Name:

Matrikelnummer:

Bearbeitungszeit: 120 min.

Aufgabe 1. Gegeben sind

- die Klassen `Layout` und `Produkt`,
- die Klasse `Fabrik` mit dem Konstruktor

  ```
  Fabrik(Layout einLayout),
  ```

  der eine Fabrik mit dem angegebenen Layout erzeugt, und der Methode

  ```
  Produkt fertige(),
  ```

  die eine Fertigung simuliert und das gefertigte Produkt zurückgibt,
- sowie folgendes Interface:

```
interface Zufallsgenerator
{
    boolean istFehlerhaft();
}
```

Implementieren Sie die Unterklasse `FabrikMitFehlern` von `Fabrik` mit dem Konstruktor

```
FabrikMitFehlern(Layout einLayout, Zufallsgenerator zufall)
```

und überschreiben Sie die Methode `fertige()` so, dass `null` zurückgegeben wird, wenn `zufall.istFehlerhaft()` als Ergebnis `true` liefert, und ansonsten das in der Oberklasse gefertigte Produkt. [8 Punkte]

Schreiben Sie auch eine Klasse `Bernoulli`, die `Zufallsgenerator` implementiert und über den Konstruktor

```
Bernoulli(double prob)
```

verfügt, sodass `istFehlerhaft()` mit Wahrscheinlichkeit `prob` als Ergebnis `true` liefert. [2 Punkte]

Aufgabe 2. Gegeben ist die Klasse `Aktie` und die Klasse `Boerse`, die Börsenkurse von Aktien verwaltet. Dabei liefert die Methode

```
Set<Aktie> getAktien()
```

alle Aktien, die am aktuellen Handelstag gehandelt werden können. Die Methode

```
double[] getKurs(Aktie eineAktie)
```

liefert die Kurse der Aktie vom ersten Handelstag der Aktie bis zum aktuellen Handelstag (für `double[] kurs = getKurs(eineAktie)` ist `kurs[kurs.length-1]` der aktuelle Kurs der Aktie und `kurs[0]` der Kurs am ersten Handelstag der Aktie).

Schreiben Sie eine Klasse `Empfehlung`, die eine Menge von empfohlenen Aktien verwaltet. Diese Klasse soll über den Konstruktor

```
Empfehlung(Boerse eineBoerse)
```

verfügen, der eine leere Menge von empfohlenen Aktien anlegt und die Börse angibt, an der die Aktien gehandelt werden. Die Methode

```
Set<Aktie> getEmpfohlene()
```

soll die Menge der derzeit empfohlenen Aktien zurückgeben, und die Methode

```
void update(int anzFallend, int anzSteigend)
```

soll jene Aktien aus der Empfehlung entfernen, deren Kurse an jedem der letzten `anzFallend` Handelstagen gefallen sind (verglichen mit dem Vortag), und jene Aktien in die Empfehlung aufnehmen, deren Kurse an jedem der letzten `anzSteigend` Handelstagen gestiegen sind.

Aufgabe 3. Für diese Aufgabe werden Dominosteine als `int`-Arrays der Länge 2 dargestellt. Schreiben Sie eine Klasse `Anordnung` mit der Methode

```
List<int[]> getAnordnung(List<int[]> dominosteine),
```

die eine Anordnung `ergebnis` aller übergebenen Dominosteine zurückgibt, sodass für `i=1,...,ergebnis.size()-2` eine Zahl von `ergebnis.get(i)` in `ergebnis.get(i-1)` und die andere Zahl von `ergebnis.get(i)` in `ergebnis.get(i+1)` vorkommt. Wenn es keine solche Anordnung gibt, dann soll die Methode `null` zurückgeben.

Beispiel: Die Anordnungen `{3,4},{4,6},{6,1}` und `{3,4},{3,2},{4,2},{4,4}` sind korrekt, während es für die Dominosteine `{3,4},{3,2},{3,1}` keine korrekte Anordnung gibt.