

## Aufgabe 1

### Berechnung einer Zufallszahl

Im Praktikumsversuch 1 haben Sie die Formel  $(a \cdot z + b) \bmod m$  zur Berechnung einer Zufallszahl programmiert. Betten Sie die Formel nun in eine Methode ein. Die Methode soll die benötigten Eingabewerte als Parameter erhalten und den berechneten Wert als Rückgabe liefern.

Schreiben Sie zudem ein Hauptprogramm, das die Parameterwerte von der Tastatur einliest, mit ihnen die Methode aufruft und ihren Rückgabewert auf dem Bildschirm ausgibt. Stellen Sie sicher, dass nur zulässige Werte eingegeben werden können. Machen Sie bitte eine geeignete Ausgabe, falls ein unzulässiger Wert eingegeben wurde.

## Aufgabe 2

### Berechnung einer Folge von Zufallszahlen

Betten Sie die Aufrufe der obigen Methode aus der vorherigen Aufgabe zur Berechnung der Zufallszahl so in eine Schleife ein, dass insgesamt **20** Zufallszahlen berechnet und auf den Bildschirm ausgegeben werden.

Beim ersten Aufruf soll der Parameterwert **0** übergeben werden, beim zweiten Methodenaufruf der Rückgabewert des ersten Aufrufs usw. Es ergibt sich also eine Folge von Zufallszahlen, bei der die jeweils nächste Zahl aus der zuletzt gelieferten berechnet wird.

Stellen Sie sicher, dass nur zulässige Werte eingegeben werden können. Machen Sie bitte eine geeignete Ausgabe, falls ein unzulässiger Wert eingegeben wurde.

## Aufgabe 3

### Berechnung eines Hashwertes

Verallgemeinern Sie die Formel  $(c_1 + c_2 + c_3 + c_4) \bmod m$  zur Berechnung eines Hash-Werts aus dem Praktikumsversuch 1: Die Formel soll nun auf Strings beliebiger Länge angewandt werden können, also die ASCII-Werte der einzelnen Zeichen des Strings addieren und anschließend die Modulo-Operation anwenden.

Schreiben Sie eine entsprechende Methode, die den String als Parameter übergeben bekommt und als Rückgabe den Hash-Wert liefert.

Schreiben Sie zudem ein Hauptprogramm, das einen String von der Tastatur einliest, mit ihm die Methode aufruft und ihren Rückgabewert auf den Bildschirm ausgibt. Das Hauptprogramm soll in einer Schleife laufen, bis der Benutzer eine Beendigung wünscht.

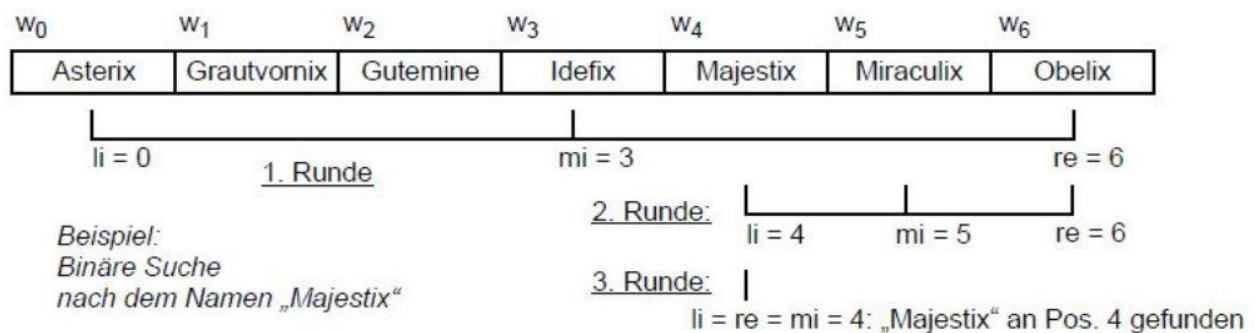
Stellen Sie sicher, dass nur zulässige Werte eingegeben werden können. Machen Sie bitte eine geeignete Ausgabe, falls ein unzulässiger Wert eingegeben wurde.

## Aufgabe 4

### Binäre Suche

Wenn eine Sammlung von Werten (ein so genanntes „Feld“) aufsteigend sortiert ist, so kann man darin effizient suchen, indem man den gesuchten Wert systematisch „einkreist“. Eine solche „binäre Suche“ geht wie folgt vor: Man überprüft zunächst den Wert in der Mitte des Felds. Ist dieser gleich dem gesuchten Wert, so ist die Suche damit beendet. Ist dieser größer als der gesuchte Wert, so wird die Suche im Teilfeld links davon fortgesetzt; ist er kleiner, sucht man im Teilfeld rechts davon weiter. Die übrigen Einträge können ignoriert werden, denn dort kann ja wegen der Sortierung der gesuchte Wert nicht sein. Setzt man diese Vorgehensweise fort, so ergeben sich immer kleinere Teilfelder, bis entweder der gewünschte Wert gefunden oder (bei Erreichen eines Teilfelds mit nur einem oder gar keinem Eintrag) erkannt wird, dass der Wert nicht vorhanden ist.

Die folgende Grafik demonstriert die Suche nach einem Wort in einem aufsteigend sortierten Feld von Wörtern. Dabei bezeichnet **li** den linken Begrenzungsindex des aktuell betrachteten Teilfelds, **re** den rechten Begrenzungsindex und **mi** den Index in der Mitte.



Stellen Sie den Algorithmus „Binäre Suche“ nun durch ein Struktogramm dar. Das Struktogramm muss so genau sein, dass in ihm alle Schleifen und Verzweigungen des späteren Programms zu erkennen sind. Zudem muss eindeutig gesagt werden, an welcher Position des Wörterarrays jeweils gearbeitet wird (Index angeben!).

Setzen Sie das Struktogramm anschließend in ein Java-Programm, das folgendes leistet:

- Frage an den Benutzer, wie viele Wörter eingelesen werden sollen. Wiederholung dieser Frage, solange der Benutzer eine Zahl kleiner als 6 eingibt.
- Erzeugung eines String-Arrays der geforderten Größe.
- Einlesen von Wörtern in den Array.
- Frage an den Benutzer, nach welchem Wort gesucht werden soll.
- Suche nach diesem Wort und Ausgabe des Ergebnisses (Position, an der das Wort gefunden wurde oder Meldung „nicht gefunden“) auf den Bildschirm.
- Frage an den Benutzer, ob eine weitere Suche gewünscht wird; wenn ja, neue Eingabe eines Suchworts und anschließende Suche.

## **Aufgabe 5**

### **Methode zur Suche**

Ändern Sie das Java-Programm aus der Aufgabe 4 so, dass der eigentliche Suchvorgang durch eine Methode ausgeführt wird. Die Methode soll vom Hauptprogramm aufgerufen werden und dabei das Array, welches zu bearbeiten ist, als Parameter übergeben bekommen. Sämtliche Einlese- und Ausgabevorgänge sollen im Hauptprogramm stattfinden!