

Aufgabenblatt

Exceptions, abstrakte Methoden

Programmierung Praktikum
Prof. Dr. Dirk Eisenbiegler
Hochschule Furtwangen

Aufgabe 1 - Stapel

In dieser Aufgabe soll eine Klasse implementiert werden, die einen Stapel realisiert, auf dem String-Werte abgelegt werden können.

Die Klasse soll einen Konstruktor und zwei Methoden zur Verfügung stellen:

- x Der Konstruktor hat keinen Parameter und soll einen leeren Stapel erzeugen.
- x Die Methode *push* legt einen String auf den Stapel.
- x Die Methode *pull* nimmt einen String vom Stapel und gibt ihn als Rückgabewert zurück.

Die Methode *pull* kann nur dann einen Wert vom Stapel nehmen, wenn der Stapel nicht leer ist. Wird die Methode *pull* bei einem leeren Stapel aufgerufen, so soll die Methode eine Exception werfen.

```
throw new Exception();
```

Hinweis: Verwenden Sie zur Implementierung der Klasse Stapel die Generic-Klasse *Vector*.

- A) Implementieren Sie die Klasse *Stapel* wie beschrieben. Testen Sie die Klasse!
- B) Überschreiben Sie in *Stapel* die Methode *toString()* so, dass die Stapelinhalte durch Kommata getrennt aufgezählt werden.
⇒ Beispiel: "auto,katze,hund"
- C) Überschreiben Sie in *Stapel* die Methode *equals()* in sinnvoller Weise: Vergleich zweier Stapel (paarweiser Vergleich mit der *equals()*-Methode der Elemente)
- D) Überschreiben Sie in *Stapel* die Methode *clone()* so, dass eine Kopie des Stapels erzeugt wird.

Aufgabe 2 - Funktionen

In dieser Aufgabe soll eine Klasse mit dem Namen *Funktion* implementiert werden, die Abbildungen von *double* nach *double* repräsentiert.

Die Klasse *Funktion* soll die folgende Methode enthalten.

```
public abstract double f (double x);
```

- A) Implementieren Sie eine abstrakte Klasse *Funktion* mit der abstrakten Methode *f*.
- B) Implementieren Sie eine Sohnklasse von *Funktion* mit dem Namen *QuadratischeFunktion*. Die Klasse *QuadratischeFunktion* hat zusätzlich drei Attribute *a*, *b* und *c* - alle vom Typ *double*. Die Klasse soll einen Konstruktor haben, dem die Werte dieser Attribute als Parameter übergeben werden. Die Methode *f* soll überschrieben werden. Die Funktion, die *f* realisieren soll, sei:

$$f(x) = a \cdot x^2 + b \cdot x + c$$

- C) Implementieren Sie in der Klasse *Funktion* eine Methode mit dem Namen *nullstellensuche*, die innerhalb eines Intervalls $[r, s]$ durch Intervallschachtelung nach einer Nullstelle sucht. Die Intervallschachtelung soll abbrechen, sobald ein *x* gefunden wird mit $|f(x)| \leq t$. Wird keine Nullstelle gefunden, so soll *r-1* zurückgegeben werden. Die Methode *nullstellensuche* hat drei Parameter: *r*, *s* und *t*. Alle drei Parameter sind vom Typ *double*. Der Rückgabewert ist ebenfalls vom Typ *double*.
Testen Sie die Methode, indem Sie eine Instanz von *QuadratischeFunktion* bilden.

- D) Implementieren Sie eine Sohnklasse von *Funktion* mit dem Namen *TrigonometrischeFunktion*. Die Klasse *TrigonometrischeFunktion* hat zusätzlich zwei Attribute *p* und *q* - beide vom Typ *double*. Die Klasse soll einen Konstruktor haben, dem die Werte dieser Attribute als Parameter übergeben werden. Die Funktion, die *f* realisieren soll, sei:

$$f(x) = \sin(p \cdot x + q)$$

- E) Implementieren Sie eine Sohnklasse von *Funktion* mit dem Namen *SummenFunktion*. Die Klasse *SummenFunktion* hat zusätzlich zwei Attribute *g* und *h* - beide vom Typ *Funktion*. Die Klasse soll einen Konstruktor haben, dem die Werte dieser Attribute als Parameter übergeben werden. Die Funktion, die *f* realisieren soll, sei:

$$f(x) = g(x) + h(x)$$