Gedächtnisprotokoll

Programmieren II für Wirtschaftsinformatiker (Prog2)

23.07.2015

Dieses Gedächtnisprotokoll ist selbstverständlich nicht in irgendeiner Weise verbindlich/offiziell/..., aber bietet vielleicht einen kleinen Eindruck hinsichtlich der Prüfung. Und selbstverständlich: Keine Garantie auf Vollständigkeit oder Korrektheit. (Version 1.0)

Aufbau

- 76,5 Punkte maximal zu erreichen
- 150min Bearbeitungszeit, 10 Minuten Einlesezeit, 3-5 Minuten, um Namen auf Blätter zu schreiben
- insgesamt ~25 einseitig bedruckte Seiten mit 5 großen Aufgaben
  - jeweils unterteilt in ca. 1-5 Teilaufgaben
- ~7 Seiten alphabetisch geordnete API
  - durchaus sehr hilfreich, um alle Aufgaben zu lösen
  - Angabe, welche `Exceptions` geworfen werden (könnten)
- Zeitlich machbar, aber sehr anstrengend
- nicht allzu viele neue Klassen
- verständliche Aufgabenstellungen, aber teilweise umfangreiche vorgegebene Methoden und Klassen
- kleinere Stolperfallen
  - hier mal ein fehlendes `extends JFrame` oder dort mal eine Klammer
  - oder auch ein fehlendes `throws XYException`

1 GUI-Programmierung

- gegeben: Abbildung eines Programmfensters
- Aufgabe: dieses ↑ „nachbauen“
- geeignet: `GridLayout(3, 3)`
  - eine Spalte für `JButtons`, eine für `JLabels`, eine für `JTextFields`
- alle Komponenten initialisieren, einer jeweiligen `LinkedList<JButton>`/... hinzufügen und anschließend in das Layout einpflegen
- zwei `ActionListeners` konstruieren
  - `mouseEntered`: wenn in einem bestimmten der Textfelder „RED“ steht, dann die anderen beiden rot einfärben
  - `mouseExited`, wenn in einem der Textfelder „GREEN“ steht, dann die jeweils anderen grün einfärben
  - `MouseListener`/`MouseAdapter-Klasse` bietet sich an
    - entsprechend auch in der API gegeben
- wichtig (für mehrere Aufgaben): `String.equals(String xy)` statt `==`-Vergleich

2 GUI/Multiple Choice

- mehrere Screenshots von GUIs gegeben
- immer verschiedene `JButtons` oder `JLabels` in spezieller Anordnung
- jeweils 6 verschiedene Antwortmöglichkeiten, mit welcher Kombination von Layoutmanagern man diese GUI zeichnen könnte
  - pro Screenshot nur eine Kombination richtig
  - Beispiel:
    - a) `FlowLayout, GridLayout, BoxLayout`
    - b) `GridLayout, BorderLayout`

3 Exceptions

- eine ca. 15-zeilige Methode ist gegeben
- `try-multi-catch`-Block gegeben, aber jeweils ohne Benennung der entsprechenden Ausnahme
- durch scharfes Hinsehen und Nutzung der APi: Ausfüllen, welche `Exceptions` möglich sind
  - durch ein `System.out.println(String beschreibung)` deutlich machen, was die `Exception` für eine Ursache hat
- Frage: Welcher `clause` wird bei mehreren möglichen aufgerufen?

4 XML

- DTD, Ordnerstruktur, `Documentbuilder` (Parsen) bereits gegeben
- Szenario: „XML-Dokument (nicht gegeben) gibt früheren Büchereibestand an. Außerdem: Zwei `LinkedLists`, die aktuelle Bücher/Paper, die verfügbar sind und außerdem ausgeliehene Medien speichern."
- Aufgabe: Innerhalb einer Methode das `Document doc` durchsuchen und mit Inhalten der `LinkedList` vergleichen; außerdem andere Inhalte zählen und `int` (als Differenz zu gegebenem Wert) zurückgeben
- kein Transformerschreiben oder XML-Struktur anlegen notwendig; keine manuelle Konstruktion von XML zu DTD oder vice versa
- verwendet wurde der `DOM-Parser`, kein `SAX`

5 Threads

- drei „Klassenskelette" gegeben
  - „Prof", „WiMi" und „Corrector"
- Aufgabe: „Vervollständigung der Klassen, um ein Producer/Consumer-Problem zu lösen (Klausurkorrektur)."
- Konstruktoren schreiben
- Threads erstellen (`extends Thread` oder `implements Runnable`) und diese starten
- `run()`-Methoden implementieren
- Stichworte: `wait(), interrupt(), sleep(long time)`...
- außerdem eventuell Spielereien wichtig: `modulo`-Operator, `for-each`-Schleifen (`Collections`), `while(true)`...

6 Netzwerkprogrammierung

- Server und Client in UDP (kein TCP) implementieren
- Szenario: „Ich und Freund wollen Nachrichten versenden. Freund sendet langen String mit mehreren durch Komma getrennte Informationen."
- `String.split()`, `String[]`,...
- `public` Variable erstellen (Sichtbarkeit)
- Zahl aus Nachricht parsen
- erstes `DatagramPacket` anders behandeln als darauffolgende
  - z.B. mit einer `boolean`-flag oder einem `int`-Zähler
- spezielle Klasse `ByteBuffer` verwenden können
  - vorher unbekannt, war aber in der API gut beschrieben
- Hinweise zur Vorgehensweise beschrieben
- GUI-Komponenten updaten mit erhaltenen Daten

7 Theorie

Einige Theoriefragen waren über alle Aufgaben hinweg verstreut, deshalb hier eine Sammlung.

- JUnit tests
  - `assertTrue`, `assertFalse` und `assertEquals` auf unkomplizierte Methode anwenden (Zahl $>$ 100: `false`; 100 $>$ Zahl $>$ 0: `true`; Zahl $<$ 0: `new XYException`)
- Swing und Thread-Sicherheit gegeben?
  - `SwingUntilities.involkeLater(){...}`
- Welche Komponenten werden bei Swing automatisch erstellt?
  - (RootPane, GlassPane etc.)
- kurze Erläuterung und Beispiel zu TCP, UDP, SMTP und HTTPS geben
  - Nachteile UDP?
- Welche Arten gibt es Threads zu erzeugen (`extends Thread` versus `implements Runnable`)?
- Model-View-Conrol-Architekturpattern erklären
- Was ist der Unterschied zwischen `isInterrupted(), interrupt(), interrupted()`...?