htl donaustadt
Donaustadtstraße 45
1220 Wien
www.htl-donaustadt.at

Abteilungsvorstand
Informatik
Dipl.-Ing. Dr. Gerhard Lindner
T: +43 1 20105-400
F: +43 1 20105-103
lind@htl-donaustadt.at

htl donaustadt

zertifiziert nach ISO 14001
validiert nach EMAS

REIFE- UND DIPLOMPRÜFUNG 2015/16

AUFGABENSTELLUNG FÜR DIE NICHT STANDARDISIERTE KLAUSURPRÜFUNG FACHTHEORIE

Jahrgang:	5AHIF/5BHIF/5EHIF
Zugeteilter Pflichtgegenstand:	Programmieren und Software Engineering
Prüfungstermin:	Haupttermin 2016
Prüfungstag:	12. 05. 2016
Arbeitszeit:	5 Stunden (300 Minuten)
Prüfer:	Prof. Dipl.-Ing. Daniel Goldack Prof. Dipl.-Ing. Herbert Feichtinger
Seitenanzahl inkl. Deckblatt:	8 Blätter

Prof. Dipl.-Ing. Daniel Goldack

Prof. Dipl.-Ing. Herbert Feichtinger

AV Dipl.-Ing. Dr. Gerhard Lindner

Dipl.-Ing. Wilhelm Bonatz Schulleiter

LSI HR Dipl.-Ing. Judith Wessely-Kirschke

htl donaustadt
Donaustadtstraße 45
1220 Wien
www.htl-donaustadt.at

Abteilungsvorstand
Informatik
Dipl.-Ing. Dr. Gerhard Lindner
T: +43 1 20105-400
F: +43 1 20105-103
lind@htl-donaustadt.at

Inhaltsübersicht:

1. Allgemeine Aufgabenstellung: Air Traffic Control
   1.1. Objektorientierte Umsetzung eines Air Traffic Control Centers
   1.2. Air Traffic Control Center mit realitätsnahem Landeanflug
   1.3. Absicherung der Landeanflüge

2. Allgemeine Aufgabenstellung: Buchstaben vertauschen
   2.1. Objektorientierte Umsetzung
   2.2. Verschiedene Algorithmen entwickeln
   2.3. Algorithmen implementieren
   2.4. Bearbeitung eines Textes
   2.5. Weitere Überlegungen

Die gesamte Aufgabenstellung ist zu bearbeiten

1) Allgemeine Aufgabenstellung: Air Traffic Control (50%)

Im Rahmen der Flugsicherung der Anflüge auf einen Flughafen soll eine Überwachungssoftware entwickelt werden. Auf Basis einer vereinfachten Simulation soll der Landeanflug der Flugzeuge zum Flughafen radartechnisch abgesichert werden. Die Software soll mit den Daten des Primärradars gespeist werden, wobei Geschwindigkeit, Entfernung und Anflugrichtung (Winkel) verarbeitet werden.

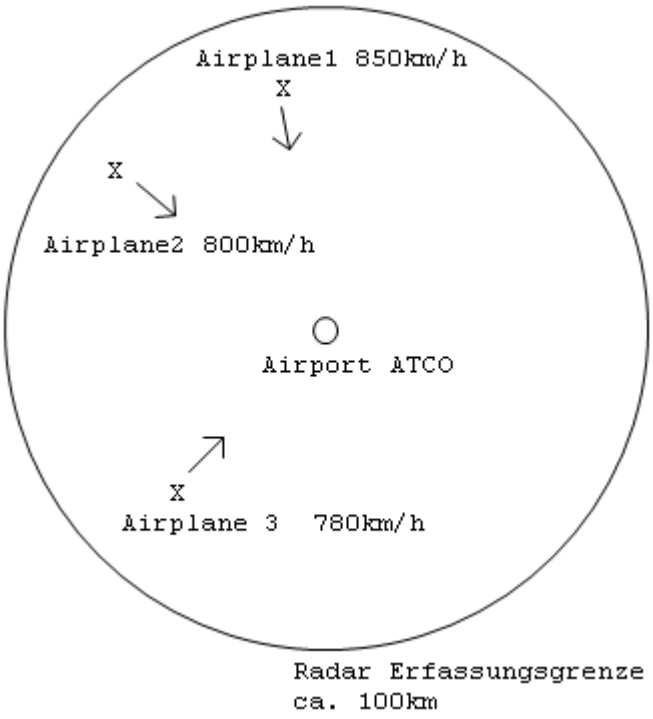

Abbildung1: Flughafen mit Radarabdeckung

Ein Flugzeug nähert sich bis zur Erfassung durch das Primärradar auf ca.100km dem Flughafen. Ab diesem Moment muss der Lotse sicherstellen, dass sich die anwesenden Flugzeuge nicht zu nahe kommen. In so einem Fall müsste eines davon in eine Warteschleife abdrehen.

Wenn ein Flugzeug sich bis auf 3km dem Tower des Flughafens genähert hat, kann es landen und geht aus dem Bereich des Lotsen heraus und wird von einem Landesystem übernommen.

htl donaustadt
Donaustadtstraße 45
1220 Wien
www.htl-donaustadt.at

Abteilungsvorstand
Informatik
Dipl.-Ing. Dr. Gerhard Lindner
T: +43 1 20105-400
F: +43 1 20105-103
lind@htl-donaustadt.at

Annahmen für die Simulation (einschließlich Vereinfachungen):

- Flugzeuge fliegen geradlinig (radial) auf den Flughafen im Zentrum zu.
- In der Software tauchen die Flugzeuge am Radarhorizont mit einer Anfangsgeschwindigkeit und einem Winkel auf.
- Es gibt 120 auf den Umfang gleichverteilte Korridore in Richtung Flughafen (alle 3°)
- Alle Flugzeuge im ganzen Überwachungsbereich haben die gleiche Reiseflughöhe (nur 2D Betrachtung zwischen 100km und 3km)
- Definition Klassengerüst:
  - `Airplane: Flugzeug mit seinen Reisedaten`
  - `AirTrafficControl: Flughafen mit Lotsenfunktion`
  - `AtcoConstants: vorgegebene Konstanten zur Berechnung`

```
Integer NSTEPS = 110;// Zahl der Durchläufe, Richtwert
Double TIMEINTERVALLSECONDS = 10.0; //Intervall eines Simulationsschritts
      (ein Berechnungsschritt entspricht 10 Sekunden in der Realität)
Double MINSPEED = 200.0;// km/h für Airplane
Double STARTDISTANCE = 100.0;// km Range of Radar
```

  - `MainSimulator: Hauptprogramm mit Gameloop,`
    `Simulationsschritte ohne Echtzeitverzögerung rechnen!`
  - *`weitere Klassen (z.B. zur Umrechnung von Koordinaten)`*
- Formel zur Umrechnung von Polar- in kartesische Koordinaten

`x= distance * cos( winkel )` und `y= distance * sin ( winkel )`
(beachten Sie bei Winkelberechnungen die Einheit Bogenmaß oder Grad!)

htl donaustadt
Donaustadtstraße 45
1220 Wien
www.htl-donaustadt.at

Abteilungsvorstand
Informatik
Dipl.-Ing. Dr. Gerhard Lindner
T: +43 1 20105-400
F: +43 1 20105-103
lind@htl-donaustadt.at

Aufgaben:

Hinweis: *Nach Bearbeiten einer Teilaufgabe muss der gesamte funktionierende workspace/Projekt (Verzeichnis kopieren) für diese Aufgabe zur Sicherung der Teilergebnisse gespeichert werden.*

Die eindeutige Kennzeichnung mit Familiennamen und Aufgabennummer ist notwendig!

1.1. Objektorientierte Umsetzung eines Air Traffic Control Centers:
Erstellen Sie die erforderlichen Klassen, simulieren Sie den Anflug für **ein** Flugzeug bis zur Landung mit **konstanter** Geschwindigkeit und protokollieren Sie mit den Statusmeldungen. Die Korridore (Anflugwinkel) können hier vernachlässigt werden. Die Ausgabe erfolgt via Log auf die Konsole gemäß anhängendem Beispiel.

- sichern!

1.2. Air Traffic Control Center mit realitätsnahem Landeanflug:
Erweitern Sie die Klassen so, dass mehrere Flugzeuge mit **variabler** Geschwindigkeit anfliegen. Für diese Teilaufgabe sollen **zwei** Flugzeuge simuliert werden. Für die Berechnung soll gelten: Die Abnahme der Geschwindigkeit soll linear mit dem Abstand vom Zentrum erfolgen. Eckpunkte sind die jeweilige Anfangsgeschwindigkeit (100%) und 0 km/h (0%) im Zentrum. Wobei eine Minimum-Geschwindigkeit von 200km/h nicht unterschritten werden darf, d.h. die Geschwindigkeit bleibt bei Erreichen dieser Schwelle konstant. Die Korridore (Anflugwinkel) und die Mindestabstände werden hier noch nicht betrachtet.
Die Ausgabe erfolgt via Log auf die Konsole gemäß anhängendem Beispiel.

- sichern!

1.3. Absicherung der Landeanflüge:
Erweitern Sie die Klassen so, dass **mehrere** Flugzeuge mit **variabler** Geschwindigkeit anfliegen (lineare Drosselung bis auf 200km/h wie in Aufgabe b).

Führen Sie eine Distanzprüfung ein, dass kein Flugzeug einem anderen zu nahe kommt (>3km).

Bei Annäherung auf <=3km zwischen zwei Flugzeugen muss eines davon in die Warteschleife. In der Simulation erfolgt dies durch direktes Setzen dieses Flugzeuges in einen freien Korridor (Anflugwinkel) auf 100km Distanz bei Beibehaltung der momentanen Geschwindigkeit.

Die Flugzeuge werden nun durch eine Generator-Methode mit zufälligen Anfangswerten erzeugt (Mindestabstände müssen eingehalten werden). Die Obergrenze sind 30 Flugzeuge im Radarbereich. Die Ausgabe erfolgt via Log auf die Konsole.

- sichern!

htl donaustadt
Donaustadtstraße 45
1220 Wien
www.htl-donaustadt.at

Abteilungsvorstand
Informatik
Dipl.-Ing. Dr. Gerhard Lindner
T: +43 1 20105-400
F: +43 1 20105-103
lind@htl-donaustadt.at

Beispiel LOG Ausgabe zu 1.1):

Anflug eines Flugzeugs mit konstanter Geschwindigkeit nach dem Auftauchen am 100km Horizont:

```
Status after Step ---------------
Airplane [speed=800km/h, distance=97.7km ]
Status after Step ---------------
Airplane [speed=800km/h, distance=95.5km ]
Status after Step ---------------
Airplane [speed=800km/h, distance=93.3km ]
Status after Step ---------------
Airplane [speed=800km/h, distance=91.1km ]
      .
Status after Step ---------------
Airplane [speed=800km/h, distance=6.6km ]
Status after Step ---------------
Airplane [speed=800km/h, distance=4.4km ]
Airplane to land [speed=800km/h, distance=2.2km ]
no Further Airplane
```

Flugzeug wurde bei 3km Distanz vom Landesystem übernommen und wird ab hier nicht mehr betrachtet.

Beispiel LOG Ausgabe zu 1.2):

Anflug mehrerer Flugzeuge mit variabler Geschwindigkeit nach dem Auftauchen am 100km Horizont:

```
Status after Step ---------------
Airplane [speed=782km/h, distance=97.7km ]
Airplane [speed=686km/h, distance=98.0km ]
Status after Step ---------------
Airplane [speed=764km/h, distance=95.6km ]
Airplane [speed=673km/h, distance=96.1km ]
Status after Step ---------------
Airplane [speed=747km/h, distance=93.4km ]
Airplane [speed=659km/h, distance=94.2km ]
Status after Step ---------------
Airplane [speed=731km/h, distance=91.4km ]
Airplane [speed=647km/h, distance=92.4km ]
      .
Airplane [speed=200km/h, distance=3.7km ]
Airplane [speed=200km/h, distance=8.4km ]
Status after Step ---------------
Airplane [speed=200km/h, distance=3.1km ]
Airplane [speed=200km/h, distance=7.9km ]
Airplane to land [speed=200km/h, distance=2.6km ]
Status after Step ---------------
Airplane [speed=200km/h, distance=7.3km ]
Status after Step ---------------
Airplane [speed=200km/h, distance=6.7km ]
Status after Step ---------------
Airplane [speed=200km/h, distance=6.2km ]
Status after Step ---------------
Airplane [speed=200km/h, distance=5.6km ]
Status after Step ---------------
Airplane [speed=200km/h, distance=5.1km ]
Status after Step ---------------
Airplane [speed=200km/h, distance=4.5km ]
Status after Step ---------------
Airplane [speed=200km/h, distance=4.0km ]
Status after Step ---------------
Airplane [speed=200km/h, distance=3.4km ]
Airplane to land [speed=200km/h, distance=2.9km ]
no Further Airplane
```

htl donaustadt
Donaustadtstraße 45
1220 Wien
www.htl-donaustadt.at

Abteilungsvorstand
Informatik
Dipl.-Ing. Dr. Gerhard Lindner
T: +43 1 20105-400
F: +43 1 20105-103
lind@htl-donaustadt.at

2. Allgemeine Aufgabenstellung: Buchstaben vertauschen (50%)

Sie arbeiten in der IT-Abteilung eines Marketing- und Werbeunternehmens. In dieser Funktion erhalten Sie den Auftrag ein Programm zu entwickeln, das lustige Sprüche für Plakate erzeugt. Eine Aufgabe darin ist das Vertauschen von Buchstaben in Worten.

Die grundlegende Idee dabei ist:
Der erste und der letzte Buchstabe eines Wortes bleiben unverändert. Alle anderen Buchstaben im Wort müssen vertauscht werden.

Ein Beispiel dazu:

Originaltext:

> Gemäß einer Studie einer englischen Universität ist es nicht wichtig, in welcher Reihenfolge die Buchstaben in einem Wort sind. Das einzige, was wichtig ist, ist, dass der erste und der letzte Buchstabe an der richtigen Position sind.

Geänderter Text:

> Gmäeß eneir Sutide eneir elgnihcesn Uvinisterät ist es nchit witihcg, in wlecehr Rneflogheie die Bustachuebn in eneim Wrot snid. Das ezniige, was wcthiig ist, ist, dass der estre und der leztte Bstabchue an der ritihcegn Pstoiion snid.

Annahmen und Vereinfachungen:
Texte werden aus Dateien eingelesen. Der Inhalt einer Datei passt zur Gänze in einen String im Hauptspeicher.
Die Texte enthalten keine Ziffern.
Es gibt keine mit Bindestrich zusammengesetzten Wörter im Text.

Hinweise für die Umsetzung:

- Name des Projekts: z.B. *Maier_BuchstabenVertauschen* (Verwenden Sie Ihren Familiennamen für die leere Solution bzw. workspace und Projekt).
- Wenn Sie Daten am Bildschirm ausgeben, erstellen Sie am Ende jeder Aufgabe einen Screenshot des Ausgabefensters. Speichern Sie dieses im Unterverzeichnis *Screenshots* mit einem eindeutigen Dateinamen mit Bezug auf die Aufgabe.
- Nach Bearbeitung jeder Teilaufgabe muss das gesamte Verzeichnis der Solution/workspace inklusive der dazugehörigen Dateien gesichert werden. Ändern Sie am Sicherungsmedium den Namen des Verzeichnisses auf *Maier_BuchstabenVertauschen_Aufgabennummer*.
- Erstellen Sie zu jeder wichtigen Methode einen Blockkommentar (XML/JAVADOC) im Sourcecode.

htl donaustadt
Donaustadtstraße 45
1220 Wien
www.htl-donaustadt.at

Abteilungsvorstand
Informatik
Dipl.-Ing. Dr. Gerhard Lindner
T: +43 1 20105-400
F: +43 1 20105-103
lind@htl-donaustadt.at

Aufgaben

2.1 Objektorientierte Umsetzung

Erstellen Sie eine Konsolen-Anwendung namens *TestConsole* und eine Klassenbibliothek/Package namens MyLib/mylib für die eigentliche Programmlogik.

Entwickeln Sie eine beliebige Methode für das Vertauschen von Buchstaben in **einem** Wort. Testen Sie diese mit mehreren Wörtern unterschiedlicher Länge, die in der Konsolen-Anwendung fix eingebaut sind. Die Wörter liegen als Beilage geduckt und elektronisch vor.

Erstellen Sie eine zweispaltige Ausgabe am Bildschirm:
Originales Wort Geändertes Wort

Screenshot erstellen, alles sichern

2.2 Verschiedene Algorithmen entwickeln

Erstellen Sie eine Textdatei namens ***Design*** in ihrem Projektverzeichnis.

Beschreiben Sie darin tabellarisch mehrere Algorithmen zum Vertauschen von Buchstaben in einem Wort. Auf welche verschiedenen Arten können Buchstaben getauscht werden?

Wie verhält sich die Performance dieser Algorithmen?
Wie sieht die Beschreibung in der O-Notation aus?

Name	Beschreibung Algorithmus	Aufwand Algorithmus

alles sichern

2.3 Algorithmen implementieren

In Teilaufgabe 2.1) haben Sie bereits einen Algorithmus implementiert. Codieren Sie nun eine zweite Methode zum Vertauschen von Buchstaben in einem Wort.
Testen Sie mit den vorhandenen Testdaten.

Erstellen Sie eine dreispaltige Ausgabe am Bildschirm:
Originales Wort Geändertes Wort Variante_1 Geändertes Wort Variante_2

Screenshot erstellen, alles sichern

htl donaustadt
Donaustadtstraße 45
1220 Wien
www.htl-donaustadt.at

Abteilungsvorstand
Informatik
Dipl.-Ing. Dr. Gerhard Lindner
T: +43 1 20105-400
F: +43 1 20105-103
lind@htl-donaustadt.at

2.4 Bearbeitung eines Textes

Erweitern Sie die Class Library um eine Methode für das Vertauschen von Buchstaben in Wörtern aus einem Satz und testen Sie diese.

Erstellen Sie dazu als Testdaten in der Konsolen- Anwendung die drei Sätze:

- Gemäß einer Studie einer englischen Universität ist es nicht wichtig, in welcher Reihenfolge die Buchstaben in einem Wort sind.
- Das einzige, was wichtig ist, ist, dass der erste und der letzte Buchstabe an der richtigen Position sind.
- The quick brown fox jumps over the lazy dog.

Erstellen Sie eine dreispaltige Ausgabe am Bildschirm mit einem Absatz pro Satz:

Satz i:
Originales Wort_1 Geändertes Wort_1 Variante_1 Geändertes Wort_1 Variante_2
Originales Wort_2 Geändertes Wort_2 Variante_1 Geändertes Wort_2 Variante_2
….

Satz j:
Originales Wort_1 Geändertes Wort_1 Variante_1 Geändertes Wort_1 Variante_2
Originales Wort_2 Geändertes Wort_2 Variante_1 Geändertes Wort_2 Variante_2
….

Ergänzen Sie in der Datei ***Design*** mögliche Verbesserungen der Algorithmen.

Screenshot erstellen, alles sichern

2.5 Weitere Überlegungen

Beantworten Sie in der Textdatei ***Design*** folgende Fragen:

i. Erläutern Sie, wie Texte aus Textdateien eingelesen werden können, wenn diese Texte nicht in einen String bzw. nicht in den Hauptspeicher passen? Wie können diese bearbeitet werden?

ii. Benennen Sie, wie beim Programmstart der Name der Input-Datei übergeben werden kann?

iii. Erklären Sie, ob bzw. wie bei Programmende ein Status an das aufrufende Programm/Betriebssystem zurückgegeben werden kann?

iv. Analysieren Sie, wie das Zerlegen von langen Texten mit sehr vielen Sätzen und Wörtern beschleunigt werden kann? Beschreiben Sie alle Varianten, die Ihnen einfallen.
Untersuchen Sie, ob „Amdahl's Law" auf diese Aufgabenstellung angewendet werden kann?

v. Beschreiben Sie, wie Sie in der Praxis die einzelnen Methoden der Bibliothek testen würden, sodass diese Tests beliebig oft wiederholt werden können?

- alles sichern!