Programmierung I – 1. Klausur WS 18/19 – bei Herr Prof. Dr. Müller

100 von 110 Punkte für 1.0 erreichen

Aufgabe 1 (7P)

7 Aufgaben: Programmschnipsel: Frage nach Ergebnis, Datentyp,   
Erklärung in eine Tabelle auf Klausurblatt schreiben

Bsp: int m = 1; double n = (int) 1.0;

Aufgabe 2 (6P)

6 Aufgaben: Frage zu Programmschnipseln nach Ergebnis oder wenn es nicht läuft den Grund

Aufgabe 3

1. Programm auf Blattpapier schriftlich erklären – (6P)
2. Programm ohne for und ohne switch programmieren (8P)

Public static int xx (long feld[], int/long? parameter){

Int value = 0;

Switch(paramerter)

Case 0: for (long k : feld) value += v; return value;

Case 1: for(long k : feld) value += v\*v; return value;

}

Aufgabe 4 (8P)

Methode schreiben die einen String übergeben bekommt;  
dieser String s soll untersucht werden, ob er aus zwei gleichen Teilstrings t besteht;   
wenn ja soll der Teilstring t zurückgegeben werden – sonst null;

Bspw: String s = abab hat t = ab;

Aufgabe 5 (8P)

Rekursion: mit if und else if und else die Anweisungen auf dem Blatt entsprechend implementieren

Aufgabe 6 (12P) - Arrays

Es wurde ein Array einer Methode übergeben;

Man sollte schauen, ob irgendeine Zahl im Array genau 2x vorkommt,   
wenn ja diese ausgeben,   
wenn nicht false;

Aufgabe 7 (15P) – Collections

Es gab: eine Klasse Professoren und eine Klasse Studenten und eine main-Klasse

Klasse Professor:   
- String domain (Fachbereich)   
-int officeHours  
- String name  
LinkedList<studenten> list;

Klasse studenten:  
- String domain (Fachbereich)   
-int time (benötigte Zeit für Beratung)  
- String name

Man sollte nun zwei Methoden schreiben:

1. Methode bekommt ein student übergeben  
   ist dieser vom selben Fachbereich (domain) wie der Prof;   
   so soll er in die Liste vom Prof eingetragen werden  
   sonst sollte eine Ausgabe auf Konsole erfolgen (Prof x ist nicht zuständig für Student y)
2. Methode soll die Liste vom Prof abarbeiten  
   es sollen die studentnamen ausgegeben werden  
   jeder student benötigt seine „time“   
   diese wird von „officeHours „ bei jeder Ausgabe vom studenten um studenten.getTime() verringert

Ist officeHours auf 0 so sollen alle restlichen Studenten in der Liste ausgegeben werden;  
mit den Zusatz (in etwa) „ Prof x konnte Student y nicht bearbeiten“  
(alle Ausgaben war nicht auf deutsch sondern auf englisch)

Aufgabe 8 (15P) Exceptiones

Es wurde ein String an eine Methode übergeben, die geschrieben werden sollte.

Dieser String beinhaltet das Schema „ISBN; titelvonBuch; abstract“   
bpsw: „123;rotkappe; abstract“

Es konnte passieren, dass der String null ist, nicht alle Angaben enthalten sind;

Oder „;“ 1x fehlt

Der String sollte auf drei String aufgeteilt werden;

Die ISBN sollte in einen int Variable gespeichert werden;

Falls isbn zu lang war oder keine Zahl war sollte eine Exception fliegen und den Wert auf -1 von isbn setzen;

Ansonsten sollte eine Ausgabe wie folgt erfolgen; wichtig war hier die Bündigkeit(tabellenartig)  
ISBN: 125

Titel: TitelBuch

Abstract: Nicht abstract!

Aufgabe 9 (15P) Vererbung

* Abstracte class camera schreiben:
  + String model
  + Int id (soll bei 0 starten ; inklusive )
  + Int pictures( Anazhl der getätigten Bilder, soll bei 0 starten, inklusive)
  + Boolean hasObjektiv (ob man Objektiv abschrauben kann)
  + Abstracte methode void takePicture();
  + toString methode
* von camera abgeleitet
  + class Smartphone
    - sollte im Konsturktor name übergeben bekommen; und im Konstruktor (super(name, false) wegen kein Objektiv abnehmbar)
  + class SRL-Kamera
    - sollte im Konstruktor (model, String Objektivname) erhalten;
    - im Konstruktor true wegen Objektiv
  + Methode takePicture in Smartphone und SRL-Kamera implementieren;  
    wenn aufgerufen soll sich die Zahl picture erhöhen und ausgeben werden plus mit welcher Kamera ( id und model) und wenn vorhanden Objektiv

Aufgabe 10 (15P)

Abgewandelte Version von Queue mit Elem; 2 Methoden schreiben;   
es gab keine getter/setter-Methoden bei Elem; es gab nur String value und Elem next;  
methode enQueue, toString(?) waren vorgegeben;

1. Methode: Insert ( int index, String word) – soll an stelle index word einfügen
2. Methode: bekommt einen String übergeben; der aus mehreren Teilen besteht;  
   diese Teile sind mittels Leerzeichen voneinander getrennt;  
   diese Teile sollen nun in die bestehe Queue hinten angehängt werden;   
   Jedes Teil ein neues Elem;