



TI1 P1 WS17/18

Vorbemerkung zum Praktikum

- Alle Teammitglieder müssen bei jedem Laborterminal ausreichend vorbereitet sein.
- Es gibt für jeden Studenten 7 Labortermine und 7 Aufgabenzettel.
- Im Labor herrscht Anwesenheitspflicht. Sollten Sie aus irgendeinem Grund verhindert sein klären Sie dies im Vorfeld. Falls Sie am Laborterminal erkranken sollten, schicken Sie sofort(!) eine Email.
- Sie arbeiten in **Zweier-Teams**. Ein Zweier-Team besteht min. für einen Laborterminal.
- Sie müssen über das ganze Labor gesehen mit min. drei verschiedenen Teampartnern zusammenarbeiten.
- Sie müssen alle Aufgaben eines Aufgabenzettels bearbeiten und in der Lage sein, die Lösung zu präsentieren. Einzig Aufgaben, die explizit als optional gekennzeichnet sind, müssen nicht zwingend bearbeitet werden.
- **Jedes Teammitglied muss jeden "Teil"/jede Zeile der vorgestellten Lösung jederzeit rechtfertigen können.**
- **Jedes Teammitglied muss in der Lage sein, kurzfristig kleine Änderungen der Aufgabenstellung implementieren zu können.**
- Die Abnahme erfolgt immer auf den Labor-Rechnern.
- Legen auf dem Z-Laufwerk ein Directory P1P. In diesem Directory sollen sich der Eclipse-Workspace mit seinen Projekten befinden und damit die Lösungen zu den jeweiligen Aufgaben. Jede Aufgabe soll in einem eigenständigen Projekt gelöst werden, sofern nicht explizit anders eingefordert.
- Bei der Abnahme müssen nicht immer alle Aufgaben kontrolliert werden. U.U. werden Ihnen Aufgaben erlassen. Sie müssen aber trotzdem immer alle Aufgaben lösen und alle vorherigen Punkte erfüllen bzw. beachten. U.U. müssen Ihre Lösung auch zu einem nachfolgenden/anderen Termin noch einmal vorführen.

Vorbemerkung zum 1.Aufgabenzettel

- Primärziel dieses Aufgabenzettels ist es, dass Sie den Umgang mit Eclipse üben. Zum 2.Laborterminal wird vorausgesetzt werden, dass Sie ausreichend Eclipse beherrschen um die Aufgaben lösen zu können.
- Es wird (meist) Source-Code gestellt.
 - Sie müssen diesen Source-Code in Eclipse einbinden können.
 - Sie müssen die Zip-Dateien entpacken können.
 - Sie müssen wissen wo Ihr Workspace liegt bzw. dessen Position einstellen können.
 - Der Workspace soll auf dem "Z-Laufwerk" liegen. Der Pfad soll einen Folder enthalten der eindeutig kennzeichnet zu welcher Veranstaltung der Workspace gehört (Z.B.: TI1_P1_s17w)

Achtung!:

Eclipse hat sehr viele eingebaute Automatismen um erfahrene SW-Entwickler optimal zu unterstützen. Die Erfahrung lehrt, dass dies für Anfänger kontraproduktiv ist. Es ist in Ihrem ureigensten Interesse ausdrücklich angeraten für die Lösung der P1-Programmieren-Aufgaben des PTP-Labors nur solche Automatismen zu nutzen, die in der Vorlesung vorgestellt bzw. "freigegeben" wurden.

- Die konkret gestellten Aufgaben dienen zur Abrundung und fordern erstes Java-Wissen ein.
- Zur Lösung dieser Aufgaben dürfen weder arrays noch Collections verwendet werden.
 - Unter https://users.informatik.haw-hamburg.de/~schafers/S17W_P1/CODE/LAB/Aufgabenzettel1_v*/ finden Sie die Code-Templates zu den einzelnen Aufgaben als ZIP-Dateien.
- Leider gehen den Laborterminalen "zum" 1.Aufgabenzettel zu wenig Vorlesungstermine voraus, so dass quasi der Vorkurs (dessen Besuch zwar angeraten, aber nicht Pflicht war) abgetestet wird. Sollten Sie echter Anfänger sein und nicht in der Lage gewesen sein am Vorkurs teilzunehmen, dann teilen Sie dies am Anfang der Abnahme mit. Der 1.Aufgabenzettel stellt den Versuch dar, den ersten Laborterminal sinnvoll zu nutzen.

Aufgaben A1.1-A1.3 Kleine Übungsaufgaben

Lösen Sie zum Einstieg die folgenden 3 kleinen Java-Übungsaufgaben **A1.1-A1.3** (Pflichtteil).

Für die Ausgabe kann Ihnen die Methode `System.out.printf()` nützliche Dienste leisten. (Sie ist an das `printf()` der Programmiersprache C angelehnt). Die Methode hat als erstes Argument in den runden Klammern einen Formatstring, der von doppelten Hochkommata begrenzt ist. Innerhalb des Formatstrings stehen Formatelemente, welche die Formatierung der Ausgabe regeln. Nach dem Formatstring kommen als weitere Argumente die auszugebenden Variablen bzw. Expressions getrennt durch Komma. Für jeden auszugebenden Wert muss im Formatstring ein Formatelement vorhanden sein.

Beispiele

<code>System.out.printf("%d", x);</code>	die ganzzahlige Variable x wird als Dezimalwert ausgegeben
<code>System.out.printf("%6d", x);</code>	wie eben – es werden 6 Stellen (rechtsbündig) reserviert
<code>System.out.printf("%-3d", x);</code>	wie eben – es werden 3 Stellen (linksbündig) reserviert
<code>System.out.printf("\n");</code>	neue Zeile

Weitere Informationen gibt es in der [API-Dokumentation](#) z.B. unter `java.util.Formatter`.

Aufgabe A1.1 Identifizieren Sie sich

Im Pub finden Sie die Zip-Datei `A1x1.zip`. Entpacken Sie die Zip-Datei, binden Sie den gestellten Code in Eclipse ein und beachten Sie dabei, dass Sie diese Aufgabe in einem eigenen Eclipse-Projekt lösen.

In der Klasse `Identifier` finden Sie eine Methode `identify()`. Ergänzen Sie diese um entsprechende Java-Zeilen, so dass Ihre `StaffelNr.` (`S1a`, `S1b`, `S2a`, `S2b`, `S3a`, `S3b`, `S4a` oder `S4b`) und Ihre Namen ausgegeben werden.

Die `StaffelNr.` bestimmt sich aus Ihrer Gruppenzugehörigkeit. Es gilt:

```
S1a ::= BTI1-PTP/01
S1b ::= BTI1-PTP/03
S2a ::= BTI1-PTP/05
S2b ::= BTI1-PTP/07
S3a ::= BTI1-PTP/02
S3b ::= BTI1-PTP/04
S4a ::= BTI1-PTP/06
S4b ::= BTI1-PTP/08
```

Ihre Namen können Sie auf Ihrem Studentenausweis finden – falls Sie diesbezüglich Unsicherheiten haben ;-)

Davon ausgehend, dass Sie in einem Zweier-Team arbeiten, soll die Ausgabe 3-zeilig erfolgen. In der ersten Zeile soll Ihre `StaffelNr.` ausgegeben werden. In der zweiten Zeile soll der erste Name und in der dritten Zeile der zweite Name ausgegeben werden. Die Ausgabe der beiden Namen soll lexikographisch sortiert erfolgen. (Damit ist auch klar, welcher Name der Erste und welcher der Zweite ist ;-)

Sollten Sie nicht in einem Zweier-team arbeiten (was eigentlich nicht sein dürfte;-), dann ist die Aufgabe geeignet zu adaptieren.

Aufgabe A1.2 Unterwegs auf dem 1€-Flohmarkt

Im Pub finden Sie die Zip-Datei A1x2.zip. Entpacken Sie die Zip-Datei, binden Sie den gestellten Code in Eclipse ein und beachten Sie dabei, dass Sie diese Aufgabe in einem eigenen Eclipse-Projekt lösen. Der gestellte Code enthält eine Lösung zu der nachfolgenden Aufgabe. Versuchen Sie die Lösung zu verstehen und seien Sie bereit die Lösung zu erklären.

Haben Sie Verbesserungsideen?

Wären Sie bereits in der Lage diese Ideen umzusetzen – also zu implementieren?

"Story im Hintergrund":

Sie sind Pleite. Sie haben also kein Geld oder konkret 0€. Allerdings haben Sie einen Keller mit unendlich viel Zeug, das Sie endlich mal loswerden wollen. Zum Glück ist Ihrer Nähe der 1€-Flohmarkt. Hier gilt die Regel, dass ALLES(!) was gekauft oder verkauft wird 1€ kostet. Es wird nur Bargeld akzeptiert.

Wer gekaufte Ware nicht bezahlen kann wird von der Flohmarktaufsicht lebenslänglich an den Schandpranger gestellt (und Sie wollen auf gar keinen Fall, dass Ihnen so etwas passiert).

In dieser Aufgabe geht es nur um das Geld.

Im Template A1x2 finden Sie eine Klasse `Wallet` (Portemonnaie) mit Methoden `payForIceCream()`, `payForCoke()` und `receiveMoneyForSoldItem()`.

Erzeugen Sie ein Objekt der Klasse `Wallet`. Dieses Objekt repräsentiert Ihr Portemonnaie. Sie mögen Eis, das Sie natürlich bezahlen müssen `payForIceCream()`. Ebenso könnten Sie Durst bekommen und ein Coke trinken, die Sie natürlich bezahlen müssen `payForCoke()`. Wenn Sie etwas verkaufen bekommen Sie dafür Geld `receiveMoneyForSoldItem()`.

Aufgabe A1.3 Die letzten Drei

Im Pub finden Sie die Zip-Datei A1x3.zip. Entpacken Sie die Zip-Datei, binden Sie den gestellten Code in Eclipse ein und beachten Sie dabei, dass Sie diese Aufgabe in einem eigenen Eclipse-Projekt lösen.

In der Klasse `LastThree` finden Sie eine Methode `processNewValue()` vor. Beim Aufruf dieser Methode wird eine Zahl "entgegengenommen" (konkret vom Typ `int`) und im formalen Parameter `value` abgelegt. Ihre Aufgabe ist es, sich die letzten 3 Werte zu merken. Beim Aufruf der Methode `printLastThree()` sollen die letzten 3 Werte ausgegeben werden.

Insbesondere zur Lösung dieser Aufgabe dürfen weder Arrays noch Collections verwendet werden. (Sofern Sie noch nicht wissen, was ein Array oder eine Collection ist, werden Sie auch nicht Gefahr laufen diese zu verwenden ;-)