University of Zurich[UZH]

s.e.a.l. software evolution & architecture lab

Informatik I – EProg HS12

Zwischentest I

Allgemeine Hinweise:

- Jede Aufgabe muss vollständig bearbeitet werden, damit Sie die volle Punktzahl erhalten. Insgesamt können **60 Punkte** erreicht werden.
- Die Bearbeitungszeit beträgt **60 Minuten**.
- Unterlagen sind grundsätzlich **keine** erlaubt. Als Ausnahme gilt, falls Sie nicht deutscher Muttersprache sind, ein entsprechendes Wörterbuch.
- Schreiben Sie Ihren **Namen** und Ihre **Matrikelnummer** bitte in die dafür vorgesehenen Kästchen am Ende **jeder** Seite.
- Die Verwendung unerlaubter Hilfsmittel oder das Abschreiben von eine(r/m) Komiliton(in / en) hat die sofortige Abgabe und das Nichtbestehen der Veranstaltung zur Folge. Zudem ist mit einem Disziplinarverfahren zu rechnen.
- Eine englische Version dieses Zwischentestes ist auf Nachfrage verfügbar. Bei Differenzen zwischen der deutschen und der englischen Version dieses Zwischentestes ist die deutsche Version massgebend.

Ich bestätige mit meiner Unterschrift, dass ich:

- die obenstehenden Hinweise gelesen und verstanden habe.
- die Prüfung unter zumutbaren Bedingungen geschrieben habe.

Unterschrift: _____ Datum: _____

Punkte
/60

1 Aufgabe: Loops (5 Punkte)

Kreuzen Sie die richtigen Aussagen an. Beachten Sie, dass Ihnen für jedes falsch gesetzte Kreuz gleich viele Punkte abgezogen werden, wie Sie für ein korrekt gesetztes Kreuz erhalten. Negative Punktzahlen ergeben null Punkte für die betreffende Frage.

1.1 (3 Punkte)

```
1  private int someMethod(int num1, int num2) {
2     while (num2 != 0 && num1 != 0) {
3        if (num1 > num2) {
4           num1 = num1 - num2;
5        } else {
6           num2 = num2 - num1;
7        }
8     }
9     return num1;
10 }
```

Was ist das Resultat der Methode `someMethod()`, wenn die Methode mit den Parametern `num1=6` und `num2=12` aufgerufen wird.

- ☐ 1
- ☐ 6
- ☐ 3
- ☐ 7

Lösung:
☒ 6

1.2 (2 Punkte)

```
1  private int someMethodModified(int num1, int num2) {
2     for(int i=0;i<1;i++){
3        if(num2 != 0 && num1 != 0){
4           if (num1 > num2) {
5              num1 = num1 - num2;
6           } else {
7              num2 = num2 - num1;
8           }
9           i--;
10       }
11    }
12    return num1;
13 }
```

Welche Aussage ist korrekt?

- ☐ Die Methode `someMethodModified` liefert bei denselben Parametern die gleichen Ergebnisse wie die Methode `someMethod`.
- ☐ Die Methode `someMethodModified` liefert bei denselben Parametern das doppelte Ergebnis wie die Methode `someMethod`.
- ☐ Die Methode `someMethodModified` liefert kein Ergebnis, da eine Endlosschleife programmiert wurde.
- ☐ Die Methode `someMethodModified` hat Syntaxfehler und könnte nicht kompiliert werden.

Lösung:
☒ 1

2 Aufgabe: Typumwandlung (4 Punkte)

Kreuzen Sie die richtigen Aussagen an. Zu jeder Frage können mehrere Antworten richtig sein. Beachten Sie, dass Ihnen für jedes falsch gesetzte Kreuz gleich viele Punkte abgezogen werden, wie Sie für ein korrekt gesetztes Kreuz erhalten. Negative Punktzahlen ergeben null Punkte für die betreffende Frage.

```
1 byte myByte = (byte) (1024 + 512);
```

Welche Aussagen sind korrekt?

- ☐ Da die Summanden ganzzahlig sind werden sie von Java als `long`-Werte interpretiert. Deshalb ist die Summe zunächst auch einmal ein `long`-Wert. Bei der Zuweisung auf die `myByte`-Variable wird dann allerdings eine explizite Typumwandlung von `long` nach `byte` vorgenommen.
- ☐ Da die Summanden ganzzahlig sind werden sie von Java als `int`-Werte interpretiert. Deshalb ist die Summe zunächst auch einmal ein `int`-Wert. Bei der Zuweisung auf die `myByte`-Variable wird dann allerdings eine explizite Typumwandlung von `int` nach `byte` vorgenommen.
- ☐ Da die Summanden ganzzahlig sind werden sie von Java als `int`-Werte interpretiert. Deshalb ist die Summe zunächst auch einmal ein `int`-Wert. Bei der Zuweisung auf die `myByte`-Variable wird dann allerdings eine implizite Typumwandlung von `int` nach `byte` vorgenommen.
- ☐ Bei dieser Umwandlung schneidet Java 24 Binärstellen ab, damit das Ergebnis in ein `byte` passt.
- ☐ Bei dieser Umwandlung schneidet Java 56 Binärstellen ab, damit das Ergebnis in ein `byte` passt.
- ☐ Bei dieser Umwandlung schneidet Java keine Binärstellen ab.

Lösung:
2 und 4

3 Aufgabe: Algorithmus (7 Punkte)

Schreiben Sie eine Funktion `sumOdd(int n)`, welche alle ungeraden Zahlen von 0 bis n addiert und die Summe als Integer-Wert zurückliefert.

Ihre Lösung:

Lösung:

```java
public int sumUpOdd1(int n) {
   int sum = 0;
   while(n > 0) {
      if ((n%2) == 1) {
         sum += n;
      }
      n--;
   }
   return sum;
}

public int sumUpOdd2(int n) {
   int sum = 0;
      if (n <= 0){
         return 0;
      }
   for (int i=0; i<=n; i++){
      if ((i%2)==1){
         sum += i;
      }
   }
   return sum;
}

public int sumUpOdd3(int n) {
   if (n == 0) {
      return n;
   } else {
      if (n%2 == 1) {
         return n + sumUpOdd3(n - 1);
      } else {
         return sumUpOdd3(n - 1);
      }
   }
}
```

Listing 1: `Algorithmus.java`

4 Aufgabe: Implementierung (44 Punkte)

4.1 Prüfungsverwaltung (44 Punkte)

Eine Lehrerin verwaltet die Prüfungen (`Exam`) ihrer Schüler auf ihrem PC. Dabei hat jede Prüfung einen Namen des Schülers (`name`) der die Prüfung abgelegt hat, sowie die erreichte Punktzahl (`points`). Die einzelnen Prüfungen werden nach Schulklassen (`SchoolClass`) verwaltet. Jede Schulklasse hat einen Namen (`name`) und besteht aus maximal 15 Schülern. Deshalb werden in einer Schulklasse auch maximal 15 Prüfungen verwaltet. Die Lehrerin hat die Möglichkeit die höchste Punktzahl auszugeben, die in einer Schulklasse erreicht wurden. Zusätzlich wird auch der Namen des Schülers, welcher die höchste Punktzahl erreicht hat, ausgegeben.

Bilden Sie den oben stehenden Sachverhalt auf Klassen, Methoden, Attribute und Objekte ab. Befolgen Sie hierzu unbedingt die nachfolgenden Implementierungshinweise:

- ✓ Die Werte von Instanzvariabeln sollen nur über Methoden verändert werden können.
- ✓ Wenn eine Prüfung zur Schulklasse hinzugefügt wird, soll zuerst überprüft werden, ob die maximale Anzahl an speicherbaren Prüfungen bereits erreicht ist. Nur wenn dies nicht der Fall ist, soll die Prüfung hinzugefügt werden. Andernfalls soll auf der Kommandozeile folgender Satz ausgegeben werden: `The class is already full!`.
- ✓ Soll die höchste Punktzahl ausgegeben werden, so wird zuerst überprüft, ob Prüfungen gespeichert sind. Ist dies nicht der Fall, wird `There are no exams registered!` auf der Kommandozeile ausgegeben. Andernfalls soll die höchste erreichte Punktzahl, sowie der Name des Schülers, der diese Punktzahl erreicht hat, auf der Kommandozeile ausgegeben werden.
- ✓ Sie können davon ausgehen, dass nur ein Schüler die höchste Punktzahl erreicht hat.

Testen Sie in einem TestDriver die folgenden Vorgänge:

1. Erstellen Sie eine Schulklasse mit dem Namen `6c`.
2. Erstellen Sie eine Prüfung mit dem Schülernamen `Carl` mit 4 Punkten, sowie eine mit dem Schülernamen `Anne` mit 5 Punkten.
3. Fügen Sie die Prüfungen zur Schulklasse hinzu.
4. Lassen Sie die höchste erreichte Punktzahl ausgeben.

Es sind verschiedene Lösungswege denkbar!

Ihre Lösung:

Lösung:

```java
public class SchoolClass {
   private int numOfStudents = 0;
   private String name;
   private Exam[] exams = new Exam[15];

   public void addExam(Exam exam){
      if(numOfStudents < 15){
         exams[numOfStudents++] = exam;
      }else{
         System.out.println("The class is already full!");
      }
   }

   public void setName(String name){
      this.name = name;
   }

   public String getName(){
      return name;
   }

   public void printBestExam(){
      if (numOfStudents <= 0){
         System.out.println("There are no exams registered!");
      }else{
         Exam bestExam = exams[0];
         for (int i = 1; i < numOfStudents; i++) {
            if (exams[i].getPoints() > bestExam.getPoints()) {
               bestExam = exams[i];
            }
         }
         System.out.println(bestExam.getStudentName() + " reached the
             best result with " + bestExam.getPoints()+ " points.");
      }
   }
}
```

Listing 2: `SchoolClass.java`

```java

public class Exam {
   private String studentName = "";
   private int points;

   public void setStudentName(String name){
      this.studentName = name;
   }

   public void setPoints (int points){
      this.points = points;
   }

   public String getStudentName(){
      return studentName;
   }

   public int getPoints(){
      return points;
   }

}
```

Listing 3: `Exam.java`

```java

public class Tester {
   public static void main(String[] args) {
      Exam exam1 = new Exam();
      exam1.setStudentName("Carl");
      exam1.setPoints(4);

      Exam exam2 = new Exam();
      exam2.setStudentName("Anne");
      exam2.setPoints(5);

      SchoolClass testClass = new SchoolClass();
      testClass.setName("6c");
      testClass.addExam(exam1);
      testClass.addExam(exam2);

      testClass.printBestExam();
   }

}
```

Listing 4: `TestDriver.java`

Hier steht zusätzlicher Raum für Ihre Lösung für Aufgabe 4.1 zur Verfügung.