes, die besagt, dass für alle  $n > 2$  mindestens  $n$   $n$ -te Potenzen nötig sind, um eine Summe zu erhalten, die selbst eine  $n$ -te Potenz ist.

Erst 1967 konnte die Eulersche Vermutung durch ein Gegenbeispiel für  $n = 5$  widerlegt werden. Bei diesem Gegenbeispiel wurden fünf Zahlen  $a, b, c, d, e$  gefunden, für die gilt  $a^5 + b^5 + c^5 + d^5 = e^5$ . Dies ist ein Widerspruch zur Eulerschen Vermutung die besagt, dass bei  $n = 5$  mindestens fünf Summanden (also insgesamt sechs Zahlen) nötig sein sollten. Mit modernen Computern kann man dieses Gegenbeispiel schnell durch einen Brute-Force-Ansatz (d.h. durch Ausprobieren) finden.

Schreiben Sie ein Programm *CounterEuler*, das dieses Gegenbeispiel für  $n = 5$  findet und die Zahlen für  $a, b, c, d, e$  ausgibt. Verwenden Sie dazu fünf ineinandergeschachtelte Schleifen. Sie dürfen die Tatsache verwenden, dass alle Zahlen  $a, b, c, d$  und  $e$  kleiner sind als 200. In der Programmvorlage finden Sie einige Tipps, wie Sie die Suche beschleunigen können. Für eine volle Punktzahl sollten diese Tipps umgesetzt werden. Benutzen Sie den Datentyp `long` um die Potenzen und Summen zu speichern.