

74.51

Minitest 4

Minh Khue Pham (2579036)

15.06.2021 13:55:02 - 14:12:00

Multiple Choice (6 Punkte)

Beantworten Sie die folgenden Fragen mit **wahr** oder **falsch**. Für jede richtige Antwort bekommen Sie 1 Punkt. Für falsche Antworten bekommen Sie 0 Punkte. Wenn Sie keine Antwort geben bekommen Sie 0.5 Punkte.

English instruction: Answer the following questions with **true** or **false**. For every correct answer you obtain 1 point. For incorrect answers you get 0 points. If you do not give an answer you receive 0.5 points.

1. Wenn eine Testsuite 100% Pfadabdeckung hat und nicht fehlschlägt, so ist das Programm korrekt.

1. If a test suite covers 100% of all control paths and does not fail, then the program is correct.

Response:

- ☐ wahr / true
- ☒ falsch / false

2. Beim funktionsbasierten Testen (black box) schreibt man Tests basierend auf einer gegebenen Spezifikation.

2. Black box testing is the design of tests around a given program specification.

Response:

- ☒ wahr / true
- ☐ falsch / false

3. Der dynamische Typ eines Objekts während der Programmausführung entspricht dem statischen Typ oder einem Untertyp des statischen Typs.

3. The dynamic type of an object during program execution is either its static type or a type derived from its static type.

Response:

- ☒ wahr / true
- ☐ falsch / false

4. In Java können zwei Objekte mittels der *equals* Methode auf Gleichheit übergeprüft werden.

4. In Java, two objects can be compared with respect to equality using the *equals* method.

Response:

- ☒ wahr / true
- ☐ falsch / false

5. Durch Überschreibung ersetzen ererbende Klassen die Implementierung einer Methode mit gleicher Signatur.

5. By overriding, subclasses substitute the implementation of a method with the same signature.

Response:

- ☒ wahr / true
- ☐ falsch / false

6. In Java kann ein Wert des Typs *boolean* durch Typumwandlung in einen *int* Wert übersetzt werden.

6. In Java, a value of type *boolean* can be type-casted to a value of type *int*.

Response:

- ☐ wahr / true
- ☒ falsch / false

[[unknown type: ``endAttemptInteraction"]]

2,5/ Spezifikationen (5 Punkte)

Geben Sie ein C0pb-Programm an, welches bezüglich der folgenden Spezifikation total korrekt ist.

English instruction: Write down a C0pb program, which is totally correct with respect to the following specification.

1. $S_1 = \{(\sigma, \sigma') \mid \sigma'x = \sigma y \cdot \sigma y \wedge \sigma'y = \sigma x - \sigma y\}$ (2 Punkte / 2 points)

Response:

```
{
  int x;
  int y;
  int a = y*y;
  int b = x-y;
  x = a;
  y = b;
}
```

Das shadowt $x_i - y_i$ - 1

2. $S_2 = \{(\sigma, \sigma') \mid (\sigma n \geq 0 \rightarrow \sigma'r = 3^{(\sigma n)}) \wedge (\sigma n < 0 \rightarrow \sigma'r = 1)\}$ (3 Punkte / 3 points)

2/3

Response:

```

int r = 1;
int n;
if (n >= 0) {
  for (int i = 0; i < n; i++) {
    r *= 3;
  }
} else {
  r = 1;
}

```

so gehen Deklarationen nicht. - 1
 r, n, ... existieren bereits

[[unknown type: ``endAttemptInteraction'']]

6/

Klassen und Überschreibung (9 Punkte)

Geben sei die folgende Klassenhierarchie:

English instruction: Assume the following class hierarchy:

```

interface A {
    void foo();
}

class B implements A {
    public void foo() {
        System.out.println("B.foo()");
    }

    public void bar(int x) {
        System.out.println("B.bar(int)");
    }
}

class C extends B {
    public void foo() {
        System.out.println("C.foo()");
    }

    public void bar(int x) {
        System.out.println("C.bar(int)");
    }
}

```

Geben Sie die Ausgaben der folgenden Codefragmente an, falls diese kompilieren. Wenn diese nicht kompilieren, begründen Sie warum. (1.5 Punkte pro Fragment)

English instruction: State the output of the following code fragments, if they compile. If they do not compile, explain why. (1.5 points per fragment)

1. `A r = new B(); r.foo();`

because A is an interface

✗
2. `B s = new C(); s.foo();`

C.foo()

✓
3. `B t = new C(); t.bar(42);`

C.bar(42)

✓
4. `C u = new C(); u.foo();`

C.foo()

✓
5. `A v = new C(); v.foo();`

because A is an interface

✗
6. `A w = new A(); w.foo();`

because A is an interface ✓

[[unknown type: ``endAttemptInteraction"]]

Unknown item: ``endAttemptInteraction"

Unknown item: ``endAttemptInteraction"

Unknown item: ``endAttemptInteraction"