



RWTH AACHEN
Lehrstuhl für Informatik 10 (Medieninformatik)
Programmierung für Alle (Java)

Weihnachtsübung 7

Abgabe der Lösungszettel: 16. Januar 2007 vor der Übung

**Deadline für die Abgabe der Programme per *E-Mail* beim Tutor:
16. Januar 2007, 8:00 Uhr**

Tragen Sie hier Ihre **Gruppennummer** ein:

Tragen Sie hier den **Namen Ihres Tutors** ein:

Die Abgabe der Übung erfolgt in **Dreiergruppen**. Tragen Sie dazu alle Namen und Matrikelnummern Ihrer Gruppenmitglieder hier ein. Der / Die erste in der Tabelle gibt die Übung beim Tutor ab.

NAMEN	MATRIKELNUMMERN

Punktetabelle für die Korrektur:

	Punkte
1 (7)	
2 (6)	
3 (7)	
Total (20)	

Hinweise zum 2. Übungsblock

Diese 7. Übung ist die erste Übung des zweiten Übungsblocks. Sie müssen in allen folgenden Übungen mindestens 50% der Übungspunkte erreichen UND die Teilklausur 1 bestanden haben, damit Sie zu der Schein-/Bachelorklausur Teil 2 am 13. Februar 2006 zugelassen werden.

Die Anmeldung zur **Bachelorklausur Teil 2** fand für alle Bachelorstudenten im Dezember 2006 statt.

Die Anmeldung zu der **Scheinklausur Teil 2** findet für alle anderen Studenten über die Lehrstuhlwebseite statt. Der Anmeldezeitraum ist 29.01.2007 – 02.02.2007 um 17:00 Uhr.

Die Ergebnisse der Schein-/Bachelorklausur Teil 1 werden für diejenigen Studenten, die sich für die Bachelorklausur Teil 2 und für die Vordiplom-/ Zwischenprüfungsklausur beim ZPA anmelden mussten, an das ZPA weitergeleitet.

Hinweise zur Abgabe der aktuellen Übung

Am Dienstag, den 09. Januar 2007, erscheint die Übung 8. Die Deadline für die aktuelle Weihnachtsübung und für die Übungen 8 ist der **16. Januar 2007**.

Hinweise zur Übungsabgabe

Bitte geben Sie zu jeder Übung das *Deckblatt*, angeheftet vor Ihren Lösungen, mit ab. *Programmieraufgaben* werden per E-Mail **und** der ausgedruckte Code vor Übungsbeginn beim Tutor abgegeben. *Verständnisaufgaben* werden **handschriftlich** beim Tutor abgegeben.

Sollten Sie Probleme oder Fragen haben, wenden Sie sich bitte an Ihren Tutor oder besuchen Sie die *Betreuungsstunden* im Lila Raum (4U15) im ZIP-Pool der Informatik: Donnerstag von 16:00 bis 18:00 und Freitag von 10:00 bis 12:00.



1. Ausnahmebehandlung in `Benutzereingabe.java` (Programmieraufgabe)

Die Methode `String getBenutzereingabe (String prompt) {...}` der Klasse `Benutzereingabe` liefert eine Tastatureingabe als `String` zurück.

- (a) [3 Punkte] Schreiben Sie ein Programm `EingabeTest.java`, das nacheinander zwei Eingaben von einem Benutzer erwartet: einen `String` und eine `int-Zahl`. Geben Sie die Eingaben des Benutzers auf der Kommandozeile aus. Wenn der `String` leer ist, dann soll "Ihr `String` ist leer." ausgegeben werden.

Beispiel:

Geben Sie einen `String` ein: xyz

Ihr `String` ist xyz.

Geben Sie eine Zahl ein: 3

Ihre Zahl ist 3.

- (b) [4 Punkte] Probieren Sie es aus: Wenn Sie als zweite Eingabe keine `Ganzzahl` eingeben, dann erhalten Sie eine `Exception`. Verbessern Sie die Klasse `Benutzereingabe.java`, damit die `Exception` abgefangen wird.

Ändern Sie zuerst den Namen der Methode `getBenutzereingabe` in `getString` um, damit er aussagekräftiger ist. Schreiben Sie nun eine Methode `getInt (String prompt)`, die eine `int-Zahl` zurückliefert. Wenn der Benutzer keine `int-Zahl` eingibt, dann geben Sie die Meldung "Es ist eine `Exception` aufgetreten: Sie haben keine Zahl eingegeben!" aus und fordern den Benutzer erneut auf, eine Zahl einzugeben. Wiederholen Sie die Fehlermeldung und die Aufforderung, bis der Benutzer eine gültige Zahl eingibt.

Beispiel:

Geben Sie einen `String` ein: xyz

Ihr `String` ist xyz.

Geben Sie eine Zahl ein: xyz

Es ist eine `Exception` aufgetreten: Sie haben keine Zahl eingegeben!

Geben Sie eine Zahl ein: ...

Tipp: Rufen Sie die Methode `getString(String prompt)` aus `getInt(String prompt)` auf. Duplizieren Sie keinen Code!

Benutzen Sie ab jetzt für die folgenden Aufgaben die verbesserte Klasse `Benutzereingabe.java`, damit die Programme bei einer fehlerhaften Eingabe nicht mehr abstürzen.

2. [6 Punkte] **Formatierung und Datumsmanipulationen** (Programmieraufgabe)

Schreiben Sie ein Programm *Geburtstagsinfos.java*, das ein Geburtstagsdatum einliest und die entsprechenden Wochentage jedes Geburtstags bis zum aktuellen Jahr ausgibt.

Beispiel:

An welchem Tag sind Sie geboren? 13

In welchem Monat sind Sie geboren? 2

In welchem Jahr sind Sie geboren? w

Es ist eine Exception aufgetreten: Sie haben keine Zahl eingegeben!

In welchem Jahr sind Sie geboren? 1980

Der 13.02.1980 ist ein Mittwoch.

Der 13.02.1981 ist ein Freitag.

...

Der 13.02.2007 ist ein Dienstag.

Benutzen Sie ein Objekt der Klasse *Calendar*, um Operationen auf ein Datum durchzuführen und das aktuelle Jahr abzufragen. Die Methode *String.format (...)*, können Sie verwenden, um ein Datum korrekt zu formatieren (evtl. mit einer führenden Null für den Tag und den Monat). Informationen zur Klasse *Calendar* und zum Formatieren eines Datums finden Sie im Internet unter:

<http://java.sun.com/j2se/1.5.0/docs/api/java/util/Calendar.html>

<http://java.sun.com/j2se/1.5.0/docs/api/java/util/Formatter.html#syntax>

Tipp: Verwenden Sie *%s* als *Typ* im Formatspezifizierer, um den Wochentag als String an die Methode *String.format (...)* zu übergeben.

Hinweis: Die Variablen der Klasse *Calendar* sind *statisch*. Achten Sie auf die korrekte Schreibweise, wenn Sie statische Variablen verwenden und auf sie zugreifen.

3. Konstruktoren und lokale Referenzvariablen

Lotto Teil 3: (Programmieraufgabe)

Benutzen Sie für die Lösung das beigefügte Programmgerüst.

- (a) [1 Punkt] Ändern Sie die Klasse *LottoSpiel*, damit es die verbesserte Klasse *Benutzereingabe* mit Ausnahmebehandlung aus Aufgabe 1 verwendet.
- (b) [1 Punkt] Schreiben Sie einen Konstruktor für *LottoTipp*. Der Konstruktor soll den Namen und den Tipp eines Spielers initialisieren. Ändern Sie die Klasse *LottoSpiel*, damit es nur noch Konstruktoren für die LottoTipps verwendet.
Tipp: Um doppelten Code zu vermeiden, können Sie im Konstruktor die Setter-Methoden aufrufen.
- (c) [5 Punkte] Schreiben Sie für die *LottoGesellschaft* eine Methode *getTreffer(...)*, die den Tipp eines Spielers erwartet und die Treffer in einem int-Array zurückliefert. Nach der Ziehung der Gewinnzahlen sollen für alle Spieler, die mindestens 3 Treffer haben, die Treffer ausgegeben werden.

Beispiel:

... Programmablauf wie bisher ...

Tom hat 4 Treffer: 18 24 39 44

Hans hat 3 Treffer: 26 39 44

Hinweis: Wenn ein Spieler z.B. genau drei Zahlen richtig getippt hat, dann sollen auch *nur diese drei Zahlen* in einem Array passender Länge zurückgeliefert werden.