



Übungsblatt 6: Strings, Zeichen, Arrays, Schleifen

Ausgabe: 13.10.17

Abgabe: 20.11.17 12:00 Uhr elektronisch mittels Git

Aufgabe 1

Die Bacon-Chiffre

Hintergrund

Steganografie ist die Kunst des Versteckens einer geheimen Nachricht (Ursprungstext) in einem unscheinbaren Trägermedium. Das Ergebnis wird Steganogramm genannt. Ein sehr bekanntes klassisches Steganografiesystem ist z.B. das Schreiben mit Zitronensäure auf (das Medium) Papier.

Der Bacon-Chiffre ist ein auf Francis Bacon (1561 - 1626) zurückgehendes klassisches Steganographieverfahren, bei dem der Ursprungstext in einen Binärcode umgewandelt und anschließend in einen anderen Text z.B. mittels Groß- und Kleinschreibung untergebracht wird.

Vorgehensweise

1. Jedem Buchstaben des Alphabets wird ein fünfstelliger Binärcode zugeordnet, um die Nachricht zu kodieren:

Buchstabe	Code	Buchstabe	Code	Buchstabe	Code
A	kkkkk	I, J	kgkkk	R	gkkkk
B	kkkkg	K	kgkkg	S	gkkkg
C	kkkgk	L	kgkgk	T	gkkgk
D	kkkgg	M	kgkgg	U, V	gkkgg
E	kkgkk	N	kggkk	W	gkgkk
F	kkgkg	O	kggkg	X	gkgkg
G	kkggk	P	kgggk	Y	gkggk
H	kkggg	Q	kgggg	Z	gkggg

2. Diese Kodierung wird nun in einem anderen größeren Text, der als Trägermedium fungiert, mittels Anpassung der Groß- und Kleinschreibung versteckt. Für "k" wird ein **K**leinbuchstabe und für "g" ein **G**roßbuchstabe verwendet.¹
3. Die Entschlüsselung findet analog statt. Die Groß- und Kleinschreibung des Steganogramms wird untersucht und als k/g-Binärcode aufgeschrieben. Dieser Binärcode wird nun anhand der obigen Tabelle in den Ursprungstext zurückgewandelt.

Beispiel: Verstecken der Botschaft "Wikipedia"

Nachricht	W	i	k	i	p	e	d	i	a
Binärcodierung	g kgk	kk g	kkkk	g kgk	kgkkkk g ggk	gkkkk g kgk	gkkk	kkkk	k...
Trägermedium	Dies ist eine fast unauffällige Nachricht, oder etwa nicht?								
Steganogramm	DiEs ist eine FasT uNauffÄLLige NachriCHt, Oder etwa nicht?								

¹Bacon selbst verwendete eine weitaus unauffälligere Variante: Er behielt die Groß- und Kleinschreibung des Mediums bei und verwendete bei allen "g"-kodierte Zeichen eine leicht abgeänderte Variante des jeweiligen kleinen bzw. großen Buchstabens in seiner Handschrift.

Aufgabenstellung

Legen Sie eine Klasse `BaconKodiererImpl` an.

a) Schreiben Sie eine Methode

```
static String reinigeNachricht(String nachricht)
```

die aus einer geheimen Nachricht alle Zeichen entfernt, die keine Buchstaben sind (also auch Leerzeichen und Satzzeichen). Zusätzlich sollen alle Kleinbuchstaben in Großbuchstaben umgewandelt und deutsche Sonderzeichen (Ä, Ö, Ü, ß) sinnvoll ersetzt werden (AE, OE, UE, SS).

Beispiel:

```
Parameter: Hallo, schöne Welt!  
Ergebnis: HALLOSCHOENEWELT
```

Hinweis:

In der Klasse `String` finden Sie Methoden, um sich die Arbeit zu erleichtern (z.B. `toUpperCase`, `replaceAll`). Sie müssen diese Methoden aber nicht verwenden!

b) Schreiben Sie eine Methode

```
static String kodiereNachricht(String gereinigteNachricht)
```

die eine zuvor gereinigte Nachricht gemäß dem Binärcode aus der Tabelle kodiert. Für den Fall, dass die Methode falsch verwendet wird (dass nicht kodierbare Zeichen bzw. keine Buchstaben kodiert werden sollen), soll eine `PRException` ausgelöst werden.

Beispiel:

```
Parameter: HALLOWELT  
Ergebnis: kkgggkkkkkkkgkkgkkgkkgkkgkkgkkgkkgkkgkkgkkgkkgk
```

c) Schreiben Sie eine Methode

```
static String dekodiereNachricht(String binaerCode)
```

welche die Dekodierung analog zu Teilaufgabe b durchführt.
Zusätzliche Anforderungen:

- Ist ein 5er-Block unbekannt, soll “#” als Zeichen in das Ergebnis eingesetzt werden.
- Verwenden Sie bei den doppeldeutigen Codes “I, J” und “U, V” jeweils den ersten Buchstaben, also “I” bzw. “U”.
- Es ist möglich, dass die Länge des Binärcodes nicht durch 5 teilbar ist. In diesem Fall soll die Methode den unvollständigen 5er-Block am Ende stillschweigend ignorieren.

Beispiel:

```
Parameter: kkgggkkkkkkkgkkgkkgkkgkkgkkgkkgkkgkkgkkgkkgkkgk  
Ergebnis: HALLOWELT
```

d) Schreiben Sie eine Methode

```
static String versteckeNachricht(String nachricht, String traegerMedium)
```

die eine beliebige Nachricht in einem Trägermedium versteckt. Nutzen Sie hierfür die zuvor implementierten Methoden `reinigeNachricht()` und `kodiereNachricht()`. Die ursprüngliche Groß- und Kleinschreibung des Trägermediums geht logischerweise verloren.

Vorsicht:

- Leerzeichen und Satzzeichen (bzw. alle Nicht-Buchstaben) des Trägermediums sowie das “ß” sollen erhalten bleiben. Die Chiffrierung mittels Änderung der Groß- und Kleinschreibung soll dann beim nächsten Buchstaben fortgeführt werden.

- Die Methode soll eine `PRException` auslösen, wenn das Trägermedium zu kurz ist, so dass der Binär-code nicht vollständig in ihm untergebracht werden kann. (Das Trägermedium muss mindestens so viele Buchstaben (außer “ß”) haben, wie die Binärkodierung des Ursprungstextes lang ist).

Beispiel:

```
nachricht:      Hallo Welt!
traegerMedium: Es ist traurig zu denken, die Natur spricht und keiner hört zu.
Ergebnis:      es IST trauriG zU deNkEn, dIE nATuR sprichT uNd KeIneR hört zu.
```

e) Schreiben Sie eine Methode

```
static String zeigeNachricht(String steganogramm)
```

welche die Groß- und Kleinschreibung eines Steganogramms analysiert und so die darin kodierte Nachricht zurückliefert. Nutzen Sie die zuvor implementierte Methode `dekodiereNachricht()`.

Beispiel:

```
Parameter: es IST trauriG zU deNkEn, dIE nATuR sprichT uNd KeIneR hört zu.
Ergebnis: HALLOWELTA
```

f) Schreiben Sie eine Methode `main`, welche die Botschaft “Treffen uns um drei Uhr am Bahnhof!” im Text “Mein Name ist Juan Sanchez Villa-Lobos Ramirez, oberster Metallurge am Hofe König Karl V. von Spanien; ich wurde 896 vor Christus in Ägypten geboren und bin unsterblich seit 846 vor Christus.” versteckt, den Code ausgibt und anschließend die Botschaft wieder sichtbar macht.

Anmerkung:

Aufgrund der Überlänge des Trägermediums werden bei der Dechiffrierung zwei zusätzliche Zeichen ermittelt.

g) Erweitern Sie die `main`-Methode aus Teilaufgabe f, sodass eine interaktive konsolenbasierte Anwendung entsteht. Zu Beginn soll der Benutzer in einem Hauptmenü zwischen folgenden Punkten auswählen können:

1. Botschaft verstecken

Der Benutzer soll eine beliebige Nachricht sowie ein beliebiges Trägermedium eingeben können. Nach der Ausgabe des Steganogramms soll das Programm ins Hauptmenü zurückkehren.

2. Botschaft sichtbar machen

Der Benutzer soll ein beliebiges Steganogramm eingeben können. Nach der Ausgabe der kodierte Nachricht soll das Programm ins Hauptmenü zurückkehren.

3. Demo

Es soll das Beispiel aus Teilaufgabe f mit anschaulicher Darstellung und Zwischenschritten ausgegeben werden. Danach soll das Programm ins Hauptmenü zurückkehren.

4. Programm beenden

Das Programm soll beendet werden.

Allgemeines

- Die Aufgaben sind in Eclipse zu bearbeiten und beim Testat vorzuführen.
- Legen Sie für die Bearbeitung dieses Übungsblattes ein Paket namens `pr1.uebung06` an, in dem Sie Ihre Klassen anlegen.

- Erlaubt sind `MakeItSimple`-Funktionen (keine nicht besprochene Funktionalität aus der Java-Standard-Bibliothek) und das bisher erworbene Wissen aus den PR1-Vorlesungen. Sie müssen alle(!) von Ihnen verwendeten Konstrukte der Sprache sowie alle verwendeten Methoden, die nicht aus der Hilfsbibliothek `MakeItSimple` stammen, gut erklären und nötigenfalls im Testat selbst programmieren können!
- Sie geben ab, indem Sie vor Ende der Abgabefrist Ihr Projekt mit den lauffähigen Programmen des Übungsblattes in das Repository auf *ovid* pushen. Achten Sie darauf, ob Sie die Kontroll-E-Mail bekommen! Die letzte hochgeladene Version Ihres Projekts wird gewertet. Andere Abgaben, ob elektronisch oder auf Papier, zählen als **nicht abgegeben!**