**GL der Programmierung**

**Eindimensionale Arrays**

**Hausübung der 6. Einheit vom 27. 4. 2018**

**! Achtung !  
Lösen Sie bitte das Beispiel 1 (Gefangene) und mindestens 6 weitere Beispiele.**

**Die Zahlen in eckigen Klammern geben die Punkte für die Präsentation der richtigen Lösung an.  
Benennen Sie die Beispiele wie in der Angabe angegeben.   
Geben Sie alle in dasselbe Package und benennen Sie das Package mit Ihrem Namen (Nachname + Vorname mit Kleinbuchstaben beginnend) und der Nummer der Hausübung, getrennt durch ‘\_‘, also z.B.**

**mayerRobert\_6**

**Exportieren Sie das Package aus eclipse – vergessen Sie dabei nicht das Package zu selektieren und „Create only selected directories“ zu wählen – in eine Datei, die Ihren Namen, die Nummer der Hausübung, sowie die Nummern der gelösten Beispiele trägt, also z.B.:**

**Mayer\_Robert\_Aufgabe\_6\_Bsps\_1\_3\_4  
Geben Sie die exportierte Datei rechtzeitig auf moodle ab.**

## Beispielgruppe 6 – eindimensionale Arrays

1. Gefangene: Der König wird 66 und will aus diesem Anlass auch eine nette Geste setzen und ein paar der 111 Gefangenen freilassen, die derzeit in den von 1 bis 111 durchnummerierten Einzelzellen schmachten. Zuerst wollte er das Los entscheiden lassen, aber dann kam er auf folgende Methode mit der entschieden wird, wer freikommt und wer nicht:

Derzeit sind natürlich alle Zellen versperrt.   
Der Gefängniswärter muss nun – im ersten Durchgang - alle Zellen abgehen und jede Zelle aufsperren.  
Im zweiten Durchgang wird jede zweite Zelle (also Zelle Nr. 2, Zelle Nr. 4, …) wieder versperrt.  
Im dritten Durchgang wird es etwas komplizierter:   
Der Durchgang betrifft jede dritte Zelle (also Nr. 3, Nr. 6, …). Ist die Zelle offen, so wird sie versperrt. Ist die Zelle versperrt, so wird sie geöffnet.  
Im vierten Durchgang wird nun mit jeder 4. Zelle so verfahren, im 5. Durchgang mit jeder 5., …

Nach dem 66. Durchgang endet das Verfahren.

Alle Gefangenen, deren Zellentür nun offen steht dürfen nach Hause gehen, alle anderen müssen weiter im Kerker darben.  
Wie viele Gefangenen werden letzten Endes nach Hause gehen dürfen?  
In welchen Zellen sitzen diese? [4]

1. WuerfelDie6: Bei diesem Spiel wird reihum gewürfelt. Wer einen 6er würfelt, hat das Spiel gewonnen. Wird dabei von niemandem ein 6er gewürfelt, so wird eine neue Würfelrunde begonnen. Dies wird so lange wiederholt, bis es endlich einen Sieger gibt. Würfeln innerhalb einer Würfelrunde mehrere Personen einen 6er, so haben diese alle gewonnen.  
   Die Namen der beteiligten Personen sollen fix in einem Array gespeichert sein (Sie müssen diese nicht einlesen). [4]
2. Wuerfel3Mal6: Bei diesem Spiel würfeln mehrere Personen reihum. Sieger ist die Person, die als erstes zum dritten Mal einen 6er würfelt. Dies muss nicht in aufeinanderfolgenden Runden geschehen. Würfeln mehrere Personen in derselben Runde den dritten Sechser, gibt es mehrere SiegerInnen.  
   Lesen Sie dabei aber Anzahl und die Namen der SpielerInnen ein. [6]
3. FolgeInArray: Berechnen Sie die Folge

1 ; ½ ; 1/3 ; ¼ ; 1/5 ; ...

bis zum 100. Glied (1/100) und speichern Sie alle Glieder in einem Array. Geben Sie danach nur die geraden Elemente des Arrays (das 2., 4., 6., ... ) aus. [4]

1. ReiheInArray: Berechnen Sie die Reihe

1 + ½ + 1/3 + ¼ + 1/5 + ...

also

1 ; 1 + ½ ; 1 + ½ + 1/3 ; ...

bis zum 100. Glied (1 + ... + 1/100) und speichern Sie diese in einem Array. Geben Sie danach nur die ungeraden Elemente des Arrays (das 1., 3., 5., ... ) aus. [4]

1. Durchschnitt: Lesen Sie beliebig viele (maximal 50) Zahlen ein (Abschluss mit 0) und speichern Sie diese in einem Array. Berechnen Sie danach den Durchschnitt der eingelesenen Werte. [4]
2. WuerfelStatistikArray: Führen Sie zuerst mit Hilfe eines Zufallszahlengenerators 100 Würfe mit einem Würfel durch und speichern Sie die Ergebnisse in einem Array. Geben Sie danach aus, wie oft jede mögliche Augenzahl geworfen wurde. [5]
3. Zufallszahlen: Generieren Sie 1000000 Zufallszahlen von 123400 bis 123499. Geben Sie danach aus, wie oft jede dieser Zahlen erzeugt wurde. [5]
4. Vorhanden: Lesen Sie 10 Zahlen in ein Array ein. Lesen Sie dann eine weitere Zahl ein und stellen Sie fest, ob die neue Zahl im Array vorkommt. [5]
5. ZahlenMix: Generieren Sie ein Array mit 10 *verschiedenen* Zufallszahlen von 65 bis 90. Stellen Sie dabei sicher, dass keine der Zahlen doppelt vorkommt. [6]
6. Doppelt: Lesen Sie 10 Zahlen in ein Array ein und stellen Sie fest ob eine Zahl doppelt im Array vorkommt. [5]
7. GeradeUngerade: Lesen Sie beliebig viele (maximal 50) Zahlen ein (Abschluss mit 0) und speichern Sie diese in einem Array. Danach teilen Sie die Zahlen auf zwei weitere Arrays auf, wobei die geraden Zahlen in das eine Array und die ungeraden Zahlen in das andere Array kommen sollen.   
   Geben Sie dann zuerst die geraden, danach die ungeraden Zahlen aus. [5]
8. Eratosthenes: Bestimmen Sie alle Primzahlen bis zu einem gegebenen n mit Hilfe des „Siebs des Eratosthenes“. [9]
9. Zahlenteilen: Lesen Sie maximal 50 Zahlen (Abschluss mit 0) in ein Feld F ein, sowie einen Wert x. Teilen Sie den Inhalt des Feldes F so auf zwei weitere Felder Fg und Fk auf, dass alle Werte, die größer als x sind, in Fg liegen und alle anderen in Fk.  
   Geben Sie dann abwechselnd je eine Zahl aus Fg und eine aus Fk aus, solange bis alle Zahlen ausgegeben sind. [7]  
   Beispiel:

Eingabe: 3 14 31 7 29 10 18 15 19 0  
      x=10  
Fg enthält dann: 14 31 29 18 15 19 und   
Fk enthält dann: 3 7 10  
Ausgabe: 14 3 31 7 29 10 18 15 19