

Objektorientierte Programmierung

Probeklausur

HS Bochum - WS 18/19

Dr.-Ing. Darius Malysiak

20.12.2018

Erlaubte Hilfsmittel:

1. Ein nicht programmierbarer Taschenrechner.
2. Zwei nicht kopierte, beidseitig handschriftlich beschriebene Hilfsblätter mit einer Größe von max. DIN A4.

Insgesamt sind 100 Punkte zu erreichen, zum bestehen der Klausur sind 50 Punkte nötig.

Streichen Sie Texte, welche nicht bewertet werden sollen deutlich durch! **Halten Sie Ihre Antworten kurz und prägnant!**

Zeit: 120 Minuten

Aufgabe 1) Versionsverwaltung - 25 Punkte

1. Erläutern Sie den Begriff Versionsverwaltung im Kontext von Software-Entwicklung. 5 Punkte
2. Erläutern Sie die folgenden git Begriffe: Branch, Commit, Tag, Merge und Staging. 5 Punkte
3. Nennen Sie im Hinblick auf Quelltextmanagement in der Softwareentwicklung ein Problem, welches trotz Versionsverwaltung nicht gelöst werden kann. 5 Punkte
4. Gegeben ist das folgende Szenario; Neben dem Master-Branch wird in einem davon abgezweigten Feature-Branch ein neues Feature des Produkts entwickelt. Nach erfolgter Entwicklung und erfolgreichen Tests wird der Feature-Branch in den Master gemerged. Beschreiben Sie einen hypothetischen Fall in welchem dieses Vorgehen trotzdem zu Fehlern im Softwareprodukt führen kann. 10 Punkte

Aufgabe 2) Java - 25 Punkte

1. Was sind Generics? Schreiben Sie eine Klasse 'A', welche einen Generic Parameter 'T' für ein Attribut verwendet. 4 Punkte
2. Instanzieren Sie Ihre Klasse 'A' mit einem Upper-Bound für 'T'. 4 Punkte
3. Erstellen Sie eine Kindklasse 'B' von 'A', welche keinen Generic Parameter benötigt. 4 Punkte
4. Was bedeutet der Begriff Lower-Bound im Kontext von Generics? Was ist hierbei zu beachten? 4 Punkte
5. Erläutern Sie den folgenden Quellcode. Falls eine Ausgabe vorhanden ist so geben Sie diese an. 9 Punkte

```

package a;

import java.util.stream.IntStream;
import java.util.stream.Stream;

public class App
{
    public static void main( String[] args )
    {
        IntStream.range(3,22).forEach(
            i -> {
                Stream.of(i).map(j -> {
                    return "" + j;
                }).filter( s -> s.contains("2") ).forEach(System.out::println);
            }
        );
    }
}

```

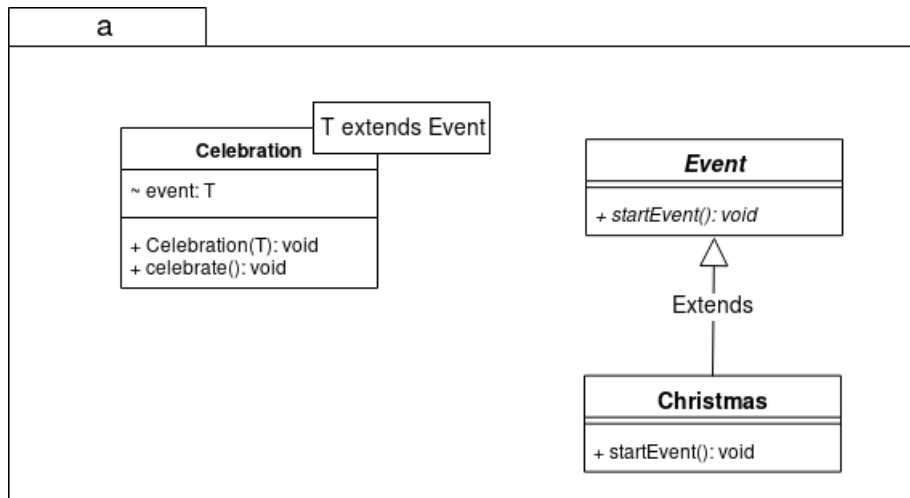
Aufgabe 3) Design Patterns - 25 Punkte

1. Erläutern Sie das Builder Pattern sowie dessen Vor- und Nachteile. 5 Punkte
2. Implementieren Sie das Singleton Pattern inkl. einer Verwendung dessens in der Main Methode. 8 Punkte
3. Zeichnen Sie ein UML Diagramm zum Singleton Pattern. 4 Punkte
4. Implementieren Sie die Klassenstruktur des folgenden UML Diagramms. Ihre Implementierung soll es ermöglichen, dass der folgende Ausdruck die Ausgabe 'Merry Christmas' erzeugt. 8 Punkte

```

Celebration<Christmas> c = new Celebration<>(new Christmas());
c.celebrate();

```



Aufgabe 4) OOP Software Engineering - 25 Punkte

TO BE DONE