

Bachelor-Studiengang Informatik  
Übungen zur Vorlesung "Grundlagen der Informatik" (GDI), WS 2011/2012

Übungsblatt 7: Arrays (nach D. Busch)

Ausgabe am: 11.11.2011  
Abgabe am: 25.11.2011

**Aufgabe 1: Parkplatzleitsystem**

**2 + 3 + 3 + 3 + 4 + 3 = 17 Punkte**

Die Fachschaft der Informatik plant dieses Jahr einen Weihnachtsmarkt an der Hochschule und reserviert aus diesem Grund 100 Stellplätze auf dem neu angelegten Parkplatzgelände. Um den Aufwand der Parkplatzsuche so gering wie möglich zu halten hat die Fachschaft folgende Idee:

- Die Parkplätze werden fortlaufend von 1 bis 100 nummeriert.
  - Jedes einfahrende Auto bekommt den ersten freien Stellplatz zugewiesen.
- a) Die Fachschaft beauftragt Sie ein Parkplatzleitsystem zu entwerfen, das die Zuweisung der Parkplätze automatisiert. Die zu implementierende Klasse soll *Parkplatzleitsystem* heißen und als repräsentative Datenstruktur für die Parkplätze sollen Sie ein Array verwenden.
- i. Was für einen Elementtyp würden Sie für das Array wählen? Begründen Sie Ihre Antwort.
  - ii. Schreiben Sie eine Funktion *static int findeFreieStelle(ihrDatentyp parkplätze[])*, welche die erste freie Stelle im Parkplatz-Array findet und die Parkplatzznummer zurückgibt. Ist keine freie Stelle mehr im Parkhaus vorhanden, so soll die Funktion als Rückgabewert -1 liefern.
  - iii. Schreiben Sie eine Funktion *static int zähleFreieStellen(ihrDatentyp parkplätze[])*, welche die noch freien Stellen im Parkplatz-Array zählt und die Anzahl zurückgibt.
- b) Zum Testen Ihrer beiden Funktionen sollen Sie eine Simulation für das Parkplatzleitsystem schreiben. Entwerfen Sie dazu eine *main*-Methode in der Klasse *Parkplatzleitsystem*, die Ihr Array erstellt und die Simulation steuert. Über ein Menü auf der Konsole soll entschieden werden, ob ein Auto ein- oder ausfahren soll. Erweitern Sie die Klasse *Parkplatzleitsystem* um folgende Prozeduren um das Ein- und Ausfahren zu simulieren. Benutzen Sie bei der Implementierung sinnvoll Ihre bisherigen Funktionen:
- i. Schreiben Sie eine Prozedur *static void autoFährtAufParkplatz(ihrDatentyp parkplätze[])*, welche das Einfahren eines Autos simuliert. In dieser Prozedur wird die zugewiesene Stelle im Parkplatz-Array belegt. Die Prozedur soll folgende Meldungen auf der Konsole ausgeben:
    - "Einfahren erfolgreich, noch X Plätze sind frei.", wobei X für die Anzahl der freien Plätze steht.
    - "Der Parkplatz ist voll!", falls der Parkplatz keine freie Stelle mehr hat.
  - ii. Schreiben Sie eine Prozedur *static void autoFährtVomParkplatz(ihrDatentyp parkplätze[])*, welche das Ausfahren eines Autos simuliert. In dieser Prozedur wird zufällig eine belegte Stelle aus dem Parkplatz-Array ausgewählt und wieder freigegeben. Die Prozedur soll folgende Meldungen auf der Konsole ausgeben:

- "Auto fährt von Parkplatzznummer: Y", wobei Y für die Nummer steht, auf der das Auto stand.
- "Der Parkplatz ist leer!", falls auf dem Parkplatz keine Autos mehr stehen.

Hinweis:

Zufallszahlen in Java können wie folgt berechnet werden:

- `int x = (int) (Math.random() * 100) + 1`  
x erhält eine ganzzahlige Zufallszahl von 1 bis 100 (jeweils inklusive) zugewiesen

**Aufgabe 2: Poisson-Rätsel**

**8 + 4 = 12 Punkte**

Siméon Denis Poisson war ein bedeutender französischer Physiker und Mathematiker des 19. Jahrhunderts. Er entdeckte sein Interesse zur Mathematik durch ein einfaches Rätsel, welches er von einem Freund gestellt bekommen hatte:



"Zwei Freunde möchten einen 8 Liter Krug voll Wasser untereinander aufteilen. Zusätzlich zu dem großen Krug besitzen die Freunde noch einen leeren 3 Liter Krug und einen leeren 5 Liter Krug. Wie können die Beiden das Wasser umschütten, sodass jeder von ihnen *genau* 4 Liter Wasser bekommt?"

Da die Administratoren der Poolräume es bestimmt nicht gerne sehen, wenn Sie beim Testat anfangen, Wasser von einem Krug in den nächsten zu schütten, müssen Sie ein Programm implementieren, welches das oben genannte Rätsel "trocken" untersuchbar macht.

Verwenden Sie hierzu 3 Boole'sche Arrays, um die Größe der Krüge und den Füllzustand zu repräsentieren. Die Startbelegung der Arrays soll folgende Form haben:

8 Liter Krug = [true | true | true | true | true | true | true | true]  
5 Liter Krug = [false | false | false | false | false]  
3 Liter Krug = [false | false | false]

Entwerfen Sie eine Klasse mit dem Namen *PoissonRätsel* und schreiben Sie eine Prozedur *static void fülleKrugUm(boolean[] ausgangsKrug, boolean[] zielKrug)*,

welche den Inhalt des Ausgangskrugs in den Zielkrug umfüllt. Beachten Sie dabei die genaue Fragestellung des Rätsels und überlegen Sie sich vorher, was für Situationen beim Umfüllen auftreten können. Das folgende Beispiel zeigt eine mögliche Verwendung der Prozedur:

Krug A = [true | true | false]  
Krug B = [true | true | false | false | false]

Rufe Prozedur *fülleKrugUm(Krug A, Krug B)* auf:

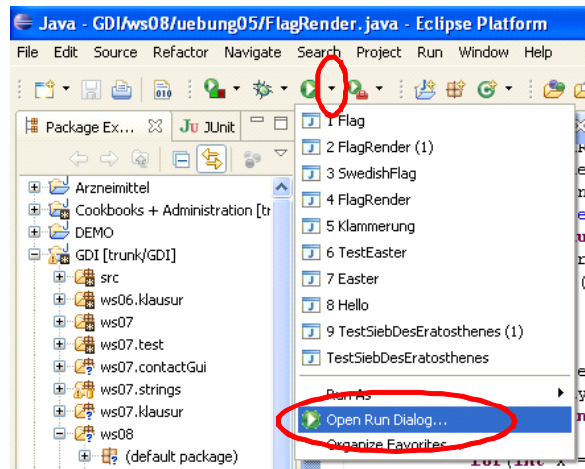
--> Krug A = [false | false | false]  
--> Krug B = [true | true | true | true | false]

Schreiben Sie ein Hauptmenü, das zu Beginn den Füllzustand der Krüge ausgibt und dem User die Möglichkeit gibt, nach Belieben Wasser von einem Krug in den anderen zu schütten, bis das Rätsel gelöst ist. Kann der User das Rätsel lösen, so bekommt er die Anzahl seiner benötigten Aktionen angezeigt.

### Zusatzaufgabe 3: Parameter von der Kommandozeile

12 Punkte

Java erlaubt es, Parameter von der Kommandozeile einzulesen, d.h. Parameter beim Programmstart zu übergeben (das wird normalerweise genutzt, um ein Programm beim Start zu konfigurieren). Die folgenden Screenshots zeigen, wo diese Parameter in Eclipse eingegeben werden (alternativ können Sie Ihr Programm natürlich auch von der Konsole starten).



- Als Parameter erlaubt sind beliebige Zeichenfolgen.
- Leerzeichen werden als Trenner zwischen verschiedenen Parametern aufgefasst.

Diese Parameter werden beim Programmstart automatisch in der Variable `args`, die als String-Array-Parameter der Methode `main` deklariert ist, abgelegt. Sie können darauf wie auf andere String-Arrays zugreifen.

Entwickeln Sie ein Programm `AnalyzeParameters`, das für alle seine Kommandozeilen-Parameter überprüft,

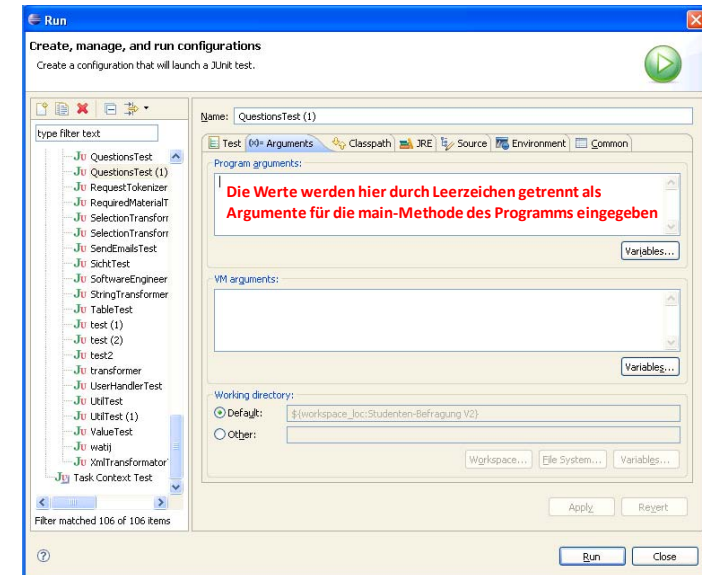
- ob es sich um eine ganze Zahl (ohne Vorzeichen!) handelt,
- ob es sich um einen Boole'schen Wert (`true` oder `false`) handelt,
- ob es sich um ein (einzelnes) Zeichen handelt oder
- ob es sich um einen String (bestehend aus mehr als einem Zeichen) handelt.

Jeder Parameter ist genau einer dieser Kategorien zuzuordnen!

Geben Sie das Ergebnis ihrer Prüfung jeweils mit der Nummer des Parameters und dem Parameter selbst aus.

#### Hinweise

- Vergeben Sie vernünftige Namen für Ihre Variablen und Parameter und vergessen Sie die Kommentare nicht!
- Die Aufgaben sind in Eclipse zu bearbeiten, legen Sie dafür ein Package `uebung07` an.



- Von allen Aufgaben sind Programmausdrucke (Listings) abzugeben, *keine* Ausdrücke von Testläufen. Die Aufgaben sind im Labor mit Eclipse vorzuführen.
- Erlaubt sind *MakeItSimple*-Funktionen (keine nicht besprochene Funktionalität aus der Java Standard Bibliothek) und das bisher erworbene Wissen aus den GDI-Vorlesungen. Fragen Sie, bevor Sie Java-Konstrukte verwenden, die noch nicht behandelt wurden! Zusätzliche eigene Hilfsfunktionen (keine fremden oder externen) sind ausdrücklich erlaubt.
- In den Laborstunden sollen Ihre Programme automatisch getestet werden. Damit Sie vorab prüfen können, ob Ihr jeweiliges Programm äußerlich korrekt ist, finden Sie im Wiki für jede geforderte Methode ein JUnit-Testprogramm.