

Lernziele zur Java-Programmierung (Teil 1)

Die Programmierprüfung wird ähnlich wie die Übungen gestaltet sein. Das Internet wird aber nur am Anfang für das Klonen des Repositories und am Schluss für die Abgabe verfügbar sein. Andersweitige Benutzung des Internets ist untersagt. Aufgabenstellung und Abgabe erfolgen über Github. Es werden nur die bei den Übungen verwendeten Module einzusetzen sein (`Java.util.Random`, `Java.util.Arrays`, `Java.lang.Math`, `Java.util.Scanner`). Andere Module sind nicht zugelassen und führen zu Punkteabzug. Das `java-infos-and-snippets` Repository sollte vor Beginn der Prüfung geklont (mit `git clone`) bzw. aktualisiert (mit `git pull`) werden. Es ist erlaubt, in diesem Repository nachzuschauen, wie gewisse Befehle heissen oder wie man sie einsetzt.

- 1) Ein funktionierendes Java-Programm (mit `main`-Methode) von Grund auf schreiben
- 2) Java-Programme kompilieren und ausführen
- 3) Im Java-Quellcode Kommentare verfassen und lesen
- 4) Über Gib/Github Aufgaben empfangen und Lösungen abgeben
- 5) Kommandozeilen-Eingaben parsen
- 6) Die primitiven Datentypen (`boolean`, `char`, `int`, `long`, `float`, `double` etc.) kennen und einsetzen
- 7) Die `String`-Klasse kennen; Strings mittels des `+`-Operators zusammenhängen und dabei das implizite Casting von verschiedenen Datentypen zu `String` verstehen
- 8) Arrays deklarieren und initialisieren, Array-Werte zuweisen und herauslesen
- 9) Arrays mit Hilfe der `Arrays.copyOf`-Methode redimensionieren
- 10) Arithmetische Operatoren (wie `+`, `-`, `*`, `/`, `%`, `++`, `--`) einsetzen
- 11) Zuweisungsoperatoren (wie `=`, `+=`, `-=`, `*=` etc.) einsetzen
- 12) Logische Operatoren (wie `&&`, `||`, `!`) einsetzen
- 13) Vergleichsoperatoren (wie `==`, `!=`, `>`, `<`, `>=`, `<=`) einsetzen
- 14) Wissen, wie man die Werte zweier Variablen vertauscht
- 15) Verzweigungen mittels `if`, `else`, `elseif`, sowie `continue` und `break` programmieren
- 16) Das `switch`-Statements verwenden, um über ganzzahlige oder `String`-Werte zu verzweigen
- 17) `for`-Schleifen, `while`-Schleifen und `do ...while`-Schleifen schreiben; über Arrays via einer `for`-Schleife iterieren
- 18) Eingabe und Ausgabe von Variablen verschiedenen Typs über die Konsole mit Hilfe von `System.out` und `System.in`, sowie der `java.util.Scanner`-Klasse
- 19) Die Klasse `Java.util.Random` verwenden, um Zufallszahlen zu erzeugen

- 20) Wissen, wofür `enums` gebraucht werden und `enums` inklusive ihrer `name`, `ordinal`, `values` und `valueOf`-Methoden einsetzen
- 21) Wissen, was eine Methode, was ein Parameter einer Methode ist und was ein Rückgabewert ist
- 22) Methoden rekursiv aufrufen, um Berechnungen durchzuführen
- 23) Die `toString`-Methode von Objekten in Java kennen, implementieren und anwenden
- 24) Wissen, was eine Klasse und was eine Instanz einer Klasse ist
- 25) Wissen, was eine Instanzvariable und was eine Klassenvariable ist
- 26) Wissen, was eine statische Methode einer Klasse ist
- 27) Konstanten mit Hilfe des `final`-Schlüsselworts definieren
- 28) Statische und nicht-statische Methoden einer Klasse aufrufen
- 29) Gross-/Kleinschreib-Konvention für Variablen, Klassen, Konstanten beherrschen
- 30) Triviale und nicht-triviale Getter und Setter für Instanzvariablen und Klassenvariablen schreiben
- 31) Verschiedene Konstruktoren für Klassen schreiben, auch mit Hilfe von `this` verkettete Konstruktoren
- 32) Klassen programmieren und instanzieren
- 33) Das `null`-Objekt kennen und verwenden
- 34) Wissen, was abgeleitete Klassen (und Basisklassen) sind und wofür sie verwendet werden
- 35) Sichtbarkeit (`private`, `protected`, `public`) von Variablen und Methoden in Klassen und abgeleiteten Klassen verstehen und korrekt einsetzen
- 36) Einen Konstruktor der Basisklasse im Konstruktor der abgeleiteten Klasse aufrufen
- 37) Das (implizite) Casting einer Instanz einer abgeleiteten Klasse zur Basisklasse verstehen und anwenden
- 38) Das (explizite) Casting einer Instanz einer Basisklasse zur passenden abgeleiteten Klasse verstehen und anwenden
- 39) Das Horner-Schema zur Berechnung von Polynomwerten verstehen und implementieren
- 40) Zweidimensionale Arrays für die Matrizen-Rechnung einsetzen
- 41) Verstehen, was ein `assert`-Statement ist und wofür man es einsetzen kann
- 42) Berechnungsmethoden in Java umsetzen