



Programmierung für Mathematik (HS13)

Übung 5

1 Aufgabe: Eclipse IDE

1.1 Lernziele

1. Die Entwicklungsumgebung Eclipse einrichten.
2. Eclipse kennen lernen und mit Eclipse arbeiten können.

1.2 Aufgabenstellung

- Falls Sie dies noch nicht gemacht haben, installieren Sie die Java Platform (JDK)¹. JDK bedeutet Java Development Kit und beinhaltet die komplette JRE (Java Runtime Environment) und benötigte Tools für das Entwickeln, Debuggen und Testen von Java Programmen. Die JRE ist für End-User gedacht und verfügt über alle benötigten Klassen, um Java Applikationen laufen zu lassen.
- Laden Sie nun die Eclipse IDE² herunter³. Sie müssen keine eigentliche Installation von Eclipse vornehmen. Extrahieren Sie das File an den gewünschten Installationsort. Anschliessend können Sie Eclipse starten indem Sie auf das Eclipse Icon klicken (unter Windows Eclipse.exe).
- Schauen Sie sich nun ein Tutorial-Video⁴ zu Eclipse an. Das Tutorial ist zugleich eine gute Repetition zu Klassen, Methoden und Variablen. Falls Sie Fragen dazu haben, können Sie diese im OLAT oder in der nächsten Übungsstunde stellen.

¹Sie finden die aktuellste Version (7u40) unter <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

²IDE steht für Integrated Development Environment

³Sie finden die aktuellste Version (4.3.1) unter <http://www.eclipse.org/downloads>. Wählen Sie Eclipse Standard.

⁴Sie finden das Video hier: <http://eclipsetutorial.sourceforge.net/totalbeginner01/lesson01.html>

2 Aufgabe: Arrays

2.1 Lernziele

1. Arrays verstehen und Aktionen darauf ausführen.

2.2 Aufgabenstellung

Erstellen Sie nun in Eclipse ein neues Projekt und erstellen Sie eine Klasse `MyList`. Diese Klasse soll ein `int` Array besitzen. Die Klasse soll die folgenden Methoden haben:

- `initializeArray`: Diese Methode erwartet einen `int`-Wert für die Grösse des Arrays als Parameter und instanziiert das Array mit dem angegebenen Wert.
- `setArray`: Diese Methode erwartet ein `int`-Array als Parameter und instantiiert dieses mit dem eigenen Array.
- `add`: Diese Methode erwartet ein `int`-Wert als Parameter und fügt diesen am Ende des Arrays hinzu. Das heisst, das Array wird um 1 vergrössert.
- `insert`: Diese Methode erwartet ein `int`-Wert und eine Position (ebenfalls `int`) als Parameter und fügt diesen an der angegebenen Position hinzu.
- `get`: Diese Methode erwartet einen `int` als Parameter und gibt den Wert des Arrays an dieser Stelle aus.
- `erase`: Diese Methode leert das Array.
- `pull`: Diese Methode löscht den letzten Eintrag des Arrays und gibt ihn zurück.
- `print`: Diese Methode gibt die Werte des Arrays der Reihe nach in der Konsole aus.
- `size`: Diese Methode gibt die Grösse des Arrays zurück.

3 Aufgabe: Nassi-Shneiderman Diagramme

3.1 Lernziele

1. Ein in Pseudocode beschriebenes Verfahren in ein entsprechendes Nassi-Shneiderman-Diagramm umsetzen können.
2. Code lesen und in Worte fassen.
3. Pseudocode lesen, verstehen und in Java schreiben.
4. Verschachtelte Schleifen anwenden können.
5. Nassi-Shneiderman Diagramme in Code übersetzen.

3.2 Aufgabenstellung

a) Nassi-Shneiderman-Diagramm zeichnen

Zeichnen Sie mittels folgender Notation ein Nassi-Shneiderman-Diagramm, das einen Algorithmus für die Bestimmung des grössten gemeinsamen Teilers (GGT) zweier Zahlen beschreibt. Schreiben Sie die Methode anschliessend in Java. Verwenden Sie den folgenden Algorithmus:

```
1 int gcd(int first, int second) {  
2     SOLANGE second UNGLEICH 0  
3         WENN first GROESSER second  
4             DANN SETZE first = first - second  
5             SONST SETZE second = second - first  
6     RETURN first  
7 }
```

Listing 1: Pseudocode für den grössten gemeinsamen Teiler

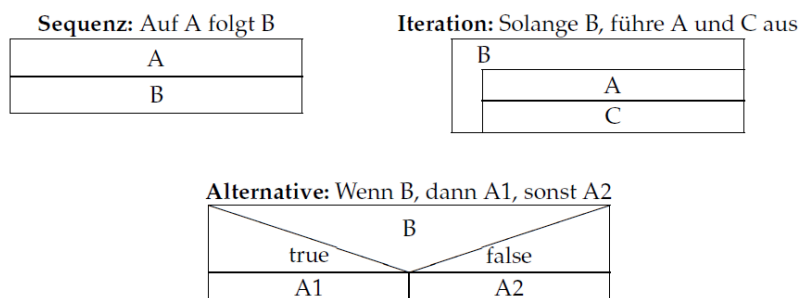


Figure 1: Nassi-Shneiderman-Diagram Notation

b) Programmcode beschreiben

Lesen Sie den untenstehenden Java Code durch, ohne ihn auszuführen. Beschreiben Sie in einigen Sätzen, was diese bewirkt.

```
1 for (int x = 1; x < 100; x++) {  
2     for (int y = 1; y < 100; y++) {  
3         for (int z = 1; z < 100; z++) {  
4             if (x * x + y * y == z * z) {  
5                 System.out.println("x = " + x + " y = " + y + " z = " + z);  
6             }  
7         }  
8     }  
9 }
```

Listing 2: Java Code für den grössten gemeinsamen Teiler

c) Nassi-Shneiderman-Diagramm interpretieren

Anhand des folgenden Nassi-Shneiderman-Diagrammes wird illustriert, wie der obenstehende Code optimiert werden kann. Implementieren Sie den Code direkt in der `main()`-Methode eines TestDrivers

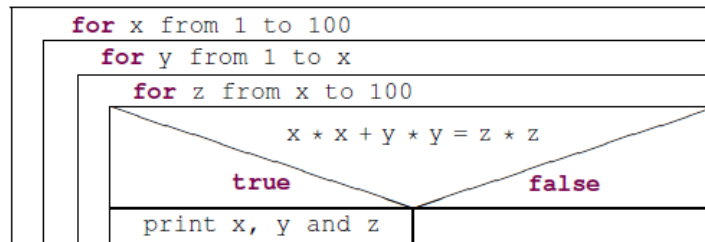


Figure 2: Nassi-Shneiderman-Diagramm zur Code Optimierung