

Vorlesung “Objektorientierte Programmierung” — 1. Programmierertest (Aufgabe A) —

Hinweise/Regeln:

Vermeiden Sie bitte jedes Verhalten, was als Täuschungsversuch (miss-)verstanden werden könnte. Wir müssten Sie sonst disqualifizieren, d.h. Sie hätten den Programmierertest nicht bestanden. Sie werden sowohl durch das Aufsichtspersonal im Rechnerpool überwacht, als auch auf den Rechnern per Programm/Aufzeichnung, sowie über das Netz.

- Bearbeitungsdauer: 70 Minuten (plus eventuell leichte Verlängerung).
- Es gibt nur “bestanden” oder “nicht bestanden”, keine Punkte für partiell korrekte Lösungen. Eventuell werden am Ende einige Minuten Bearbeitungszeit angehängt, wenn mehrere Teilnehmer knapp vor einer fertigen Lösung sind — aber nicht funktionierende Lösungen werden nicht akzeptiert, selbst wenn nur eine Kleinigkeit fehlt.
- Sie dürfen bis zu 3 Blätter “Spickzettel” / “Quick Reference” verwenden, sowie ein Buch (nicht zu groß, es muß noch auf den Tisch passen ohne den Nachbarn zu stören — ein Aktenordner wäre nicht akzeptabel). Sie dürfen außerdem ein Blatt leeres Papier und einen Stift verwenden, um sich Notizen zu machen.
- Mobiltelefone bitte ausschalten (oder mit der Aufsicht besprechen).
- Sie dürfen ein eigenes Notebook verwenden, wenn Sie es bisher in den Übungen benutzt haben. WLAN und sonstige Netzverbindungen müssen ausgeschaltet sein.
- Es ist nicht erlaubt, einen Web-Browser zu verwenden, ein EMail-Programm, oder sonstige Netzwerk-Zugriffe. Automatische Netzwerk-Zugriffe Ihrer Entwicklungsumgebung sind in Ordnung, aber ggf. geöffnete Webseiten schließen Sie bitte sofort.
- Die automatisch eingeblendeten Methoden-Signaturen und andere Hinweise gehören zur Entwicklungsumgebung, aber ansonsten verwenden Sie die Hilfe bitte nicht (insbesondere keine Suchfunktionen).
- Sie dürfen nicht versuchen, Dateien außerhalb Ihres Homeverzeichnisses abzulegen, oder auf Dateien außerhalb Ihres Homeverzeichnisses zuzugreifen (außer Dateien, die der Compiler bzw. die Entwicklungsumgebung verwendet).
- Auch auf vorhandene Dateien innerhalb Ihres Homeverzeichnisses dürfen Sie nicht zugreifen (außer Voreinstellungen etc. durch die Entwicklungsumgebung). Sie müssen daher das Programm vollständig neu eintippen, und nicht durch Modifikation von eventuell in Ihrem Homeverzeichnis bereits vorhandenen Dateien erstellen.
- Auch vorhandene Text-Dokumente (z.B. PDF) dürfen Sie nicht anzeigen.
- Die Homeverzeichnisse werden für Zugriffe von außen gesperrt. Falls Sie spezielle Zugriffsrechte gesetzt hatten, müssen Sie diese nach dem Test selbst wieder herstellen.

- Tauschen Sie keinesfalls irgendwelche Dinge mit den Nachbarn aus. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Sie müssen Mindestanforderungen an den Programmierstil erfüllen, z.B. entsprechend der Programmstruktur einrücken.
- Fragen Sie, wenn Ihnen die Aufgabe nicht klar ist! Wenn Sie an einer unverständlichen Fehlermeldung länger festhängen, können Sie probieren, zu fragen. Wir wollen aber nicht zu viele Tipps geben.
- Wenn Sie glauben, fertig zu sein, melden Sie sich bitte zur Kontrolle. Sollten wir noch einen Fehler finden, können Sie bis zur offiziellen Abgabefrist weiterarbeiten.

Aufgabe (Variante A)

Schreiben Sie ein Programm, welches einen String verschlüsselt wieder ausgibt. Der Klartext wird als Kommandozeilen-Parameter übergeben, ebenso eine Zahl, welche für das Verschlüsseln benötigt wird. Der Rahmen Ihres Hauptprogramms sieht somit folgendermaßen aus:

Listing 1: Hauptprogramm

```
1 public class Verschluesselung
2 {
3     public static void main(String [] args)
4     {
5         String alphabet = "abcdefghijklmnopqrstuvwxy";
6         String klartext = args[0];
7         int schluessel = Integer.parseInt(args[1]);
8         // ...
9     }
10 }
```

Die Zahl, welche als Kommandozeilen-Parameter übergeben wird, dient als Schlüssel und gibt an, durch welches Zeichen ein gegebenes Zeichen ersetzt wird. So wird das Zeichen ‘a’ mit einem Schlüssel von 3 durch ‘d’ ersetzt, ‘b’ durch ‘e’, ‘c’ durch ‘f’ usw. Wird dabei das Ende des Strings `alphabet` erreicht, so wird am Anfang des Strings `alphabet` fortgefahren, somit würde in unserem Beispiel ‘y’ durch ‘b’ und ‘z’ durch ‘c’ ersetzt werden. Enthält der Eingabestring ein Zeichen, welches nicht im String `alphabet` vorkommt, so wird dies unverschlüsselt mit ausgegeben. Der Aufruf

```
java Verschluesselung "Halloxyz" 3
```

sollte die Ausgabe `Hdoorabc` ergeben (der Großbuchstabe ‘H’ ist im String `alphabet` nicht enthalten und wird deswegen unverändert in die Ausgabe übernommen).

Ihr Programm sollte auch mit anderen Zeichenketten für `alphabet` funktionieren (nicht nur für die Liste der Kleinbuchstaben in alphabetischer Anordnung). Sie dürfen davon ausgehen, dass der Schlüssel (das zweite Argument) nicht-negativ ist, und kleiner als die Länge der Zeichenkette `alphabet` (im Beispiel also zwischen 0 und 25 liegt). Beachten Sie (bei Bedarf) die Tipps auf der nächsten Seite.

Tipps:

- Um auf ein Zeichen eines Strings zuzugreifen, gibt es die Methode

```
char charAt(int)
```

in der Klasse `String`. Z.B. liefert der Aufruf `alphabet.charAt(0)` das Zeichen 'a'.

- Mit der Methode

```
int length()
```

erhalten Sie die Länge eines Strings, z.B. liefert `alphabet.length()` den Wert 26.

- Sie können den Algorithmus für die Aufgabe frei wählen. Es ist aber recht offensichtlich, dass Sie jedes Zeichen der zu verschlüsselnden Zeichenkette nacheinander durchgehen müssen. Für dieses Zeichen müssen Sie dann die Position in der Zeichenkette `alphabet` bestimmen — oder feststellen, dass es in der Zeichenkette nicht enthalten ist. Das ginge z.B. auch mit einer Schleife. Falls es nicht enthalten ist, geben Sie es einfach aus. Falls es enthalten ist, berechnen Sie die neue Position und bestimmen das Ausgabezeichen mit `charAt`.