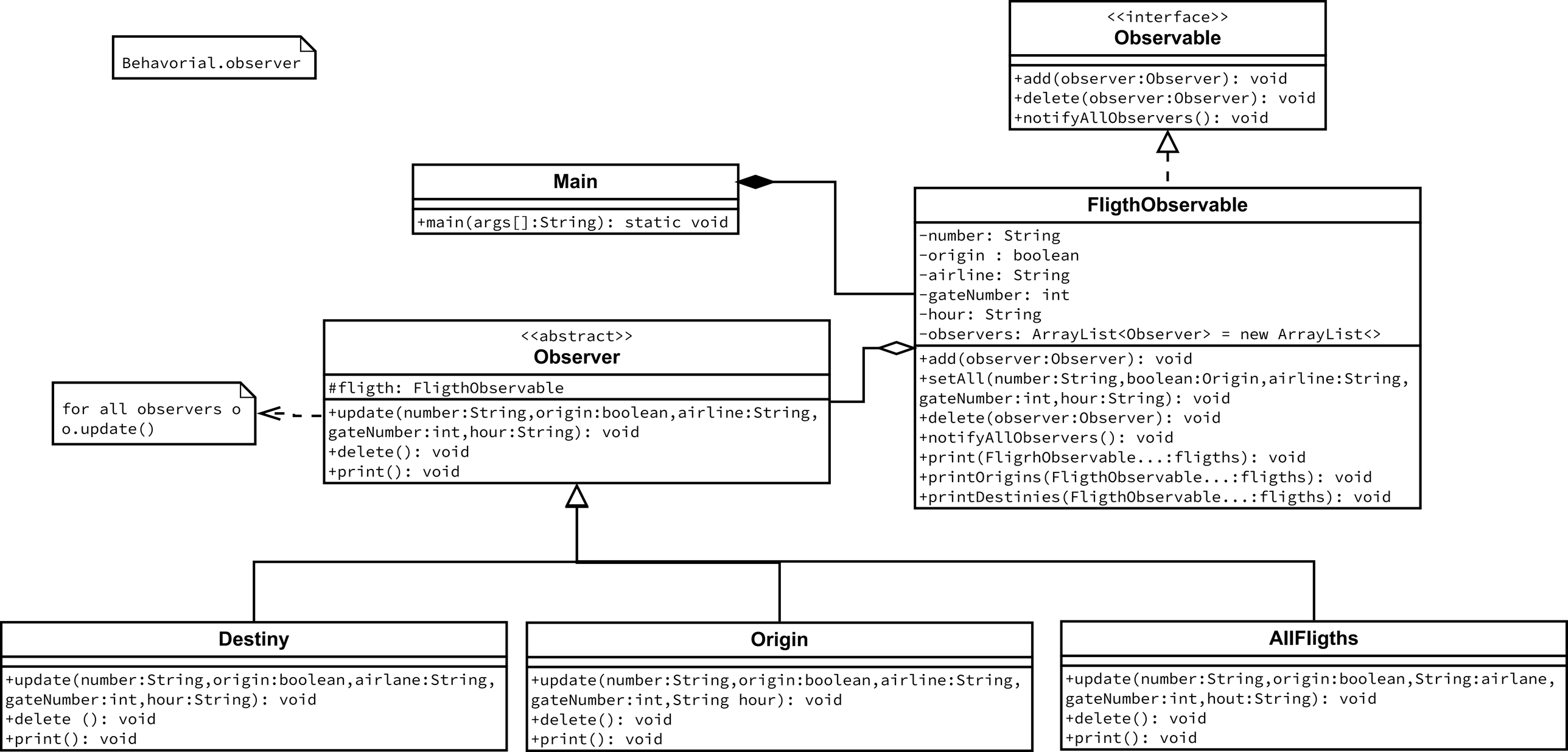
Behavorial.observer
<<interface>>
Observable
+add(observer:Observer): void
+delete(observer:Observer): void
+notifyAllObservers(): void
Main
+main(args[]:String): static void
FligthObservable
-number: String
-origin : boolean
-airline: String
-gateNumber: int
-hour: String
-observers: ArrayList<Observer> = new ArrayList<>
+add(observer:Observer): void
+setAll(number:String,boolean:Origin,airline:String,
gateNumber:int,hour:String): void
+delete(observer:Observer): void
+notifyAllObservers(): void
+print(FligrhObservable...:fligths): void
+printOrigins(FligthObservable...:fligths): void
+printDestinies(FligthObservable...:fligths): void
for all observers o
o.update()
<<abstract>>
Observer
#fligth: FligthObservable
+update(number:String,origin:boolean,airline:String,
gateNumber:int,hour:String): void
+delete(): void
+print(): void
Destiny
+update(number:String,origin:boolean,airlane:String,
gateNumber:int,hour:String): void
+delete (): void
+print(): void
Origin
+update(number:String,origin:boolean,airline:String,
gateNumber:int,String hour): void
+delete(): void
+print(): void
AllFligths
+update(number:String,origin:boolean,String:airlane,
gateNumber:int,hout:String): void
+delete(): void
+print(): void