

Bachelor-Studiengang Informatik
Übungen zur Vorlesung "Grundlagen der Informatik" (GDI), WS 2010/2011

Übungsblatt 5: Funktionen und Rekursion

Ausgabe am: 4.11.2010
Abgabe am: 18.11.2010

Aufgabe 1: String-Funktionen

5 + 9 + 6 = 20 Punkte

Erstellen Sie ergänzend zu den String-Funktionen, die Sie in *MakeItSimple* vorfinden, drei weitere Funktionen. Erstellen Sie dazu ein Programm *StringExtensions* und implementieren Sie darin folgende Funktionen:

- String strToUpper(String original)
Diese Funktion akzeptiert einen String als Parameter und soll einen neuen String zurückgeben, in dem alle Buchstaben des ursprünglichen Strings in Großbuchstaben umgewandelt enthalten sind. Gewandelt werden sollen die Buchstaben a bis z und die Umlaute ä, ö und ü. Beispiel: "Viel Erfolg bei der Aufgabe!" soll umgewandelt werden in "VIEL ERFOLG BEI DER AUFGABE!".
- String[] strExplode(String original, char delimiter)
Diese Funktion soll eine Zeichenkette, die als Parameter an *original* übergeben wird, in Teil-Strings aufteilen, die in einem passend langen String-Array zurückgegeben werden. Dazu wird ein weiteres Argument benötigt, welches das Trennzeichen (*delimiter*) vorgibt, mit dem ein Teil-String vom anderen getrennt wird. Zum besseren Verständnis wieder ein kleines Beispiel: "Banane;Erdbeere;Apfel" soll über das Semikolon als Trennzeichen aufgetrennt werden in die Strings "Banane", "Erdbeere" und "Apfel". Diese Teil-Strings werden als Array an den Aufrufer zurückgegeben.
- int strScan(String original, String needle)
Diese Funktion soll den String-Parameter *original* nach einem Teil-String *needle* durchsuchen und dessen Position innerhalb von *original* zurückgeben (0 gilt dabei als erste Position). Falls *needle* nicht auffindbar ist, soll -1 zurückgegeben werden. Sollte der String mehr als einmal innerhalb von *original* vorkommen, wird die Position des ersten Treffers zurückgeliefert. Beispiel: In "Hello, world! This world is cool!" soll das Wort "world" aufgespürt werden; Ergebnis ist die Position 7.

Schreiben Sie eine *main*-Methode mit Testaufrufen dieser drei Funktionen, die deren Funktionalität unter Beweis stellen.

Aufgabe 2: ISBN-13 Prüzfiffer-Berechnung und Validierung

8 + 8 = 16 Punkte

Schreiben Sie eine Funktion zur Berechnung der Prüzfiffer einer ISBN-13 (Internationale Standardbuchnummer). Die Prüzfiffer ermöglicht es, eine ISBN-13-Nummer auf ihre Gültigkeit zu überprüfen und so Eingabe- oder Lesefehler zu erkennen. Schauen Sie sich dazu unter folgender URL die Formeln zur Berechnung der Prüzfiffer an: <http://tinyurl.com/3xovech>. Zusätzlich soll eine Prüffunktion geschrieben werden, die eine gültige ISBN-13 Nummer von einer ungültigen unterscheidet.



Verwenden Sie zur Berechnung und Prüfung der ISBN-Nummer folgende Signaturen:

- int computeISBN13CheckDigit(String isbn13)
Mögliche Eingaben sind ISBN-13 Nummern *ohne* die letzte Prüzfiffer; das letzte Trennzeichen darf, muss aber nicht vorhanden sein. Geliefert wird die Prüzfiffer für die ISBN oder -1, falls es sich nicht um eine gültige ISBN-13 handelt.
- boolean isISBN13Valid(String isbn13)
Die Funktion liefert genau dann *true*, wenn es sich um eine gültige ISBN-13-Nummer inklusive Prüzfiffer handelt, sonst *false*.

Erstellen Sie diese Funktionen in einem Programm *ISBNToolkit*, das auch eine *main*-Methode besitzt, in der die Funktionen mithilfe einiger Testfälle gründlich getestet werden.

Vorsicht:

ISBN Nummern können auch Bindestriche oder Leerzeichen enthalten (nach ISO 2108). Entfernen Sie diese am besten vorher oder ignorieren Sie sie. Sie dürfen davon ausgehen, dass das ISBN-13 Format, sprich die Position der Trennzeichen, richtig eingehalten wird.

Aufgabe 3: Zahlenumrechnung

8 Punkte

Entwickeln Sie ein Programm *NumberConverter* mit einer *rekursiven* Umrechnungsfunktion von Dezimal (Basis 10) auf Hexadezimal (Basis 16) mit natürlichen Zahlen. Verwenden Sie dazu folgende Signatur:

String convertDecToHex(int dec)

Beim Aufruf der Funktion soll eine dezimale Zahl als Integer eingegeben werden und als umgerechnete hexadezimale Zahl in Form einer Zeichenkette zurückgegeben werden; bei Übergabe einer negativen Zahl soll ein Leerstring zurückgegeben werden. Zur Verdeutlichung ein kurzes Beispiel: 6780 → "1A7C" (Beachten Sie die Großschreibweise der Ziffern A-F).

Erstellen Sie auch hier eine *main*-Methode mit guten Testaufrufen für Ihre Implementierung.

Einschränkung

Bei der Lösung dieser Aufgabe dürfen keine Schleifen verwendet werden.

Hinweise

- Die Aufgaben sind in Eclipse zu bearbeiten. Legen Sie für die Bearbeitung dieses Übungsblattes ein Paket (engl. Package) namens *uebung05* an.
- Von allen Aufgaben sind Listings abzugeben, *keine* Testläufe.
- Erlaubt sind *MakeItSimple*-Funktionen (keine nicht besprochene Funktionalität aus der Java Standard Bibliothek) und das bisher erworbene Wissen aus den GDI-Vorlesungen (keine Java-Konstrukte, 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 (was nicht heißt, dass es korrekt funktioniert!), finden Sie im Wiki für jedes Ihrer Programme ein JUnit-Testprogramm.