

Matrikelnummer

Familienname, Vorname

## 1 Rekursion (8 Punkte)

Gegeben sei die folgende Methode:

```
class Aufgabe1
{
    public static int collatz (int n)
    {
        System.out.println(n);
        while (n%2==0) n/=2;
        return n==1 ? n : collatz (3*n+1);
    }

    public static void main(String[] args)
    {
        int ergebnis = collatz (37);
        System.out.println("Ende: " + ergebnis);
    }
}
```

### Fragen:

- Welche Ausgabe liefert das Programm?  
37  
112  
22  
34  
52  
40  
16  
*Ende: 1*
- Welche Rekursionstiefe wird erreicht?  
7
- Was ist ungewöhnlich an der Methode *collatz* im Vergleich zu meinen Ausführungen in der Vorlesung?  
*Der rekursive Aufruf erfolgt mit einem größeren Parameter.*

## 2 Zahlenumwandlung Hexadezimal zu Dezimal (12 Punkte)

Hexadezimalzahlen sind Zahlen, welche mit einem Stellenwert zur Basis 16 dargestellt werden. Für die sechs zusätzlichen notwendigen Ziffern für Stellenwerte von 10 bis 15 werden die Buchstaben A bis F als Zahlzeichen verwendet.

Beispiel:  $1FE_{16} = (1 \cdot 16 + 15) \cdot 16 + 14 = 510$

- a) Schreiben Sie eine Java-Funktion, welche als Parameter eine Hexadezimalzahl als String übergeben bekommt und als Ergebnis die entsprechende Dezimalzahl zurückgibt.

Dabei können zwei Fehler auftreten:

1. Die Zahl ist zu lang (mehr als 8 Hexadezimal-Stellen),
2. Die Zahl enthält ungültige Zeichen.

Beide Fehler sollen definierte Exceptions auslösen.

- b) Schreiben Sie ein kurzes Hauptprogramm, das die obige Funktion aufruft und alle auftretenden Fehler in geeigneter Weise behandelt.

```
class ArgTooLongException extends Exception {};  
class IllegalCharException extends Exception {};  
  
class Aufgabe2  
{  
    public static int hexconvert(String s)  
        throws ArgTooLongException, IllegalCharException  
    {  
        if (s.length()>8)  
            throw new ArgTooLongException();  
  
        int zahl=0, ziffer =0;  
        for (int i=0; i<s.length(); i++)  
        {  
            char c=s.charAt(i);  
            if (!(c>='0' && c<='9' || c>='A' && c<='F'))  
                throw new IllegalCharException();  
  
            ziffer = c<'A' ? c-'0' : c-'A'+10;  
            zahl=16*zahl+ziffer;  
        }  
        return zahl;  
    }  
}  
  
public static void main(String[] args)  
{  
    try  
    {  
        System.out.println(hexconvert("1FE"));  
    }  
    catch (ArgTooLongException e)  
    {  
        System.out.println("Error: Argument too long.");  
    }  
    catch (IllegalCharException e)  
    {  
        System.out.println("Error: Illegal input character.");  
    }  
}
```

### 3 Beantworten Sie die folgenden Fragen (10 Punkte):

1. Im Copy-Konstruktor einer implementierenden Klasse wird als Parameter der Interface-Typ angegeben. Ist dies zulässig und falls ja, welchen Vor- oder Nachteil hat das?

*Das ist zulässig. Der Vorteil ist, dass jede beliebige implementierende Klasse übergeben werden kann.*

2. Worauf müssen Sie bei der Programmierung einer rekursiven Methode achten?

*Beim rekursiven Aufruf muss ein kleineres Problem behandelt werden und es muss einen Ausstieg ohne weitere Rekursion geben (direkte Lösung).*

3. Was ist der Unterschied zwischen statischem und dynamischem Typ einer Variablen? Geben Sie ein Beispiel an.

*Der statische Typ wird bei der Variablendefinition festgelegt.*

*Der dynamische Typ wird zur Laufzeit anhand des gerade zugewiesenen Objekts bestimmt.*

*Beispiel:*     **interface** Complex { /\* siehe Vorlesung \*/ }  
                  **class** Cartesian **implements** Complex { /\* siehe Vorlesung \*/ }  
                  Complex c = **new** Cartesian(1,2);

*Der statische Typ der Variablen c ist Complex, der dynamische Typ ist Cartesian.*

4. Was versteht man unter einer Exception-Signatur und warum ist diese wichtig?

*Die Exception-Signatur gibt an, welche Exceptions von einer Methode ausgelöst werden können.*

*Dies ist für die aufrufende Methode wichtig, weil sie diese Exceptions entweder behandeln (catch) oder selbst in der eigenen Exception-Signatur angeben muss.*

5. Was passiert, wenn in einem Java-Programm eine Exception auftritt, für die der Programmierer keinen **catch**-Block vorgesehen hat?

*Die Exception wird solange nach außen durchgereicht, bis sie vom Laufzeitsystem behandelt wird. Dies führt zur Standard-Meldung der JVM.*

6. Welche Vor- und Nachteile haben die lineare bzw. binäre Suche?

*Die lineare Suche benötigt keinen sortierten Datenbestand, die binäre Suche ist wesentlich schneller, braucht aber sortierte Daten.*

7. In welchem Zusammenhang kommen die reservierten Wörter **final** und **finally** in Java vor?

***final** ist ein Modifier um Variablen vor Veränderung nach der Initialisierung zu schützen.*

***finally** ist ein Teil der Exception-Behandlung.*

*Anweisungen im **finally**-Block werden immer ausgeführt, egal ob eine Exception auftritt oder nicht.*

8. Beschreiben Sie in Worten die grundsätzliche Funktionsweise von Quicksort.

*1. Wahl eines Aufteilungselements.*

*2. Durch geschickte Vertauschungen kommt das Aufteilungselement an den richtigen Platz, davor stehen dann nur kleinere, dahinter nur größere Elemente.*

*3. Rekursiver Aufruf mit vorderer bzw. hinterer Teilfolge.*

9. Was versteht man unter Autoboxing bzw. Auto-Unboxing?

*Das ist der Mechanismus, welcher eine direkte Zuweisung eines primitiven Typs an den zugehörigen Wrapper-Typ ermöglicht und auch umgekehrt. Immer wenn ein primitiver Typ verwendet wird, aber eigentlich ein Referenztyp erforderlich ist, wird der primitive Typ stillschweigend in den entsprechenden Wrapper-Typ verpackt (boxing). Bei einer Zuweisung an eine primitive Variable wird er entsprechend wieder ausgepackt.*

10. Was bedeutet es, wenn bei einer Datenstruktur in der Beschreibung gesagt wird, dass das Einfügen eines neuen Elements die Zeitkomplexität  $\log(N)$  hat?

*Es bedeutet, dass die Zeit zum Einfügen eines neuen Elements vom Füllgrad abhängig ist.*

*Diese Zeit wächst mit dem Logarithmus der Anzahl  $N$  bereits enthaltener Elemente.*

*( Das ist eine typisches Wachstumsverhalten für binäre Suchbäume.)*