Programmieren 2
Selbststudium Semesterwoche 9

Aufgaben zum Studienelement OOP
1. Beantworten Sie noch einmal die Kontrollfragen A bis D auf den Folien 9, 20, 32 und 43 des Inputs zu OOP.
A.1. Was ist ein Prozess?
→ Ein Prozess (process, task) ist eine dynamische Folge von Aktionen (Zustandsänderungen), die durch Ausführung eines Programms auf einem Prozessor zustande kommt.
Das Betriebssystem verwaltet für jeden Prozess einen eigenen Kontext, der alle relevanten Informationen über den Prozess enthält. Wegen ihres umfangreichen Kontextes sind Prozesse oft "schwergewichtig". Dadurch ist der Aufwand beim Prozesswechsel erheblich (Änderung der Prozessorzuteilung).
→ In der Regel wir ein Programm als einziger Prozess ausgeführt.

A.2. Was ist ein Thread?
→ Ein Thread ist ein Aktivitätsträger (sequentieller Ausführungsfaden) mit minimalem Kontext (Stack und Register) innerhalb einer Ausführungsumgebung (Prozess). Alle Threads, die zu ein und demselben Prozess gehören, benutzen denselben Adressraum sowie weitere Ressourcen (Betriebsmittel) dieses Prozesses gemeinsam.
→ Ein Prozess kann ein oder mehrere Threads haben.
→ Häufiges Erkennungsmerkmal bei Single-threaded Applikationen ist das "einfriehren des GUI's". (Problem: GUI läuft im selben Thread wie der Rest der Anwendung).

A.3. Was bedeutet Nebenläufigkeit?
→ Mehrere Vorgänge heissen nebenläufig, wenn sie voneinander unabhängig bearbeitet werden können.
→ Mit der Nebenläufigkeit von Prozessen findet zugleich ein Wettstreit um notwendige Ressourcen (Betriebsmittel) statt.
→ Zwei nebenläufige Prozesse sind also nur bezüglich ihrer Abarbeitung unabhängig; sie sind jedoch indirekt voneinander abhängig, wenn sie auf gleiche Ressourcen zugreifen wollen.
→ Bemerkung: Analoges gilt für Threads.

B.1. Nennen Sie fünf Methoden der Klasse Thread.
→ public void run();
→ public void start();
→ public static void sleep(int millis) throws InterruptedException;
→ public final int getPriority();
→ public final void setPriority(int newPrio);

B.2. Welches Package müssen Sie importieren, wenn Sie die Klasse Thread verwenden wollen?
→ java.lang

C.1. Welche Methode besitzt das Interface Runnable?
→ void run();

C.2. Beschreiben Sie kurz zwei Varianten, Threads basierend auf dem Interface Runnable zur Verfügung zu stellen.
→ Variante 1:
- Die aufrufende Klasse erzeugt eine Instanz der Klasse CountDown2, die das Interface Runnable implementiert.
- Anschliessend erzeugt sie den Thread. Dem Konstruktor des Threads wird dann das erzeugte Objekt der Klasse XY übergeben.

→ Variante 2:
- Die Klasse stellt ebenfalls eine Methode start() zur Verfügung. Zusätzlich verwaltet sie selber eine Instanz der Klasse Thread.

- Dadurch kann die Klasse in der aufrufenden Klasse genauso verwendet werden wie die Klasse CountDown1 (extends Thread) in der Klasse ThreadEx1.

C.3. Sie entscheiden sich für die Variante einer Klasse, mit einem Attribut vom Typ Thread. Was müssen Sie tun, um andere Methoden der Klasse Threads nach aussen zur Verfügung zu stellen?
→ Eine Methode `public void start()` welche das lokale Thread-Objekt initialisiert und startet.

```
public void start()
{
	if (thread == null)
	{
		thread = new Thread(this);
		thread.start();
	}
}
```

D.1. Beschreiben Sie kurz in Ihren eigenen Worten, was synchronized macht.
→ Der Begriff Wechselseitiger Ausschluss bzw. Mutex (Abk. für engl. mutual exclusion, auf deutsch etwa wechselseitiger Ausschluss) bezeichnet eine Gruppe von Verfahren, mit denen das Problem des kritischen Abschnitts gelöst wird.
→ Mutex-Verfahren verhindern, dass nebenläufige Prozesse bzw. Threads gleichzeitig oder zeitlich verschränkt gemeinsam genutzte Datenstrukturen unkoordiniert verändern, wodurch die Datenstrukturen in einen inkonsistenten Zustand geraten können, auch wenn die Aktionen jedes einzelnen Prozesses/Threads für sich betrachtet konsistenzerhaltend sind.
→ Mutex-Verfahren koordinieren den zeitlichen Ablauf nebenläufiger Prozesse/Threads derart, dass andere Prozesse/Threads von der Ausführung kritischer Abschnitte ausgeschlossen sind, wenn sich bereits ein Prozess/Thread im kritischen Abschnitt befindet (die Datenstruktur verändert).

http://java.sun.com/docs/books/tutorial/essential/concurrency/syncmeth.html
http://de.wikipedia.org/wiki/Mutex

D.2. Nennen Sie ein Beispiel, wann Sie synchronized benötigen.
→ Überall dort, wo sichergestellt werden muss, dass eine bestimmte Programmsequenz keinesfalls durch den Ressource Scheduler des Betriebssystems unterbrochen werden darf.
→ Zum Beispiel: Ein gemeinsamer Counter für mehrere (quasi-) parallele Threads

2. Bearbeiten Sie das Tutorial "VJT: Visual Java Threads". Sie finden das Tutorial unter ILIAS wie folgt: Magazin > Hochschule Luzern > Frei verfügbare Lerninhalte > Informatik > Multithreading oder direkt: https://elearning.hslu.ch/ilias/data/hslu/lm_data/lm_441799/Visual%20Java%20Threads_tinylms/index.html Beachten Sie für den Abschnitt "Threads: Zustände" die "Anleitung zum VJT Tutorial", die Sie ebenfalls im ILIAS finden. Beantworten Sie die Kontrollfragen, die Sie im Tutorial finden.

Aufgaben Studienelement SWE
3. Wer sind im PRGII-Projekt Ihrer Ansicht nach die relevanten Stakeholder?
→ Der Auftraggeber (Dozententeam)
→ Der Benutzer (in diesem Fall anonym)

4. Nennen Sie mindestens je 2 Informationsquellen und Techniken zur Informationsbeschaffung für Requirements für das PRGII-Projekt.
→ Informationsquellen
- Stakeholder
- Rollenträger: Endbenutzer, Auftraggeberin, Betreiber, Entwicklerin, Projektleitung,

→ Techniken
- Interviews: Beteiligte werden einzeln oder in Gruppen befragt
- Umfragen/Fragebogen: Erfassung von Begriffswelt und Bedürfnissen einer grösseren Gruppe von Beteiligten
- Beobachtung: Beteiligte bei der Arbeit beobachten
- Workshops: Erarbeitung von Anforderungen in einer Gruppe ausgewählter Beteiligter
- Prototyping: Gewinnen von Anforderungen durch Erprobung möglicher Lösungen

5. Lesen Sie auf der HTAgil Website (http://edu.enterpriselab.ch/htagil) im Kapitel HTAgil Dokumente die beiden Unterkapitel über die Kundenanforderungen und die Systemspezifikation. Laden Sie die Word-Vorlagen dieser zwei Dokumente herunter.

6. Notieren Sie in Stichworten
was Sie für das PRGII-Projekt im Dokument Kundenanforderung festhalten wollen und

- Produktübersicht
- Produktfunktionen (wichtige Arbeitsabläufe, Akteure, Geschäftsprozesse, Funktionalitäten des Produktes)
- Produktdaten (wichtige Daten und Mengengerüste)
- Qualitätsanforderungen, Leistungsanforderungen

welche Aspekte Sie in der Systemspezifikation näher spezifizieren wollen.

- Architektur-Modelle:eingesetzte Architektur-Prinzipien und Stile, eingesetzte Modell-Prüfungen, Kommentare, Bezüge zu den Software-Anforderungen
- Spezifikation externer Schnittstellen
- Softwareanforderungen:funktionale und nicht-funktionale Anforderungen
- Environment-Anforderungen:z.B. HW, OS, VM

7. Formulieren Sie auf oberster Ebene für das PRGII-Projekt ein WAS und wenigstens zwei sinnvolle mögliche WIE.
→ WAS: Die Spielsteine sollen in der Anzahl und Farbe variieren können.
→ WIE: Spielsteine als Bilder oder Vektorgrafiken anzeigen
→ WIE: Spielsteine und Anzahl in Form von programmierbaren Schwierigkeitsprofilen vorgeben.