

10. Übungsblatt

Ziel: Auseinandersetzung mit der Programmierung von Ausnahmefehlern (Exceptions) und generischen Klassen in Java.

1. Aufgabe (8 Punkte)

a) Programmieren Sie eine **Datum**-Klasse, die folgende zwei Schnittstellen implementiert:

```
public interface Comparable {
    public int compareTo(Object o);
}

public interface Date {

    public int getDay();
    public int getMonth();
    public int getYear();

    /* Das Datum verändert sich zum nächsten Tag */
    public void nextDay();

    /* Bei falschen Eingaben des Datums wird ein Ausnahme-Objekt
       erzeugt und das Datum bleibt unverändert, sonst wird das
       neue Datum gesetzt */
    public void setDate( int day, int month, int year )
                       throws IllegalArgumentException;

    /* testet, ob das Datum-Objekt selber (this) nach date ist */
    public boolean after( Date date );

    /* testet, ob das Datum-Objekt selber (this) vor date ist */
    public boolean before( Date date);

    /* gibt das Datum in Textform zurück */
    public String toString();
}
```

b) Sie müssen eine eigene **IllegalArgumentException**-Klasse programmieren. In Ihren **IllegalArgumentException**-Objekten sollen Sie genaue Angaben über das illegale Datum speichern, sodass beim Bedarf diese Information von aufrufenden Methoden verwendet werden kann.

c) Schreiben Sie eine **TestDatum**-Klasse, um alle Ihre Methoden und das Auftreten von Exceptions ausführlich zu testen.

2. Aufgabe (16 Punkte)

Programmieren Sie eine Warteschlange-Klasse, die folgende Schnittstelle implementiert:

```
public interface Queue<E> {  
  
    /* fügt ein Element am Ende der Warteschlange ein */  
    public void enqueue( E element );  
  
    /* das erste Element der Warteschlange wird entfernt und  
       als Rückgabewert der Operation zurückgegeben */  
    public E dequeue() throws EmptyQueueException;  
  
    /* die Referenz des ersten Elements der Warteschlange  
       wird zurückgegeben ohne diese zu entfernen */  
    public E head() throws EmptyQueueException;  
  
    /* überprüft, ob die Warteschlange leer ist */  
    public boolean empty();  
  
    /* die Warteschlange wird als Zeichenkette zurückgegeben */  
    public String toString();  
}
```

- a) Die Warteschlange sollte ein selbst implementiertes dynamisches Array (siehe Vorlesungsfolien) verwenden, um die Elemente der Warteschlange zu speichern.
- b) Eine Wrap-Around-Strategie soll für die Einfüge- und Löschoptionen benutzt werden (siehe Vorlesungsfolien).
- c) Die **EmptyQueueException**-Klasse ist nicht vorgegeben und soll als erstes programmiert werden.
- d) Testen Sie Ihre Warteschlange, indem Sie eine Reihe zufälliger **Root3Num**-Objekte oder **Rectangle**-Objekte einfügen und löschen.
- e) Schreiben Sie eine geeignete **print**-Methode für Ihre Warteschlange, um zwischen den Einfüge- und Löschoptionen die Warteschlange ausgeben zu können. In der **print**-Methode sollen nur die gespeicherten Objekte in der richtigen Reihenfolge ausgegeben werden.

3. Aufgabe (6 Punkte)

Betrachten Sie folgendes Python-Programm, wobei die Variablen `CONST_X` und `CONST_Y`, die für die Eingabe verwendet werden als beliebige Konstanten betrachtet werden.

```
# Eingabe
CONST_X = int(input("x = "))
CONST_Y = int(input("y = "))
# Variableninitialisierung
x = CONST_X
y = CONST_Y
p = 1
if x != 0 or y == 0:
    while y != 0:
        if y%2 != 0:
            p = p*x
        y = y//2
        x = x*x
else:
    p = 0
# Ausgabe
print(p)
```

a) Welche der folgenden Prädikate sind Schleifeninvarianten der **while**-Schleife?

- $p \cdot x^y = \text{CONST_X}^{\text{CONST_Y}} \wedge x > 0 \wedge y \neq 0$
- $p \cdot x^y = \text{CONST_X}^{\text{CONST_Y}} \wedge y \geq 0$
- $x^y = \text{CONST_X}^{\text{CONST_Y}} \wedge y \geq 0 \wedge x \neq 0$
- $p \cdot x^y = \text{CONST_X}^{\text{CONST_Y}} \wedge x > 0 \wedge y \geq 0$
- $p \cdot x^y = \text{CONST_X}^{\text{CONST_Y}} \wedge x \neq 0 \wedge y > 0 \wedge p \neq 0$

b) Begründen Sie Ihre Antwort (Mit der Begründung wird selbstverständlich ein kurzer Beweis mit Hilfe der Hoare-Regel gemeint).