Софийски университет „Св. Климент Охридски“

Факултет по математика и информатика

Система за управление на БД със самолети за авиобаза

Автор: Даниел Здравков Стоянов

Специалност: Информатика

ФН:45574

Група: 1

Курс: 2

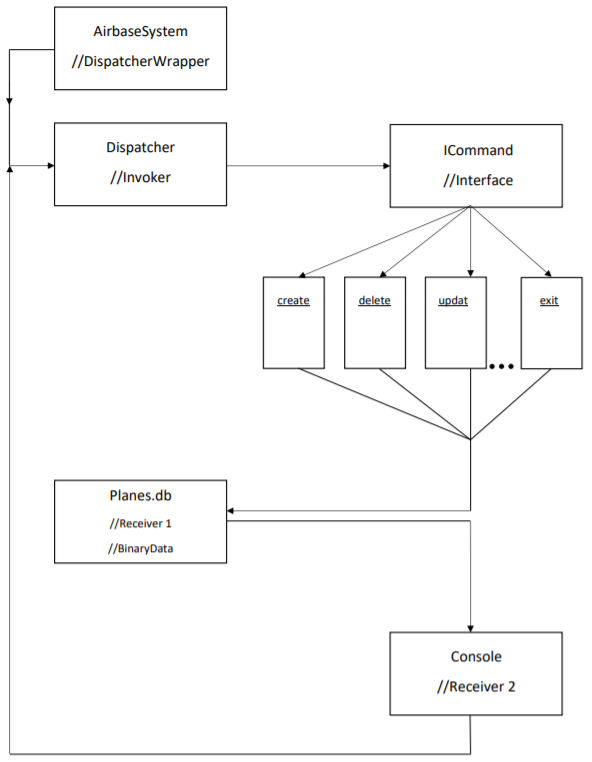
Курс Структури от данни и програмиране

за специалност Информатика  
зимен семестър *2019/2020г.*

1. **Увод:**
   1. Описание и идея на проекта:

Програмата представлява система за управление на БД със самолети за авиобаза, като предоставя възможност за следните основни действия: създаване, изтриване, редактиране, преглед и търсене по id (identifier - идентификатор) (два вида) на записи с брой осъществени полети на самолети. Информацията за всеки самолет се съхранява в постоянната памет във вид на запис, като всеки запис е достъпен по всяко време.

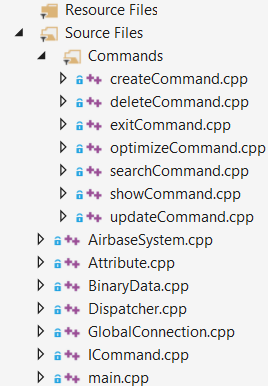
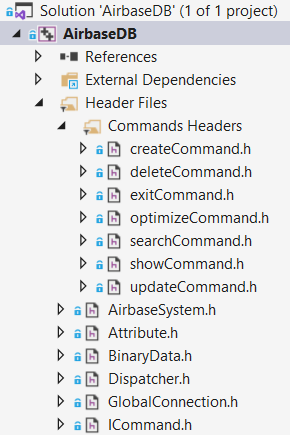
1. **Проектиране:**
   1. Обща архитектура:
      * За архитектурата на проекта е използван така наречения команден модел – Фиг. 1



Фиг. 1

* + - Основни класове са: *Dispatcher*(диспечер), *ICommand*(интерфейс за команди), *createCommand*(команда „създай“ и останалите 6 команди), *BinaryData*(запис на информация) без които функционалността на проекта не би била възможна.
    - Наследяване е използвано само при интерфейса на командите. Добавени са и помощни класове като AirbaseSystem(обвивка на диспечер), Attribute(тип за обработване на записи на самолети), GlobalConnection с цел улеснение и по-добра реализация.
    - Функционалността започва от AirbaseSystem(обвивка на диспечер), където се въвежда команда от потребителя, тя се обработва от диспечера, който разпознава коя е командата и я изпълнява. Командите форматират записите на двоичен файл на име „Planes.db“, като промените се запазват и след затваряне на програмата.

1. **Реализация:**
   1. Реализиране на класовете:
      * Използвано е разделно компилиране за по-добра огранизация и четимост – Фиг. 2

****

Фиг. 2

* + - В класовете е реализирано канонично представяне там, където е необходимо.
    - Също така са реализирани са и помощни функции за оптимизация и по-добра четимост.
    - Осъществена е връзката между отделните класове за успешното изпълнение на поставените задачи.
    - За командата *optimize* е използван алгоритъма *Merge sort* чрез който се постига сортиране за O(n.log(n)), а сложността за търсене е реализирана чрез двоично търсене за O(logn).

1. **Заключение:**
   1. Крайният продукт е работещо приложение, което отговаря на критериите на поставената задача.
   2. Възможно е бъдеще разширение на проекта, като може да се добавят лесно нови команди и функционалности.
   3. **Референция към git хранилище:** [**https://github.com/DannyStoyanov/AirbaseDB.git**](https://github.com/DannyStoyanov/AirbaseDB.git)
   4. Използвана литература:
      * <http://www.cplusplus.com/>
      * <https://en.cppreference.com/w/>
      * <https://www.geeksforgeeks.org/>
      * <https://www.youtube.com/>
      * <https://www.wikipedia.org/>