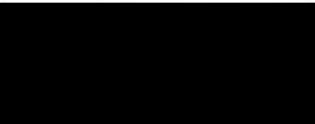


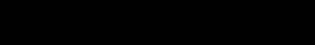
Name



Punkte: 19

Note 1.0

Matr.



Datum: 08.07.2006

Dies ist mein 3. Prüfungsversuch zu dieser Lehrveranstaltung:  
 Ich studiere im Bachelor-Studiengang:

Ja  Nein   
 Ja  Nein

*Pav*

Achtung, bearbeiten Sie die Klausur keinesfalls mit Bleistift oder unter Verwendung roter Farbe. Geben Sie auf jeder Seite Ihren Namen und Ihre Matrikel-Nr. an. Klausuren, die nicht leicht lesbar sind, werden mit 5.0 bewertet.

**Aufgabe 1:**

(max. 1+1+1+1 Punkte)

Was gibt das folgende Programm in welcher Reihenfolge aus, wenn der Benutzer 1, 2, 3 oder 4 eingibt?

```

class Exc1 extends Exception {
    void println() { System.out.println("Exc1"); }
}

class Exc2 extends Exception {
    void println() { System.out.println("Exc2"); }
}

class Exc3 extends Exc1 {
    void println() { System.out.println("Exc3"); }
}

public class ExGame {

    public static int f(int x) throws Exc1, Exc2, Exc3 {

        switch(x){
            case 1: throw new Exc1();
            case 2: throw new Exc2();
            case 3: throw new Exc3();
        }
        return -x;
    }
}

public static void main(String[] args) {
    try {
        int i = IM.readInt();
        int y = f(i);
        System.out.println(y);
    } catch (Exc1 exc1) {
        exc1.println();
    } catch (Exc2 exc2) {
        exc2.println();
    } finally {
        System.out.println("finally");
    }
    System.out.println("Ende");
}
    
```

Input:	1	2	3	4
Output:	Exc1 finally Ende	Exc2 finally	Exc3 finally Ende	-4 finally Ende

4

**Aufgabe 2:**

(max. 4 Punkte)

Schreiben Sie eine Methode, die als Parameter einen String erhält und diesen in ein char-Array mit derselben Reihenfolge und Länge konvertiert und als Return-Wert zurückgibt. Dazu dürfen die Methoden toCharArray() und getBytes(...) hier nicht verwendet werden.

Beispiel: Wenn der Input: "abcdefg" ist, sollte der Output {'a','b','c','d','e','f','g'} entsprechen.

### Aufgabe 3

(entweder 2 oder 0 Punkte)

Ordnen Sie die unten aufgeführten Programmzeilen so, dass ein syntaktisch korrektes Java-Programm entsteht, das keine Fehlermeldungen des Java-Compilers verursacht. Dabei ergibt sich eine andere Reihenfolge der Zeilennummern. Geben Sie die so erhaltene Reihenfolge der Zeilennummern als Ergebnis an.

```

public int plu(int param) {           //1
    t.i = Integer.parseInt(args[0]);   //2
class Test {                           //3
    t.plu(1);                             //4
int helper = i + param;               //5
}                                       //6
public static void main( String args[]){ //7
    Test t = new Test();               //8
int i;                                   //9
return helper * 2;}                   //10

```

3	9	1	5	10	7	8	2	4	6
---	---	---	---	----	---	---	---	---	---

2

### Aufgabe 4

(max. 4 Punkte)

Ergänzen Sie die folgende unvollständige Methode, so dass sie fehlerfrei ist und unter der Bedingung der Vorbedingung die Nachbedingung erfüllt.

```

// Precondition: number ist eine positive gerade Zahl.
// Postcondition: Die Methode hat ein Quadrat auf der Konsole ausgegeben. Das Quadrat hat so
// viele Zeilen bzw. Spalten wie numberOfRows angibt. Drei Quadranten des Quadrats setzen sich
// aus dem Zeichen '+' zusammen. Der Quadrant rechts unten wird aus dem Zeichen '-' gebildet.
//
// Beispiel: printSquarePattern(4); bewirkt folgende Ausgabe:
//
//                                     + + + +
//                                     + + + +
//                                     + + - -
//                                     + + - -

```

```

public static void printSquarePattern(int numberOfRows) {
    for(int row = 0; row < numberOfRows; row++){
        for (int column = 0; column < numberOfRows; column++){
            if (row >= numberOfRows/2 && column >= numberOfRows/2)
                System.out.print("-");
            else
                System.out.print("+");
        }
        System.out.print("\n");
    }
}

```

4

**Aufgabe 5**

(entweder 2 oder 0 Punkte)

Was gibt das folgende Programm aus?

```
class Animal {
    static boolean alive = true;
    String name;

    Animal(String name) {
        this.name = name;
    }

    String sayHello(){
        if(alive)
            return "Hello, my name is " + name;
        else
            return "splash";
    }

    void bye(){
        alive = false;
    }
}
```

```
public static void main(String[] args){
    Animal adam = new Animal("Adam");
    System.out.println(adam.sayHello());
    Animal beata = new Animal("Beata");
    beata.bye();
    System.out.println(adam.sayHello());
    Animal caesar = new Animal("Caesar");
    System.out.println(caesar.sayHello());
}
```

Output:

```
Hello, my name is Adam
splash
splash
```

2

**Aufgabe 6**

(max. 1+1+1+1 Punkte)

Ergänzen Sie das folgende Programm unter Beachtung der bereits fertigen Testmethode (*main*).

- (1a) Implementieren Sie den Konstruktor der Klasse *Student*
- (1b) und den Konstruktor der Klasse *BachelorStudent*.
- (2) Überschreiben Sie in der Klasse *BachelorStudent* die Methode *greet()*.
- (3) Überladen Sie in der Klasse *BachelorStudent* die Methode *greet()*.
- (4) Rufen Sie in der Methode *delegate()* die Methode *greet()* der Klasse *Person* auf.

```
class Person {
    private String name;

    public Person(String name){
        this.name = name;
    }

    public void greet(){
        System.out.println("Hallo");
    }
}
```

```

class Student extends Person {
    private long number;

    public Student(String name, long number){           //(1a)
        super(name);
        this.number = number;
    }
}

```

```

class BachelorStudent extends Student {
    public BachelorStudent(String name, long number){   //(1b)
        super(name, number);
    }
}

```

```

    public void greet(){                               //(2)
        System.out.println("Servus");
    }

```

```

    public void greet(String s){                       //(3)
        if(s != null)
            System.out.println(s);
    }

```

```

    public void delegate(){                            //(4)
        ((Person) this).greet();
    }
}

```

```

public class Test {
    public static void main(String[] args){
        Person p = new Person("Schulze, Ralf");
        p.greet();
        Student s = new Student("Müller, Max", 2214);
        s.greet();
        BachelorStudent b = new BachelorStudent("Krüger, Jo", 2215);
        b.greet();
        b.greet("Hey folks!");
        b.delegate();
    }
}

```

4

9 Aufgabe 2:

```
public return char static char[] getCharArray(  
    String s) {  
    if (s == null || s.length() == 0) ← 2 return null;  
    return new char[s.length()];
```

```
char[] ch =  
    new char[s.length()];  
    for (int i = 0; i < s.length(); i++) {  
        ch[i] = s.charAt(i);  
    }  
    return ch;  
}
```

3 P