

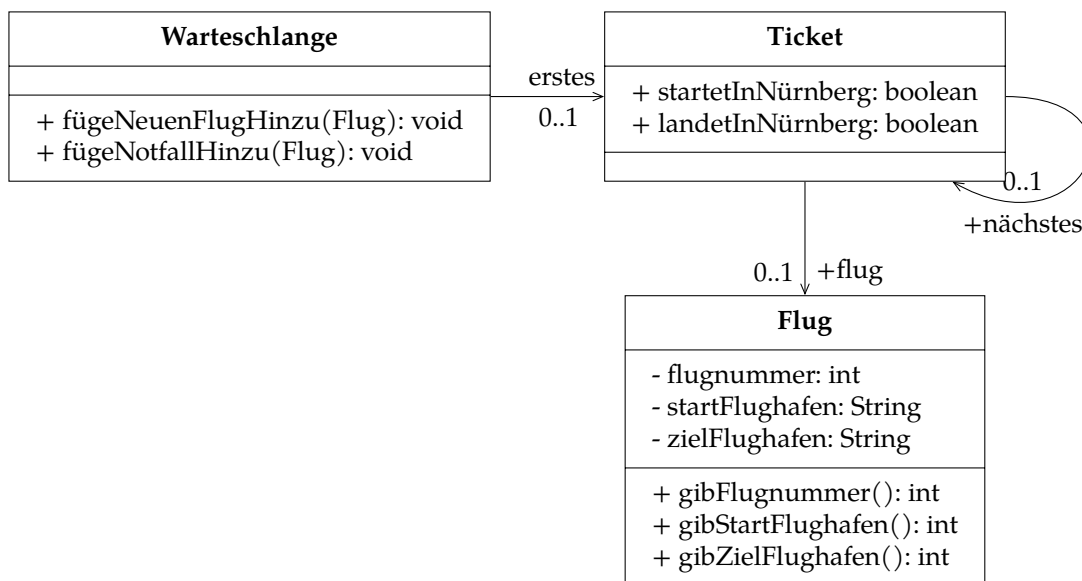
Listen

(Flughafen)

Stichwörter: Warteschlange (Queue)

Am Nürnberger Flughafen starten und landen täglich viele Flugzeuge. Der Flughafen verfügt jedoch nur über eine einzige Start- bzw. Landebahn, sodass Starts und Landungen gemeinsam koordiniert werden müssen. Für die interne Verwaltung, welcher Flieger als nächstes bearbeitet werden soll, wird eine neue Software entwickelt. Sie haben die Aufgabe einen Teil dieser Software zu erstellen.

Das folgende UML-Klassendiagramm gibt einen Überblick über den für Sie relevanten Teil der Software:



Implementieren Sie die beiden Methoden `fügeNeuenFlugHinzu(Flug)` und `fügeNotfallHinzu(Flug)` der Klasse `Warteschlange`!

- In der Methode `fügeNeuenFlugHinzu()`
 - soll ein neu übergebener Flug der Warteschlange hinzugefügt werden.
- In der Methode `fügeNotfallHinzu()`
 - soll für den übergebenen Flug überprüft werden, ob der Flug bereits in der Warteschlange vorkommt oder ob es sich um einen komplett neuen Flug handelt.
 - soll der Notfall die erste Priorität in der Warteschlange erhalten, özwingend als nächstes abgearbeitet werden.
 - soll die korrekte Funktionalität der Warteschlange weiterhin gegeben sein.

Jeder Flug wird in der Warteschlange mit einem eigenen Ticket verwaltet. Die Funktionalitäten der Tickets und der Flüge entnehmen Sie dem Quelltext.

Klasse „Flug“

```
public class Flug {
    private int flugnummer;
    private String startFlughafen;
    private String zielFlughafen;

    public Flug(int flugnummer, String startFlughafen, String zielFlughafen) {
        this.flugnummer = flugnummer;
        this.startFlughafen = startFlughafen;
        this.zielFlughafen = zielFlughafen;
    }

    public int gibFlugnummer() {
        return flugnummer;
    }

    public String gibStartFlughafen() {
        return startFlughafen;
    }

    public String gibZielFlughafen() {
        return zielFlughafen;
    }
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/listen/flughafen/Flug.java](https://github.com/bschlangaul/aufgaben/aud/listen/flughafen/Flug.java)

Klasse „Ticket“

```
public class Ticket {
    public Flug flug;
    public Ticket nächstes;
    public boolean startetInNürnberg = false;
    public boolean landetInNürnberg = false;

    public Ticket(Flug flug) {
        this.flug = flug;
        if (flug.gibStartFlughafen().equals("NUE")) {
            startetInNürnberg = true;
        } else {
            landetInNürnberg = true;
        }
    }
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/listen/flughafen/Ticket.java](https://github.com/bschlangaul/aufgaben/aud/listen/flughafen/Ticket.java)

Klasse „Warteschlange“

```
public class Warteschlange {
    Ticket erstes = null;
```

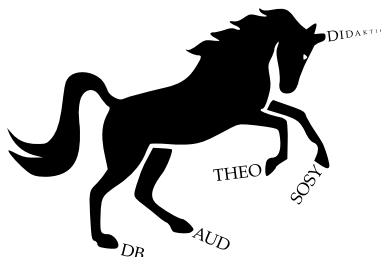
```

public void fügeNeuenFlugHinzu(Flug flug) {
    Ticket zähler = erstes;
    if (erstes == null) {
        erstes = new Ticket(flug);
    } else {
        while (zähler.nächstes != null) {
            zähler = zähler.nächstes;
        }
        zähler.nächstes = new Ticket(flug);
    }
}

public void fügeNotfallHinzu(Flug flug) {
    if (erstes.flug.gibFlugnummer() == flug.gibFlugnummer()) {
        return;
    }
    Ticket zähler = erstes;
    Ticket zähler2 = erstes.nächstes;
    Ticket notfall = null;
    while (zähler2 != null) {
        if (zähler2.flug.gibFlugnummer() == flug.gibFlugnummer()) {
            notfall = zähler2;
            zähler.nächstes = zähler2.nächstes;
            break;
        } else {
            zähler = zähler2;
            zähler2 = zähler2.nächstes;
        }
    }
    if (notfall == null) {
        notfall = new Ticket(flug);
    }
    notfall.nächstes = erstes;
    erstes = notfall;
}
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/listen/flughafen/Warteschlange.java](https://github.com/bschlangaul/aufgaben/aud/listen/flughafen/Warteschlange.java)



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der TeX-Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: https://github.com/bschlangaul-sammlung/examens-aufgaben-tex/blob/main/Module/30_AUD/70_Listen/20_Warteschlange/Aufgabe_Flughafen.tex