

Aufgabe 1.1

Implementieren Sie eine Klasse „Person“

- A) Die Klasse Person hat eine Instanzvariable „name“ (String)
Die Klasse Person hat ein Feld („pilot“), welches besagt, ob diese Person Pilot (Fliegerei) ist oder nicht (boolean) (Instanzvariable)
- B) Implementieren Sie den Konstruktor mit folgender Signatur:
`public Person(String name)`
Wenn dieser Konstruktor aufgerufen wird, ist die Person **nicht** Pilot!
- C) Implementieren Sie einen zweiten (überladenen) Konstruktor.
Beim Erzeugen eines Objektes soll nebst dem Namen ein zweiter Parameter übergeben werden, welcher der „pilot“ Variable zugewiesen wird.
- D) Implementieren Sie eine Methode mit dem Namen „getName“, die den Namen der Person zurück gibt.
- E) Implementieren Sie eine Methode mit dem Namen „isPilot“ die den boolean Wert zurück gibt, welche besagt, ob die Person Pilot ist oder nicht.

Aufgabe 1.2

Die Aufgabe 1.1 ist Voraussetzung zum Lösen dieser Aufgabe. Lösen Sie zuerst Aufgabe 1.1, bevor Sie Aufgabe 1.2 lösen.

Implementieren Sie eine Klasse „Flugzeug“

- A) Die Klasse „Flugzeug“ hat eine Instanzvariable, welche die maximale Anzahl an Passagieren (exklusive Pilot) speichert.
Die Klasse „Flugzeug“ hat ein Pilot (Instanzvariable vom Typ Person)
Die Klasse „Flugzeug“ hat Passagiere (Instanzvariable vom Typ ArrayList, welche Person-Objekte halten kann)
- B) Implementieren Sie den Konstruktor der „Flugzeug“-Klasse. Beim Erzeugen eines Flugzeug-Objektes wird die maximale Anzahl an Passagieren für dieses Flugzeug-Objekt übergeben.

C) Implementieren Sie die Methode mit folgender Signatur:

```
/**
 * Weist dem Flugzeug den Pilot zu.
 * @param pilot der Pilot
 */
public void setPilot(Person pilot)
```

Bedingung:

Der Pilot kann nur zugewiesen werden, wenn die Person, die Übergeben wird, Pilot ist. Wenn die Person nicht Pilot ist, so wird null zugewiesen. Geben Sie eine Meldung auf der Konsole aus, welche darüber informiert, wenn der Pilot nicht gesetzt werden konnte. (Achten Sie auf eine mögliche NullPointerException)

D) Implementieren Sie die Methode mit folgender Signatur:

```
/**
 * Fügt einen Passagier hinzu
 * @param passagier ein Passagier
 * @return true, wenn der Passagier hinzugefügt werden konnte, sonst false
 */
public boolean addPassagier(Person passagier)
```

Bedingungen:

- Die Person darf nicht null sein
- Es dürfen sich nicht mehr Passagiere (Personen exklusive Pilot) im Flugzeug befinden, als die maximale Anzahl Passagiere. Ein Passagier darf also nur hinzugefügt werden, falls im Flugzeug noch freie Plätze vorhanden sind.
- Ein Passagier darf nicht zweimal hinzugefügt werden (Hinweis: Javadoc verwenden)

E) Implementieren Sie die Methode mit folgender Signatur:

```
/**
 * Entfernt einen Passagier aus dem Flugzeug.
 * @param passagier die Position (index)
 * des Passagiers, der entfernt werden soll.
 * @return true, wenn der Passagier entfernt werden konnte,
 * sonst false
 */
public boolean removePassagier(int passagier)
```

Achten Sie auf eine mögliche IndexOutOfBoundsException

F) Implementieren Sie eine Methode mit dem Namen „getPilot“, welche den Piloten zurückgibt.

G) Implementieren Sie eine Methode „getPassagiere“, welche die Passagierliste (ArrayList von Personen) zurückgibt.

H) Implementieren Sie eine Methode mit dem Namen „canFly“. Diese Methode gibt true zurück, wenn ein Pilot im Flugzeug ist. Wenn kein Pilot im Flugzeug ist, gibt sie false zurück.

I) Implementieren Sie eine Methode mit folgender Signatur:

```
public void printPassagierliste()
```

Diese Methode printet (System.out.println()) alle Passagiernamen auf die Konsole aus. Verwenden Sie einen Loop!

Aufgabe 1.3

Zeichnen Sie das Klassendiagramm zur Aufgabe 1.1 und 1.2

Aufgabe 1.4

Zeichnen Sie das Objektdiagramm zur Aufgabe 1.1 und 1.2 mit der Annahme, dass sich ein Pilot und zwei Passagiere im Flugzeug aufhalten.

Aufgabe 2

A) Gegeben ist folgender Code:

```
String name = "Hans Fischer";
```

Was Gibt folgende Methode zurück?

```
name.length();
```

B) Gegeben ist folgender Code:

```
System.out.println(18 % 4);
```

Was wird auf der Konsole ausgegeben?

C) Schreiben Sie den Default-Konstruktor (ohne Parameter) für eine Klasse mit dem Namen „Haus“ auf:

D) Gegeben ist folgender Code:

```
int a = 0;
while (a < 10) {
    int b = 0;
    b++;
    a++;
}
System.out.println(b);
```

Was wird auf der Konsole ausgegeben? (Eine Antwort ankreuzen)

A) 10

B) 1

C) Das kompiliert nicht

- E) Gegeben ist folgender Array:
`int[] intArr = new int[25];`

Welche Aussage ist nach der Ausführung dieses Codes richtig? Achtung! Eine oder mehrere Antworten sind richtig. Wenn eine falsche angekreuzt wird gibt es keine Punkte.

- A) `intArr[24]` ist 0
 - B) `intArr[24]` ist undefiniert
 - C) `intArr[25]` ist 0
 - D) `intArr[0]` ist null
 - E) `intArr.length` ist 25
- F) Gegeben sei ein String-Array mit namen `strArray`. Schreiben Sie einen for-loop und einen for-each-loop, der alle Strings auf der Konsole ausgibt.

G) Gegeben ist folgender Code:

```
public class Test {  
  
    private String einString = "hallo";  
  
    public Test(String einString) {  
        System.out.println(this.einString);  
        System.out.println(einString);  
    }  
  
}
```

In einer anderen Klasse wird folgender Code aufgerufen:

```
Test t = new Test("wie gehts?");
```

Was wird auf der Konsole ausgegeben?

H) Gegeben ist folgender Code:

```
public class Test {  
  
    private static int zaehler;  
    private int zaehler2;  
  
    public Test() {  
        zaehler++;  
        zaehler2++;  
    }  
  
}
```

Welche Werte haben zaehler und zaehler2 nach dem Erstellen von 3 Objekten vom Typ „Test“ im zuletzt erstellten Objekt t3?

```
Test t1 = new Test();  
Test t2 = new Test();  
Test t3 = new Test();
```

I) Gegeben ist folgender Code:

```
public class Test {  
  
    public void hello() {  
        System.out.println("Hello");  
    }  
  
    public void hello(String hello) {  
        System.out.println(hello);  
    }  
  
    public Test() {  
        System.out.println("Hallo Test");  
    }  
  
    public Test(String string) {  
        System.out.println("Hallo " + string);  
    }  
}
```

In einer anderen Klasse wird folgender Code ausgeführt:

```
Test t = new Test();  
t.hello("wie gehts?");
```

Was wird auf der Konsole ausgegeben?