

Übungen zur Vorlesung
Objektorientierte Programmierung: Wintersemester 2021/2022

Nr. 7, Abgabe bis 13.12.2021

Hinweis: Sie können auf diesem Zettel mehr als die üblichen 12 Punkte erreichen, es zählen allerdings nur 12 zu Ihrem Soll. Alle weiteren erreichten Punkte werden als Bonuspunkte gewertet.

Aufgabe 7.1: Fair spielen

10 Punkte

In dieser Aufgabe wollen wir die Playfair-Verschlüsselung implementieren. Bei der Playfair-Verschlüsselung handelt es sich um eine Verschlüsselungsmethode aus dem Jahr 1854, die von Charles Wheatstone erfunden wurde. Zum Verschlüsseln eines Satzes oder Wortes wird eine 5 mal 5 Matrix aus Buchstaben zur Hand genommen.

Diese Matrix setzt sich zusammen aus einem Codewort und den restlichen Buchstaben des Alphabets, die nicht im Codewort vorgekommen sind. Wenn das Codewort einen Buchstaben doppelt enthält, wird nur das erste Vorkommen behalten und die restlichen Doppelungen aus dem Codewort gelöscht. Alle im Codewort vorkommenden J werden entfernt. Sobald das Codewort bereinigt ist, wird die Matrix mit den restlichen Buchstaben des Alphabets (in der normalen Reihenfolge) ausgenommen J aufgefüllt.

Beispiel: APFELSTRUDEL

A	P	F	E	L
S	T	R	U	D
B	C	G	H	I
K	M	N	O	Q
V	W	X	Y	Z

Das zu übersetzende Wort wird zunächst von Leerzeichen und Sonderzeichen bereinigt und dann in Buchstabenpaare aufgeteilt. Taucht in einem solchen Paar ein Buchstabe zwei Mal auf, wird ein X zwischen beide Buchstaben geschoben und der doppelte Buchstabe wird in das nächste Paar übernommen. Wenn das letzte Paar nur einen Buchstaben enthält, wird einfach ein A an das Wort angehängt. Beispielsweise würde MITTWOCH zu MI TX TW OC HA, aber OTTO könnte zu OT TO werden.

Jedes dieser Paare wird nun mithilfe der Matrix übersetzt. Hierfür gelten die folgenden Regeln:

1. Befinden sich beide Buchstaben des Paares in derselben Zeile der Matrix, wird der rechts-stehende Buchstabe in der Matrix verwendet. Wenn ein Buchstabe an der letzten Stelle stehen sollte, wird der Buchstabe an der ersten Stelle derselben Zeile verwendet.

Beispiel: Codewort *Apfelstrudel*

AP → PF, EL → LA

2. Befinden sich beide Buchstaben des Paares in derselben Spalte der Matrix, wird der darunter stehende Buchstabe verwendet. Ist der Originalbuchstabe der letzte in der Spalte, wird der erste Buchstabe derselben Spalte verwendet.

Beispiel: Codewort *Apfelstrudel*

PT → TC, MW → WP

3. Sollte keine der beiden Regeln zutreffen, wird der Buchstabe verwendet, der in derselben Zeile wie der zu kodierende in der Spalte des anderen Buchstaben steht.

Beispiel: Codewort *Apfelstrudel*

AU → ES, HK → BO

- a) Schreiben Sie zunächst eine Klasse *Playfair*. Legen Sie das private finale Feld *playfairSquare* vom Typ *Character[][]* an. Fügen Sie der Klasse außerdem ein privates, finales, statisches Feld vom Typ *String* mit dem Wert *"ABCDEFGHIJKLMNOPQRSTUVWXYZ"* hinzu. Achten Sie bei diesem Feld auf eine bezeichnende Benennung.
- b) Zum Befüllen von *playfairSquare* benötigen wir eine Hilfsmethode. Implementieren Sie `boolean characterInString(String s, Character character)`, die `true` zurückliefert, wenn ein übergebener *Character* *c* im übergebenen *String* *s* vorhanden ist.
- c) Implementieren Sie nun den Konstruktor `public Playfair(String codeword)`, der mithilfe Ihrer Hilfsfunktion das Quadrat anlegt. Gehen Sie dabei wie folgt vor:
1. Wandeln Sie das Codewort in Großbuchstaben um.
 2. Bereinigen Sie das neue Codewort von Doppelungen wie oben beschrieben.
 3. Fügen Sie an das daraus resultierende Codewort die fehlenden Buchstaben des Alphabets an, sodass das finale Wort 25 Zeichen enthält.
 4. Befüllen Sie *playfairSquare* mit dem fertigen Codewort.
- d) Schreiben Sie eine Methode `public void printSquare()`, mit der Sie *playfairSquare* auf der Konsole ausgeben können.
- e) Für das Enkodieren benötigen wir erneut ein wenig Hilfe. Implementieren Sie das folgende:
1. Eine Klasse *Position*, die eine x- und eine y-Koordinate als finale öffentliche Felder besitzt.
 2. Eine Methode `private Position findInSquare(Character c)`, die die *Position* eines *Characters* in *playfairSquare* als Ergebnis liefert.
 3. Eine Methode `private String cleanWord(String word)`, die das zu übersetzende Wort säubert. Die Methode soll zunächst alle Vorkommen von Sonderzeichen entfernen. Danach soll wie oben beschrieben eine Paarbildung stattfinden. Zu guter Letzt muss das Wort noch mit Großbuchstaben ersetzt werden.
- f) Implementieren Sie nun die Methode `public String encode(String word)`, die das Wort verschlüsselt zurückliefert. Verwenden Sie dabei Ihre Hilfsmethoden aus dem vorherigen Aufgabenteil.
- g) Testen Sie Ihre Verschlüsselung mit geeigneten Aufrufen innerhalb einer Main-Methode.

Hinweis: Sie dürfen das Codewort "Apfelstrudel" verwenden.

In dieser Aufgabe sollen Sie eine DVD-Sammlung nachstellen.

- a) Zuerst wollen wir die folgenden Klassen erstellen (stellen Sie sicher, dass Felder über korrekte Sichtbarkeiten verfügen und **final** sind, wenn nötig):
- *Library*
Eine *Library* hat einen Namen, ein Datum (aktueller Stand der Sammlung) und ein Array von *DVD*.
 - *DVD*
Eine *DVD* hat einen Titel, eine ISBN-Nummer, einen Regisseur und ein Array mit den wichtigsten Schauspielern (Headliner).
 - *Actor*
Ein *Actor* hat einen Vornamen, einen Nachnamen und ein Geburtsdatum.
 - *Director*
Ein *Director* hat einen Vornamen, einen Nachnamen und ein Geburtsdatum.

Hinweis: Verwenden Sie die Klasse `java.util.Date` als Datums-Klasse.

- b) Erstellen Sie sinnvolle Konstruktoren für die oben gegebenen Klassen.
c) Erstellen Sie für private Felder Methoden, um lesend auf die Felder zugreifen zu können.
d) Fügen Sie *Library* eine Methode **void** `addDVD` (*DVD* dvd) hinzu, die Ihrer Sammlung eine neue DVD hinzufügt.

Fügen Sie außerdem *Library* eine Methode **void** `removeDVD` (`String title`), die jede DVD mit dem Titel *title* aus Ihrer Sammlung entfernt. Achten Sie darauf, dass Sie das Datum Ihrer Sammlung anpassen, wenn Sie sie verändern.

Hinweis: Arrays können nach Ihrem Erstellen nicht mehr vergrößert werden. Erstellen Sie also ein neues Array mit einem Platz mehr und kopieren Sie die alten Werte in das neue Array.

- e) Fügen Sie *Library* eine Methode **boolean** `doIOwn` (*DVD* dvd) hinzu, die **true** als Ergebnis liefert, wenn sich eine DVD mit demselben Titel in Ihrer Sammlung befindet.
f) Schreiben Sie eine Main-Methode, in der Sie:
- Eine neue Sammlung von DVDs anlegen
 - Ihrer Sammlung mindestens 5 verschiedene DVDs über die Methode `addDVD` hinzufügen
 - 2 DVDs aus Ihrer Sammlung entfernen
 - Überprüfen, ob Sie eine der entfernten und eine der vorhandenen DVDs besitzen