Heinrich-Heine-Universität Düsseldorf
Institut für Informatik
Dr. J. Bendisposto

HEINRICH HEINE
UNIVERSITÄT
DÜSSELDORF

23. Juli 2020

Klausur

Professionelle Softwareentwicklung
Sommersemester 2020

Nachname: _____ Vorname: _____

Matrikelnummer: _____

Unterschrift: _____

Zugelassene Hilfsmittel: Eine handschriftlich beschriebene A4 Seite. Matrikelnummer und Name muss auf der Seite stehen

Der Klausur hängen Seiten für weitere Notizen an. Sie dürfen keine eigenen Blätter verwenden!

Diese Klausur enthält 10 nummerierte Seiten. Prüfen Sie bitte zuerst, ob alle Seiten vorhanden sind.

Aufgrund der Coronaschutzverordnung sind Sie verpflichtet persönliche Informationen zur Rückverfolgbarkeit auf einem separaten Zettel mit anzugeben. Bitte füllen Sie diesen Zettel zunächst aus.

Diesen Teil bitte nicht ausfüllen:

Aufgabe	1	2	3	4	5	6	Σ
Punktzahl	5	8	10	10	5	12	50
Erreicht							

Aufgabe 1

[5 Punkte]

Wir haben eine Liste von Strings, die wir ausgeben wollen. Schreiben Sie den Code um diese Liste auf der Standardausgabe zu printen, indem Sie die in Java für solche Aufgaben idiomatische Schleifenform verwenden. Es gibt nur Punkte für die **idiomatische** Lösung.

```
public void print(List<String> names) {




}
```

Aufgabe 2

[8 Punkte]

Gegeben sei folgendes funktionales Interface:

```
@FunctionalInterface
interface Operator<T> {
    T apply(T t);

    default String print(T input) {
        return "so sieht eine default Methode aus";
    }
}
```

Schreiben Sie eine default Methode `butBefore`, die wie folgt benutzt wird:

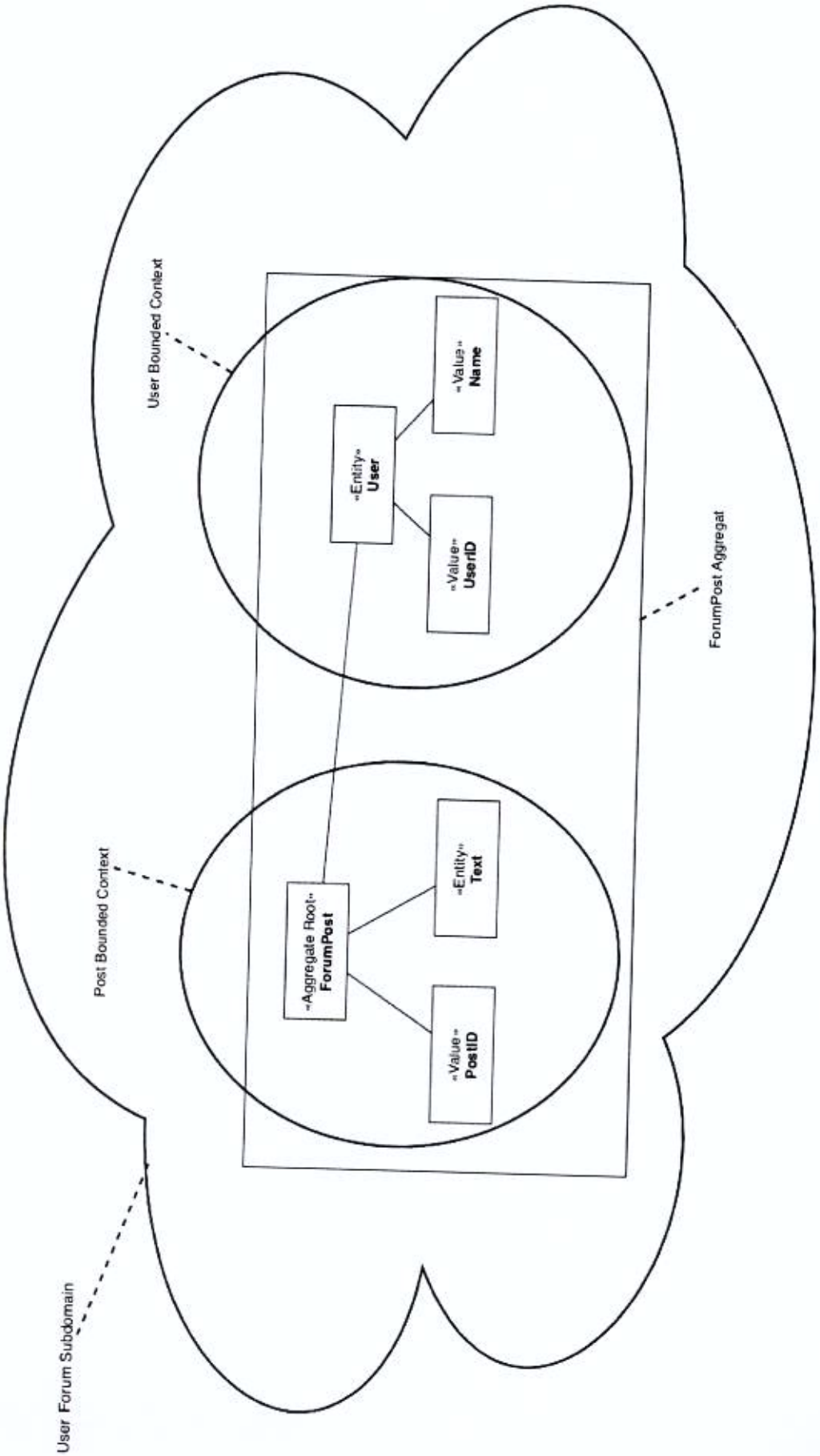
```
Operator<Integer> inc = x -> x + 1;
Operator<Integer> square = x -> x * x;
Operator<Integer> foo = square.butBefore(inc);
// foo quadriert die Eingabe, führt aber vorher inc aus
System.out.println(foo.apply(3)); // 16
```

Aufgabe 3

[10 Punkte]

Sie bekommen *auf der nächsten Seite* einen Vorschlag für eine Aufteilung einer Software nach dem Domain-Driven Design vorgelegt. Das Design enthält einen sehr groben Fehler. Korrigieren Sie diesen einen groben Fehler. Beachten Sie, dass die Bounded Contexte durch die Ubiquitous Language vorgegeben sind.

`Grafik auf nächster Seite beachten und die Lösung hier beschreiben.`


User Forum Subdomain
User Bounded Context
«Entity»
User
«Value»
Name
«Value»
UserID
ForumPost Aggregat
Post Bounded Context
«Aggregate Root»
ForumPost
«Entity»
Text
«Value»
PostID

Aufgabe 4

[10 Punkte]

Eine Implementierung aus Pflichtabgabe 7 enthält folgenden Codeschnipsel:

```
// Zugangspunkt ohne Nutzungsreserve (z.B. Park)
public class ZugangsPunkt {

  private void handlePersonEntering() {
    switch(scanner.category()) {
      case SONDERBERECHTIGUNG: handleSonderberechtigung(); break;
      default: handleNormalPerson();
    }
  }

  // ...

}

// Zugangspunkt mit Nutzungsreserve (z.B. Geschäft)
public class BusinessZugangsPunkt extends Zugangspunkt {

  private void handlePersonEntering() {
    switch(scanner.category()) {
      case SONDERBERECHTIGUNG: handleSonderberechtigung(); break;
      case ANGESTELLT: handleAngestellt(); break;
      default: handleNormalPerson();
    }
  }

  // ...

}
```

Prüfen Sie den vorliegenden Code im Hinblick auf Verletzungen der SOLID Prinzipien. Geben Sie für jedes der Prinzipien an, ob es verletzt wurde und begründen Sie Ihre Entscheidung in je einem Satz. Wenn keine Verletzung vorliegt, sollten Sie das auch kurz begründen.

Verwenden Sie zur Beantwortung der Aufgabe die Tabelle auf der nächsten Seite.

	Name des Prinzips	Verletzt?	Begründung
S			
O			
L			
I			
D			

Aufgabe 5

[5 Punkte]

Gegeben ist folgender Code:

```
@Getter @Setter @EqualsAndHashcode
public class Punkt {
  private double x, y;
}

@Getter @Setter @EqualsAndHashcode @RequiredArgsConstructor
public class Gerade {
  private final Punkt a, b;
  public double getSteigung() {
    return (a.getY() - b.getY()) / (a.getX() - b.getX());
  }
  public double getAchsenabschnitt() {
    return a.getY() - getSteigung() * a.getX();
  }
}
```

Welche Probleme könnten bei diesen Klassen im Zusammenhang mit nebenläufigen Programmen auftreten und wodurch werden diese Probleme verursacht? Führen Sie **minimale** Änderungen durch, sodass die Probleme nicht mehr auftreten können.

Aufgabe 6

[12 Punkte]

Angenommen, folgender Test wäre für Pflichtabgabe 7 eingereicht worden:

```
public class Zugangskontrolle {
  // ...
  public long testAlter() {
    return Duration.between(person.getTestDatum(), LocalDateTime.now())
        .get(ChronoUnit.HOURS);
  }
}

@Test
void CovidTestdatum(){
  Person person = mock(Person.class);
  LocalDateTime d = LocalDateTime.of(2020,07,23,0,0) // 23.7.2020 Mitternacht
  when(person.getTestDatum()).thenReturn(d);
  Zugangskontrolle zugang = mock(Zugangskontrolle.class);
  assertThat(zugang.testAlter()).isEqualTo(d.getHour());
}
```

Ist der Test grün oder rot? Was genau testet dieser Test? Begründen Sie Ihre Antwort.

Raum für Notizen. Sollten Sie hier Lösungen formulieren, dann machen Sie das für uns kenntlich und markieren Sie für welche Aufgabe sich hier die Lösung befindet.

Raum für Notizen. Sollten Sie hier Lösungen formulieren, dann machen Sie das für uns kenntlich und markieren Sie für welche Aufgabe sich hier die Lösung befindet.