Žilinská univerzita

Fakulta riadenia a informatiky



ALGORITMY A ÚDAJOVÉ ŠTRUKTÚRY 2

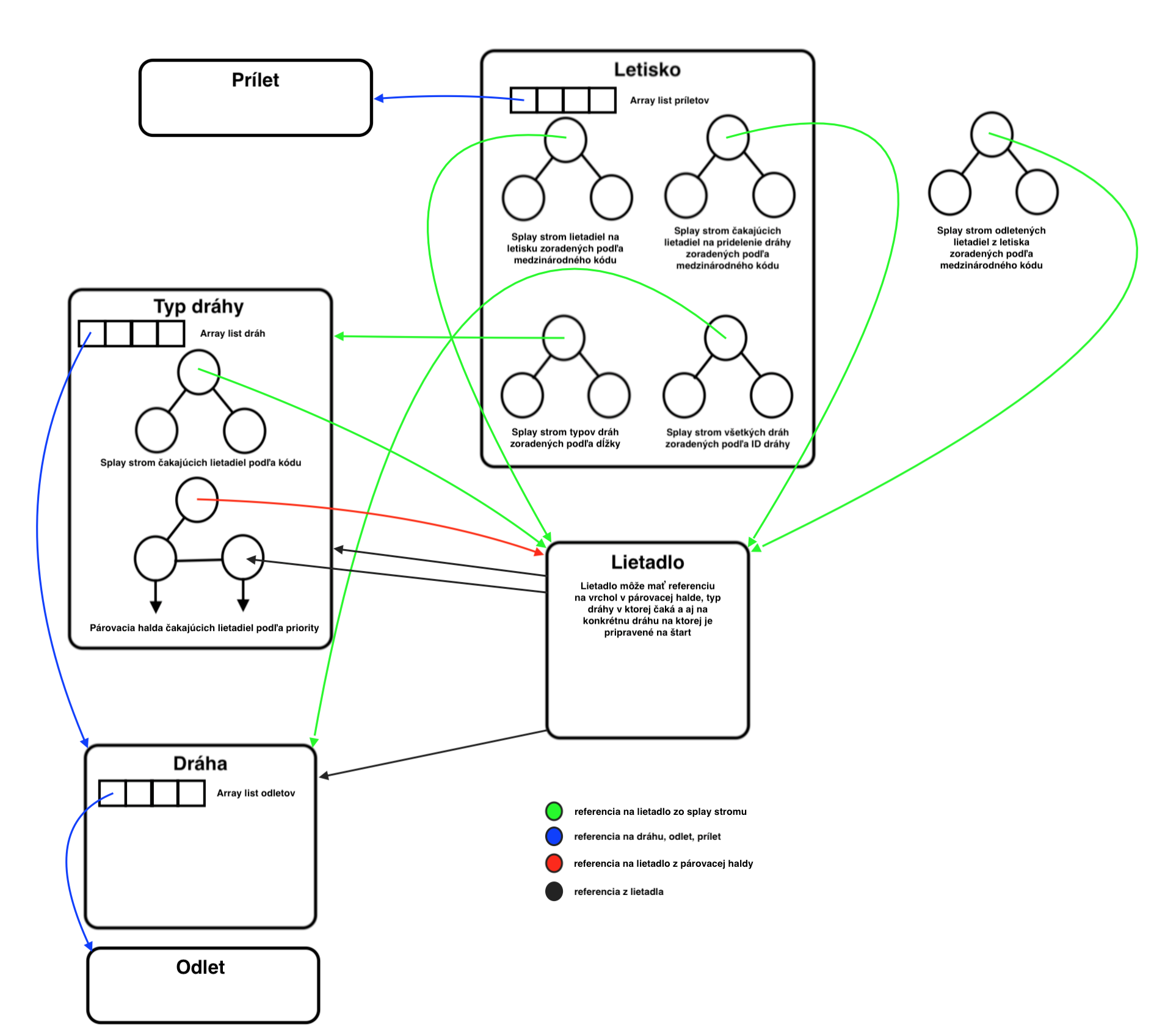
Semestrálna práca 1

Dávid Pavličko

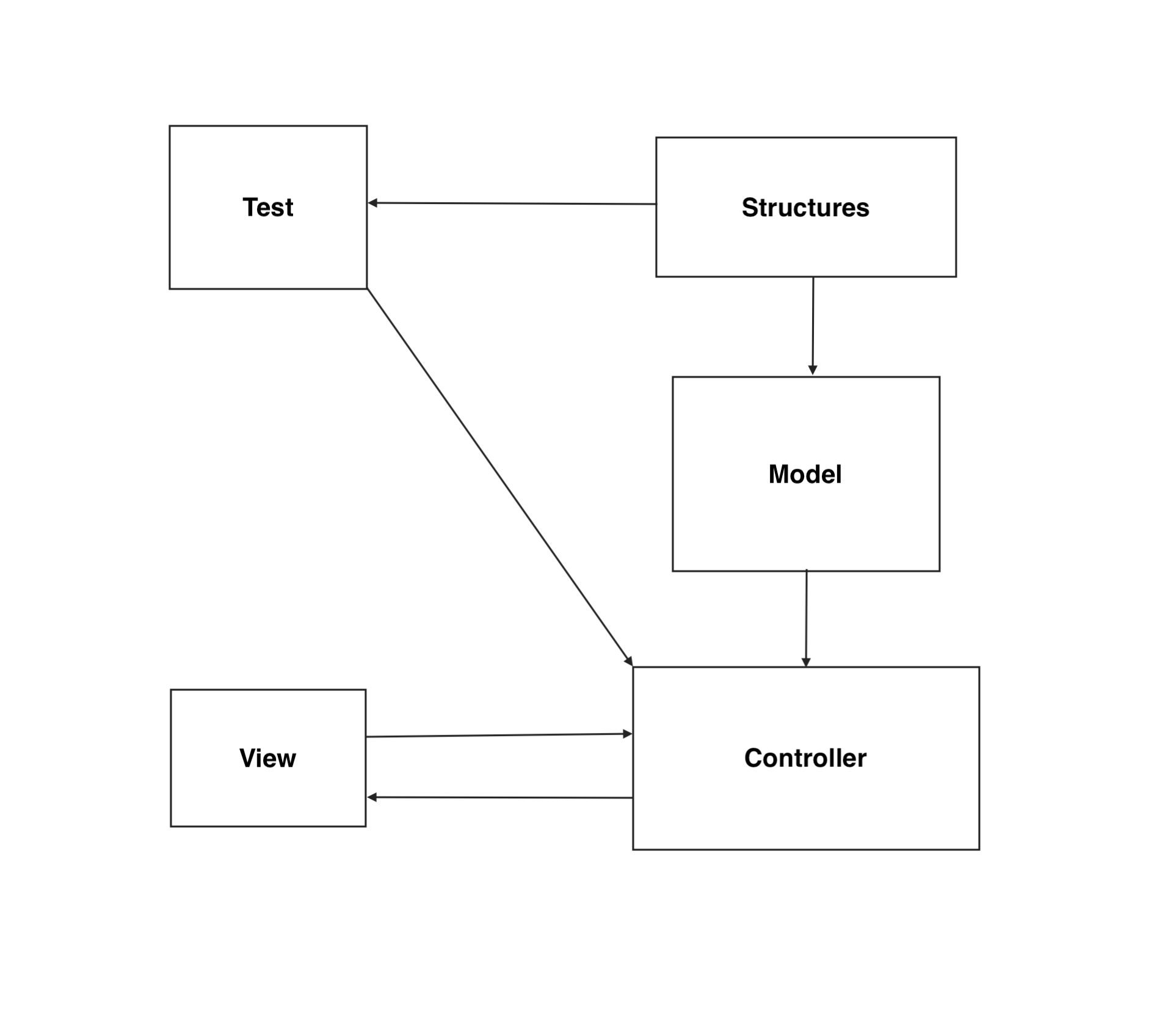
5ZZS12

2019/2020

# Návrh údajovej štruktúry

Na ukladanie dát v semestrálnej práci som použil štruktúru Splay strom, Párovaciu haldu a Array list.

# Diagram balíčkov



## Popis diagramu balíčkov a tried

### 1. Structures

Balíček štruktúr obsahuje vlastnú implementáciu údajových štruktúr, ktoré boli použité vrámci semestrálnej práce a na testovanie.

* **Splay strom**
* **Párovacia halda**
* **Front**
* **Linked list**
* **Binárny vyhľadávací strom**

### 2. Test

Balíček obsahuje testovací generátor, ktorý bol prezentovaný počas 1. a 2. kontroly rozpracovania

### 3. Model

Triedy objektov pre samotnú aplikáciu sú uložené v tomto balíčku.

* **Airport** –letisko je hlavnou triedou celej aplikácie, ktorá sa využíva skoro všade. Vie o celom dianí v systéme a je možné ju zdieľať naprieč celou aplikáciou.
* **Airplane** – lietadlo prezentuje systémového agenta s ktorým budeme pracovať a zobrazovať jeho stav koncovému užívateľovi v mobilnej aplikácií
* **RunwayType** – typ dráhy zaobaľuje viacero dráh rovnakého typu, kvôli skupinkovaniu lietadiel do čakacích frontov
* **Runway** – odletová dráha si pamätá všetky odlety lietadiel a aj aktuálne lietadlo pripravené na štart
* **Arrival** – trieda príchod zaznamenáva čas a referenciu na lietadlo, ktoré priletelo na letisko
* **Departure** – táto trieda uchováva okrem referencie na lietadlo a presného času odletu aj čas podania požiadavky o odlet a aj príchod lietadla pre zobrazenie celkového prehľadu lietadiel odletených z letiska
* **DateTime** – trieda pre vlastný manažment dátumu a času v systéme
* **AirplaneTableViewCell** – model pre View, ktorý prezentuje jednu bunku v dynamickom liste mobilnej aplikácie
* **Spinner** – ikona načítavania, ktorá je volaná hlavným vláknom

### 4. Controller

Kontrolóri spájajú modely s rozhraním, spracúvajú informácie od používateľa aplikácie a po vykonaní operácií s modelmi posielajú výsledok naspäť koncovému používateľovi. Každý „pohľad“ má svoj vlastný kontrolór.

### 5. View

Obsahuje diagram všetkých pohľadov, ktorý je prepojený s kontrolórmi.

# Zoznam požadovaných operácií a ich zložitosti

## 1. Vloženie informácií o prílete lietadla do systému

Aplikácia eviduje všetky lietadla, ktoré sú na letisku v samotnom Splay strome „airplanes“, ktorá má zložitosť P(log(n)), ihneď sa ukladá referencia na lietadlo aj do Splay stromu pre čakajúce lietadla na dráhu O(log(n)), ale ak priletí lietadlo, ktoré sa už v systéme nachádza vyhľadáva sa aj v Splay strome pre odletené lietadla, z ktorého sa potom preradí do vyššie uvedených Splay stromov na letisku. Vyhľadávanie v Splay strome má zložitosť O(log(n))

**Zložitosť: 3 \* log(n)**

## 2. Vloženie informácie o požiadavke na pridelenie odletovej dráhy (lietadlo sa zaradí́ medzi čakajúce lietadlá na odletovú dráhu) pre lietadlo (lietadlo je identifikované svojim kódom).

Lietadlo sa vyhľadá v strome čakajúcich lietadiel na odletovú dráhu O(log(n)), nájde sa typ dráhy, ktorú požaduje O(log(n)), odstráni sa zo stromu čakajúcich lietadiel na dráhu a ak je nejaká dráha požadovanej dĺžky voľná ihneď sa k nej priradí, prehľadávanie má zložitosť n, pretože ide o array list v type dráh. Ak sa voľná dráha nenachádza lietadlo sa vloží do stromu čakajúcich lietadiel na konkrétnu dráhu O(log(n)) a aj do párovacej haldy čakajúcich lietadiel O(1)

**Zložitosť: 3 log(n) + n + 1 /alebo/ log(n) + n**

## 3. Vyhľadanie lietadla čakajúceho na pridelenie odletovej dráhy podľa medzinárodného kódu.

Vyhľadanie lietadla zo stromu čakajúcich lietadiel na pridelenie dráhy.

**Zložitosť: log(n)**

## 4. Vyhľadanie lietadla čakajúceho na pridelenie odletovej dráhy podľa medzinárodného kódu na odletovej dráhe (užívateľ zadá konkrétnu dráhu).

Vyhľadanie konkrétnej dráhy O(log(n)), zistenie pripraveného lietadla a zobrazenie aj všetkých lietadiel, ktoré potencionálne môžu mať priradenú odletovú dráhu pomocou prehliadky „level order” so zložitosťou O(2n)

**Zložitosť: log(n) + 2n**

## 5. Vloženie informácií o odlete lietadla (lietadlo je identifikované svojim kódom) a uvoľnení dráhy – ak nejaké čakajúce lietadlo môže dráhu využiť je mu ihneď pridelená a systém zobrazí informácie o takomto lietadle.

Vyhľadanie lietadla na letisku O(log(n)), zistenie či je pripravené na štart, ak áno, lietadlo odletí do stromu „airplanesInAir“, odstránenie a pridanie do nového stromu má zložitosť O(3log(n)). Následne sa popne z párovacej haldy ďalšie lietadlo a priradí na štart danej dráhy a vymaže sa aj zo stromu čakajúcich lietadiel na odlet O(log(n)). Odletenému lietadlu sa vytvorí odlet do histórie na danej dráhe.

**Zložitosť: 5 \* log(n)**

## 6. Výpis všetkých lietadiel čakajúcich na pridelenie odletovej dráhy usporiadaných podľa ich kódov.

Výpis lietadiel pomocou prehliadky „inOrder“.

**Zložitosť: 2 \* log(n)**

## 7. Výpis všetkých lietadiel čakajúcich na pridelenie konkrétnej odletovej dráhy usporiadaných podľa ich kódov.

Vyhľadanie konkrétnej dráhy, zistenie jej typu a výpis všetkých čakajúcich lietadiel prehliadkov „inOrder“

**Zložitosť: 3 \* log(n)**

## 8. Zmena priority lietadla čakajúceho na pridelenie odletovej dráhy (lietadlo je identifikované svojim kódom)

Vyhľadanie lietadla, zistenie či má pridelený typ dráhy kde čaká a zmena priority lietadla s preusporiadaním párovacej haldy.

**Zložitosť: log(n) /alebo/ 2 \* log(n)**

## 9. Výpis všetkých odletových dráh, pričom pre každú dráhu sa vypíšu všetky realizované odlety so všetkými dostupnými informáciami usporiadané podľa poradia v akom sa realizovali.

Prehliadka všetkých dráh so zložitosťou O(2n) a výpis histórie pre každú dráhu O(n).

**Zložitosť: 2n^2**

## 10. Vyradenie lietadla(lietadlo je identifikované svojim kódom) zo zoznamu čakajúcich lietadiel (zrušenie požiadavky na pridelenie odletovej dráhy)

Referencie na lietadlo sú vymazané so stromu čakajúcich lietadiel na konkrétnu dráhu O(log(n)) a paŕovacej haldy O(log(n)) a vložené do stromu čakajúcich lietadiel na odlet O(log(n)). Ak lietadlo už čaká na štart tak sa vykonajú aj operácie priradenia nového lietadla na štart ako v úlohe č.5. O(5log(n))

**Zložitosť: 3 \* log(n) /alebo/ 8 \* log(n)**