



Ausgabe: 2008-01-25

Abgabe: 2008-01-31

Objektorientierte Programmierung

14. Übungsblatt

Hinweis: Bitte alle Programme kommentieren! Programme ohne Kommentar werden mit 0 Punkten bewertet!

Aufgabe 1 (Punkte: 2)

Schreiben Sie ein kleines C#-Programm (ohne Klassen) mit folgender Funktionalität:

- Anlegen eines Feldes für 10 ganze Zahlen
- Lesen Sie die Zahlen von der Konsole ein. Implementieren Sie den Kopf der Schleife so, dass das Einlesen abgebrochen wird, wenn eine -99 eingegeben wurde. Dies soll die einzige Abbruchbedingung sein. Eine Überschreitung der Feldgrenze soll somit explizit möglich sein.
- Implementieren Sie innerhalb der Schleife eine Fehlerbehandlung, welche prüft, ob auch wirklich eine ganze Zahl eingegeben wurde. Falls dies nicht der Fall war, so soll die Eingabe für dieses Feldelement wiederholt werden und es soll die Meldung „Bitte nur ganze Zahlen eingeben!“ auf der Konsole erscheinen.
- Implementieren Sie weiterhin eine Fehlerbehandlung zur Bereichsüberschreitung beim Zugriff auf das i . Feldelement. Beim Überschreiten der Feldgrenze wird die Ausnahme `IndexOutOfRangeException` ausgelöst. In diesem Fall soll die Meldung „Feld ist voll, bitte die Eingabe mit -99 beenden“ ausgegeben werden.
- Geben Sie nach dem Einlesen das Feld vom ersten bis zum letzten Feldelement aus. Dabei sollen alle Zahlen in einer Zeile durch ein Leerzeichen getrennt ausgegeben werden.

Aufgabe 2 (Punkte: 2)

Erstellen Sie eine generische Klasse `Polynom`. Ein Polynom ist definiert als eine Summe von Vielfachen von Potenzen einer Variablen x .

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$$

Die Variablen x kann beliebige reelle Werte annehmen. Dagegen soll der Typ der Koeffizienten a_i generisch sein. Die Klasse besitzt somit nur das Koeffizientenfeld mit dem Namen `Koeff` als Attribut, auf welches nur die Methoden der Klasse selbst zugreifen können.

Implementieren Sie folgende öffentliche Methoden:

- `init` - bekommt den höchsten Exponenten als Parameter übergeben und initialisiert anhand dieses Wertes das Koeffizientenfeld mit Nullen. Benutzen Sie hierfür den Operator `default(Koeff[i] = default(T));`. Wendet man den Operator `default` auf eine generischen Wertetypen an, so wird 0 zurückgegeben und bei einem generischen Referenztypen wird `NULL` zurückgeliefert.

- **setKoeff** - bekommt als Parameter das Feld mit den Koeffizienten übergeben und überträgt die Werte in das Feld **Koeff**.
- **berechne** - bekommt den Wert **x** als Parameter übergeben und wertet das Polynom an der Stelle **x** aus. Das Ergebnis wird im Funktionswert zurückgegeben.

Testen Sie Ihre Klasse mit folgendem kleinem Hauptprogramm:

```
static void Main ()
{
    Polynom<int> gPoly = new Polynom<int>();
    Polynom<double> dPoly = new Polynom<double>();

    double [] dFeld = new double [5] { 3.4, -1.3, 5.9, 0.0, 67.23 };
    int [] gFeld = new int [5] { 3, 0, -2, 18, -45 };

    gPoly.init(4);
    dPoly.init(4);

    gPoly.setKoeff(gFeld);
    dPoly.setKoeff(dFeld);

    Console.WriteLine(gPoly.berechne(5));
    Console.WriteLine(dPoly.berechne(3.3));
}
```

Hinweis: Senden Sie nur Ihre Klasse **Polynom** ein, nicht das Hauptprogramm, den Namespace oder using-Direktiven. Einsendungen die somit nicht mit dem Schlüsselwort **class** beginnen und mit der zur Klasse **Polynom** gehörigen schließenden Klammer **}** enden, werden mit 0 Punkten bewertet.