UML Comments

**Class *Case* ():**

(Getters and setters)

*toString():* the abstract method used to display all the info after the analysis of the file is complete.

**Class *CriminalCase ():***

(Getters and setters)

*+toString():* (abstract class)

**Class CrimeAgainstPerson ():**

+CrimeAgainstPerson (severe: String, typOff: String): constructor that will initialize the fields of the class

(Getters and setters)

**Class DrugCrime ():**

+DrugCrime(pos: boolean, traff: boolean, amt: double, typeOfDrug: String): constructor that initializes the fields of the class plus the parent classes

(Getters and setters)

**Class MoneyCrimes ():**

+MoneyCrimes(amtSt: double, viol: boolean, typeProp: String): constructor that will initialize the fields of the class plus the parent classes

(getters and setters)

**Class *HumanRights ()*:**

(Getters and setters)

+*toString*(): String

**Class Charter():**

+Charter(right: String): constructor that initialize the fields of the class plus parent classes

(Getters and setters)

**Class OHRC():**

+OHRC(gr: String, ar: String, typeDis: String): constructor that initialize the fields of the class plus parent classes

(Getters and setters)

**Class *Civil():***

(Getters and setters)

*+toString*(): String

**Class Family():**

+ Family(mar: Boolean, div: Boolean, child: Boolean): Family: constructor that initialize the fields of the class plus parent classes

**Class Contract():**

+ Contract(pur: String, hasCon: Boolean, parCon: Boolean): Contract: constructor that initialize the fields of the class plus parent classes

**Class willEstate():**

**+**willEstate(ast: Boolean, dec: Boolean, hW: Boolean): willEstate: constructor that initialize the fields of the class plus parent classes

**Class caseChecker():**

+setCase(complaint: file): Case: implement the case type from the text file (for now)

+setCaseType (): void: finds the keywords from the file and according to that, it sets the case type

**Class Account () => aggregation with caseChecker() and Case():**

+addCase (): Boolean: when a case is recognized it will be added to the array of different types of cases, and if adding a case was successful then the method will return true, else it will return false

+withdrawCase (): Boolean: method used to withdraw cases, if the withdrawal is successful then the method will return true, else it will return a false

+assignLawyer (): void: according to the cases the client is dealing with, this method will assign a lawyer to the client

+setAppointment(availability: String): String: will return an appointment made with a lawyer depending on the lawyers availability, plus the clients availability

**Class Lawyer () => aggregation with Account():**

+Lawyer (a: String, s: String, n: String, pn: String): sets the fields of the class

+appointCase(userAvailability: String): void: according to availability the method will appoint that certain date

(Getters and setters)

**Class Person () => composition with Account():**

+Person (n: String, c: String, pn: String): sets the fields of the class

(Getters and setters)