

Basisprüfung Frühjahr 2018

0027 – Einführung in die Programmierung

Departement Informatik
ETH Zürich

9. August 2018 – Schriftlicher Teil

Nachname: _____ Vorname: _____ Stud.number: _____

Mit Ihrer Unterschrift bestätigen Sie, dass Sie folgende Hinweise zur Kenntnis genommen haben, die Aufgaben selbständig gelöst haben, Sie Ihre eigene Lösung abgeben, Sie keine Kopie der Prüfung mitnehmen, es keine störenden äusseren Einflüsse gab, und Sie keine gesundheitlichen Probleme hatten, die Ihre Leistungen in dieser Prüfung beeinträchtigten.

Signature: _____

Aufgabe	Wert	Punkte	Aufgabe	Wert	Punkte
1	6		5	12	
2	8		6	8	
3	12		7	6	
4	8		Prog.		
Zsumme			Total	60	

Allgemeine Informationen

1. Öffnen Sie diese Prüfung erst, wenn die Aufsicht den Beginn der Prüfung bekannt gibt.
2. Schreiben Sie zuerst Ihren Namen auf diese Prüfung, damit Sie es später nicht vergessen.
3. Die Prüfung dauert 1 Stunde (60 Minuten). Falls Sie sich durch irgendjemanden oder irgendetwas gestört fühlen, melden Sie dies sofort der Aufsichtsperson.
4. Die Prüfung hat 10 Seiten. Vergewissern Sie sich dass Ihr Exemplar vollständig ist.
5. In dieser Prüfung gibt es 60 Punkte. Benutzen Sie die Anzahl der Punkte als *Hinweis*, wie Sie Ihre Zeit einteilen können.
6. Lesen Sie die Aufgabenstellungen genau durch.
7. Tragen Sie Ihre Antwort(en) direkt in die Prüfungsbögen ein. Falls Sie mehr Platz brauchen, ist Ihre Antwort wahrscheinlich zu lang. Wenn Sie doch mehr Platz brauchen, so benutzen Sie die Rückseiten.
8. *Benutzen Sie einen Kugelschreiber (blau oder schwarz) oder Füller der nicht ausradiert werden kann. Benutzen Sie keinen Bleistift. Bitte schreiben Sie deutlich und leserlich! Wenn Sie etwas durchstreichen wollen, so machen Sie dies bitte klar und deutlich.*
9. Es ist wichtig, dass Ihre Antworten die Aufgaben klar und *unzweideutig* behandeln. Die Klarheit der Antworten beeinflusst Ihre Note.
Wenn Sie Annahmen (über die in den Aufgaben aufgeführten hinaus) treffen, so geben Sie diese bitte an.
10. Wir sammeln die Prüfung zum Schluss ein. Wichtig: stellen Sie unbedingt selbst sicher, dass Ihre Prüfung von einer Aufsichtsperson eingezogen wird. Stecken Sie keine Prüfung ein und lassen Sie Ihre Prüfung nicht einfach am Platz liegen. Dasselbe gilt, wenn Sie früher abgeben wollen: bitte melden Sie sich lautlos, und wir holen die Prüfung ab. Vorzeitige Abgaben sind nur bis 30 Minuten vor Prüfungsende möglich.
11. Wenn Sie zur Toilette müssen, melden Sie dies einer Aufsichtsperson durch Handzeichen. Es darf zur gleichen Zeit immer nur eine Studentin oder ein Student zur Toilette.
12. Wir beantworten keine inhaltlichen Fragen während der Prüfung. Kommentare zur Aufgabe schreiben Sie bitte auf das Aufgabenblatt.
13. Wenn die Aufsicht die Prüfung beendet schliessen Sie bitte die Prüfung und schreiben nicht mehr in die Prüfung. Weiterarbeiten ueber die erlaubte Zeit gilt als Täuschungsversuch.

Aufgabe 1 (6 Punkte)

Gegeben sei eine Methode `main` in einer Java Klasse.

```
public static void main(String[] args) {  
    /* body */  
}
```

Die folgenden Anweisungen sollen als "Body" (Rumpf) anstelle des Kommentars `/* body */` eingefügt werden. Geben Sie für jede Anweisung an, was für eine Ausgabe erzeugt wird. Achten Sie auf die korrekte Formatierung der verschiedenen Typen, also z.B. 7.0 statt 7 für eine reelle Zahl (double).

1. `System.out.println(-(8 + 1 / 3 + 5 * 4 + 8 * -2));`

2. `System.out.println(1 + "x" + 9 / 10 + " is " + 10 / 5);`

3. `System.out.println(12 % 9 * 12 % 9 * 12 % 9);`

4. `System.out.println(2 - 4 / 8 * 4.0 + 12 / 5);`

5. `System.out.println(0 + "1" + 3 + 5 + "7" + 7 * 5 + "3" + (1 + 0));`

6. `System.out.println(0 > 7 || 12 < 10 + 4);`

Aufgabe 2 (8 Punkte)

Gegeben sei eine Java Methode:

```
public static void method(int[] a) {  
    for (int i = 0; i < a.length - 2; i++) {  
        a[i+1] = (a[i] + a[i + 2]) / 2;  
    }  
}
```

Geben Sie auf der rechten Seite die Werte an, die im Array gespeichert sind, nachdem diese Methode `method` ausgeführt wurde (mit einem Array als Parameter, der wie auf der linken Seite angegeben initialisiert wurde).

Ursprünglicher Array	Endgültiger Array
<pre>int[] a1 = { 1, 1, 3 }; method(a1);</pre>	<hr/>
<pre>int[] a2 = { 2, 1, 2, 4 }; method(a2);</pre>	<hr/>
<pre>int[] a3 = { -1, 6, 3, 5, -3 }; method(a3);</pre>	<hr/>
<pre>int[] a4 = { 7, 2, 3, 1, -3, 12 }; method(a4);</pre>	<hr/>

Aufgabe 3 (12 Punkte)

Geben Sie (für die angegebenen Stellen im Programm) an ob die drei Aussagen `next == 0` und `a > 0` und `b >= next` IMMER, MANCHMAL oder NIE wahr sind. Sie können diese drei Möglichkeiten mit I/M/N abkürzen.

```
public static int assertionRaetsely(Scanner console) {
    int a = -1;
    int b = 0;
    int next = console.nextInt();

    // Point A
    while(next != 0) {

        // Point B
        b = Math.max(a, next);
        a = next;

        // Point C
        if (a == 42) {

            // Point D

            a++;
        }
        next = console.nextInt();
    }
    return b;
}
```

	<code>next == 0</code>	<code>a > 0</code>	<code>b >= next</code>
Point A			
Point B			
Point C			
Point D			

Aufgabe 4 (8 Punkte)

Gegeben sei eine Methode `sonderbar`.

```
public static void sonderbar(int i, int j) {  
    int k = 0;  
    while (i < j && k < j) {  
        i = i + k;  
        j--;  
        k++;  
        System.out.print(i + ", ");  
    }  
    System.out.println(k);  
}
```

Geben Sie für jeden der Aufrufe an, welchen Output diese Methode auf der Konsole produziert.

1. `sonderbar(3, 5);` // _____

2. `sonderbar(5, 3);` // _____

3. `sonderbar(-3, 6);` // _____

4. `sonderbar(2, 12);` // _____

Aufgabe 5 (12 Punkte)

Gegeben seien diese Klassen:

```
class SchwanzLurche extends Lurche {
    void info() {
        System.out.print("Salamander&Molche ");
    }
    void zuerst() {
        // Larven entwickeln ...
        System.out.println("Arme zuerst");
    }
}

class FroschLurche extends Lurche {
    public String toString() {
        return "FroschLurche " + super.toString();
    }
    void zuerst() {
        // Larven entwickeln ...
        System.out.println("Beine zuerst");
    }
}

class Lurche extends Amphibien {
    void info() {
        super.info();
        System.out.print("Amphibien ");
    }

    void zuerst() {
    }

    public String toString() {
        return "Lurche ";
    }
}

class Amphibien {
    void info() {
        System.out.print("Amphibien ");
    }

    void status() {
        System.out.print("bedroht ");
    }

    public String toString() {
        return "Amphibien ";
    }
}

class Feuersalamander extends SchwanzLurche {
    void zuerst() {
        System.out.println("Lebend geboren");
    }
    void status () {
        System.out.print("fast ausgestorben ");
    }
}
```

Das folgende Programmsegment ist ein Klient dieser Klassen. Welchen Output produziert dieses Programmsegment?

```

Amphibien[] meinTerrarium = { new Amphibien(), new Lurche(),
                                new Feuersalamander() };
for (int i = 0; i < meinTerrarium.length; i++) {
    System.out.println(meinTerrarium[i]);
    meinTerrarium[i].info();
    System.out.println();
    meinTerrarium[i].status();
    System.out.println();
}

System.out.println("---");

Lurche[] ihrTerrarium = { new SchwanzLurche(), new FroschLurche(),
                           new Feuersalamander() };
for (int i = 0; i < ihrTerrarium.length; i++) {
    System.out.println(ihrTerrarium[i]);
    ihrTerrarium[i].info();
    System.out.println();
    ihrTerrarium[i].zuerst();
}

```

Antwort:

Aufgabe 6 (8 Punkte)

Gegeben seien diese Klassen:

```
interface One {
    public int value(int v);
}

class Product {
    int cost = 0;

    Product(int c) {
        cost = c;
    }
}

class Superior extends Product
    implements One {
    int profit = 0;

    Superior(int c) {
        super(c);
    }

    Superior(int c, int p) {
        super(c);
        profit = p;
    }

    public int value(int factor) {
        return factor * profit;
    }
}

interface Two {
    public int value(int x);
}

class Special extends Superior {

    Special(int c) {
        super(c);
    }

    Special(int c, int p) {
        super(c, p);
    }
}

class Deluxe extends Product
    implements One, Two {
    int profit = 0;

    Deluxe(int c, int p) {
        super(c);
        profit = 10 + p;
    }

    public int value(int factor) {
        return 11 + factor * profit;
    }
}
```

Geben Sie für dieses Programmsegment den Output oder das Ergebnis an wenn diese Klassen kompiliert und verwendet werden.

```
One[] stock = { new Superior(5, 10), new Special(3, 6), new Deluxe(2, 8) };
for (int i = 0; i < stock.length; i++) {
    System.out.println(stock[i].value(i));
}
```

Antwort:

Aufgabe 7 (6 Punkte)

Welche dieser Hoare Tripel sind (un)gültig? Die Zuweisungen sind Teil einer Java Methode. Alle Variablen sind vom Typ `int` und es gibt keinen Overflow.

1. $\{ x < 0 \ \&\& \ y > 0 \ \&\& \ -x > y \} \quad z = x + y; \quad \{ z > 0 \}$

2. $\{ x > 0 \ || \ y > 0 \} \quad z = x * y; \quad \{ z > 0 \}$

3. $\{ x > 0 \ \&\& \ y > 0 \} \quad z = x / y; \quad \{ z > 0 \}$

4. $\{ \text{true} \}$

```
if (x >= y) {  
    y = x;  
}
```

```
{ y >= x }
```