Quizzies

Aktuelle Punktzahl: 0 / 16

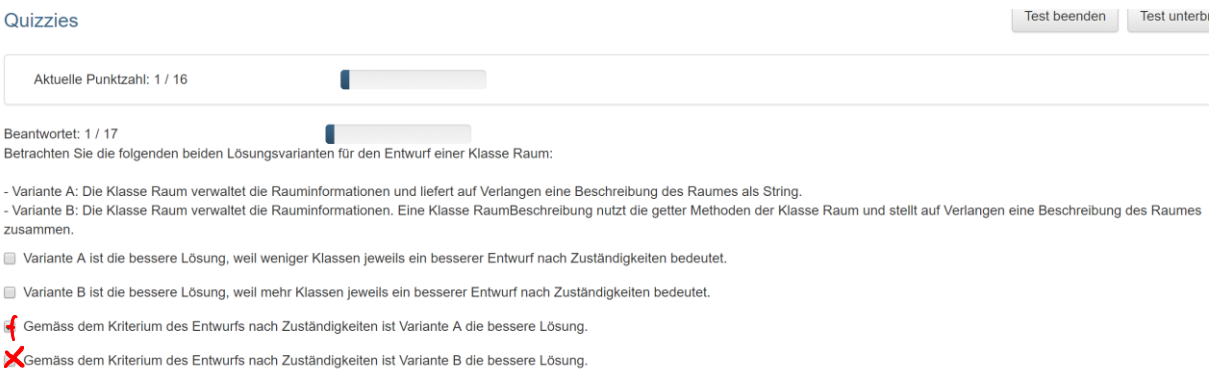
Beantwortet: 0 / 17

Erfüllt die nachfolgende Klasse das Prinzip des Entwurfs nach Zuständigkeiten?

```
public class Filter{

  public OutputStream wendeAnKalmanFilter(InputStream in){
    //mache etwas
    return out;
  }

  public OutputStream wendeAnNewtonInterpolationFilter(InputStream in){
    //mache etwas
    return out;
  }

  public OutputStream wendeAnSplineInterpolationFilter(InputStream in){
    //mache etwas
    return out;
  }
}
```

- [x] Ja
- [ ] Nein

➔ 1 Punkt



➔ 0 Punkte

Quizzies

Aktuelle Punktzahl: 1 / 16

Beantwortet: 2 / 17

Was ist der grösste Vorteil, wenn man die Änderungen im Code lokal behält?

- [x] Es gibt weniger (Anpassungs-)Aufwand bei den andern Klassen.
- [ ] Lose Kopplung und hohe Kohäsion werden automatisch erfüllt.
- [ ] Es gibt weniger Folgefehler in andern Klassen.
- [ ] Wenn sich ein logischer Programmfehler nur lokal einschleicht, dann sind die negativen Auswirkungen sehr gering.

➔ 1 Punkt

Quizzies

Aktuelle Punktzahl: 2 / 16

Beantwortet: 3 / 17

Implizite Kopplung erkennt man typischerweise daran, dass wenn man in einer Klasse etwas am Code ändert, der die Kopplung verursacht, kompilierfehler entstehen.

- [ ] Richtig
- [x] Falsch

➔ 1 Punkt

Quizzies

Aktuelle Punktzahl: 3 / 16

Beantwortet: 4 / 17

Handelt es sich hier zwischen den Klassen A und B um implizite Kopplung?

```
public class A {
    private int[] nummernArray = new int[] { 1, 2, 3 };

    public int gibNummern(int index) {
        return nummernArray[index];
    }
}

public class B {
    public int[] addiereNummern(int addNummer) {
        A a = new A();
        int index0 = a.gibNummern(0) + addNummer;
        int index1 = a.gibNummern(1) + addNummer;
        int index2 = a.gibNummern(2) + addNummer;

        return new int[] { index0, index1, index2 };
    }
}
```

- [ ] Nein
- [x] Ja

➔ 1 Punkt

Quizzies

Aktuelle Punktzahl: 4 / 16

Beantwortet: 5 / 17

Gibt es zwischen den Klassen A und B implizite Kopplung?

```
public class A {
    public final static String ANREDE_FRAU = "Frau";
    public final static String ANREDE_HERR = "Herr";
}

public class B {
    public String formuliereAnredeBrief(String anrede) {
        if (A.ANREDE_FRAU.equals(anrede)) {
            return "Sehr geehrte Frau ";
        } else {
            return "Sehr geehrter Herr ";
        }
    }
}
```

- [x] Ja
- [ ] Nein

➔ 1 Punkt

Quizzies

Aktuelle Punktzahl: 5 / 16

Beantwortet: 6 / 17

Eine Methode mit hoher Kohäsion ist verantwortlich für genau eine wohldefinierte Aufgabe.

- [ ] Falsch
- [x] Richtig

➔ 1 Punkt

Quizzies

Aktuelle Punktzahl: 6 / 16

Beantwortet: 7 / 17

Handelt es sich bei dieser Methode um eine Methode mit hoher Kohäsion?

```
public static int[] entferneNullenAusArray(int[] zahlenArray) {
     int[] neuerArray = new int[zahlenArray.length];
     int counter = 0;


     for (int a : zahlenArray) {
         if (a != 0) {
             neuerArray[counter] = a;
             counter++;
         }
     }
     return Arrays.copyOf(neuerArray, counter);
}
```

- [ ] Nein
- [x] Ja

➔ 1 Punkt

Quizzies

Aktuelle Punktzahl: 7 / 16

Beantwortet: 8 / 17

Handelt es sich bei dieser Methode um eine Methode mit hohe Kohäsion?

```
public int manipuliereZahl(int befehl, int zahl1, int zahl2) {
     int zahl = 0;
     if (befehl == 1) {
        zahl = zahl1 + zahl2;
     }
     if (befehl == 2) {
         zahl = zahl1 - zahl2;
     }
     return zahl;
 }
```

☒ Nein

☑ Ja

➔ 0 Punkte

Quizzies

Aktuelle Punktzahl: 7 / 16

Beantwortet: 9 / 17

Wenn alle Methoden in einer Klasse eine hohe Kohäsion aufweisen, dann weisst die Klasse selber auch eine hohe Kohäsion auf.

☐ Ja

☑ Nein

➔ 1 Punkt

Quizzies

Aktuelle Punktzahl: 8 / 16

Beantwortet: 10 / 17

Weisst diese Klasse eine hohe Kohäsion auf?

```
public class A {
  public int rechneAlterInTagen(Calendar geburtstag) {
    //rechnet das Alter dieser Person/Event in Tagen aus
  }

  public OutputStream wendeFilterAn(InputStream in) {
    // wendet einen Filter an
  }

  public String gibAktuelleWetterPrognose (Location loc){
    // gibt die aktuelle Wetterprogrnose vom dem übergebenen Standort
  }
}
```

- [ ] Ja
- [x] Nein

➔ 1 Punkt

Quizzies

Aktuelle Punktzahl: 9 / 16

Beantwortet: 11 / 17

Welche der unten aufgeführten Aussagen zu Kohäsion sind korrekt?

- [ ] Hohe Kohäsion wird nur erreicht, wenn Aspekte der losen Kopplung auch berücksichtigt werden.
- [x] Hohe Kohäsion erhöht die Wiederverwendbarkeit von Klassen und Methoden.
- [ ] Hohe Kohäsion löst Kopplungsprobleme.
- [x] Hohe Kohäsion verbessert die Lesbarkeit.
- [ ] Hohe Kohäsion vermindert logische Programmierfehler.
- [x] Hohe Kohäsion wird erreicht wenn Klassen und Methoden jeweils für eine wohldefinierte, klar abgegrenzte Aufgabe zuständig sind.

➔ 1 Punkt

Quizzies

Aktuelle Punktzahl: 10 / 16

Beantwortet: 12 / 17

Mit Refactoring kann man sowohl die Kohäsion als auch die Kopplung verbessern.

- [ ] Falsch
- [x] Richtig

➔ 1 Punkt

Quizzies

Aktuelle Punktzahl: 11 / 16

Beantwortet: 13 / 17

Welche Aussagen bezüglich Refactoring sind richtig?

- [x] Eine Art von Refactoring kann sein, dass man eine Klasse in zwei Klassen aufteilt.
- [x] Um zu garantieren, dass ein Refactoring erfolgreich war, sind Tests zwingend notwendig.
- [x] Refactoring dient lediglich zur Verbesserung der Lesbarkeit. Zusätzliche Nutzen ist kaum spürbar.
- [x] Hohe Kohäsion und geringe Kopplung können mit Refactoring erreicht werden.
- [x] Die Wiederverwendbarkeit sollte mit Refactoring verbessert werden.

➔ 0 Punkte

Quizzies

Aktuelle Punktzahl: 11 / 16

Beantwortet: 14 / 17

Sollte bei den Klassen A und B ein Refactoring durchgeführt werden?

```
public class A {
     private B b = null;

     public A() {
         b = new B();
     }
     public void run() {
         b.run();
         int ergebnis = b.gibErgebnis();
     }
 }

 public class B {
     private int ergebnis = 0;

     public void run() {
         ergebnis = 1 + 1;
     }
     public int gibErgebnis(){
         return ergebnis;
     }
 }
```

- [ ] Nein
- [x] Ja

➔ 1 Punkt

Quizzies

Aktuelle Punktzahl: 12 / 16

Beantwortet: 15 / 17

Welche Behauptungen zu Switch sind richtig?

- [x] Eine switch-Anweisung kann so geschrieben werden, dass sie aus allen verschiedenen Anweisungsfolgen nur eine Möglichkeit auswählt.
- [x] Eine switch-Anweisung kann teilweise eine if- und else-Anweisung ersetzen und hat obendrein den Vorteil, dass sie die Übersichtlichkeit verbessert.
- [ ] Die switch-Anweisung ist gleich mächtig wie if- und else-Anweisungen.

➔ 1 Punkt