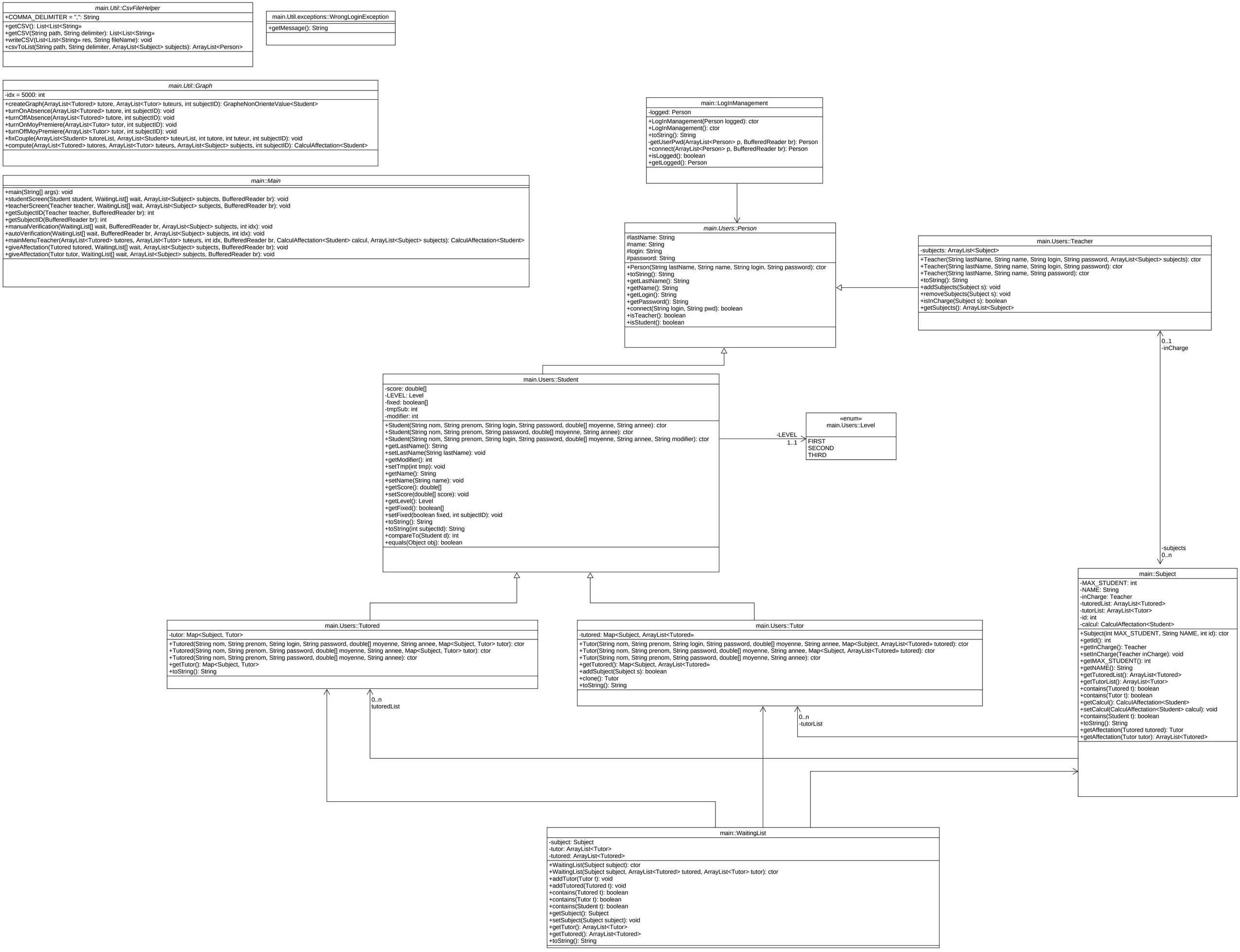
main.Util::CsvFileHelper
+COMMA_DELIMITER = ",": String
+getCSV(): List<List<String»
+getCSV(String path, String delimiter): List<List<String»
+writeCSV(List<List<String» res, String fileName): void
+csvToList(String path, String delimiter, ArrayList<Subject> subjects): ArrayList<Person>
main.Util.exceptions::WrongLoginException
+getMessage(): String
main.Util::Graph
-idx = 5000: int
+createGraph(ArrayList<Tutored> tutore, ArrayList<Tutor> tuteurs, int subjectID): GrapheNonOrienteValue<Student>
+turnOnAbsence(ArrayList<Tutored> tutore, int subjectID): void
+turnOffAbsence(ArrayList<Tutored> tutore, int subjectID): void
+turnOnMoyPremiere(ArrayList<Tutor> tutor, int subjectID): void
+turnOffMoyPremiere(ArrayList<Tutor> tutor, int subjectID): void
+fixCouple(ArrayList<Student> tutoreList, ArrayList<Student> tuteurList, int tutore, int tuteur, int subjectID): void
+compute(ArrayList<Tutored> tutores, ArrayList<Tutor> tuteurs, ArrayList<Subject> subjects, int subjectID): CalculAffectation<Student>
main::Main
+main(String[] args): void
+studentScreen(Student student, WaitingList[] wait, ArrayList<Subject> subjects, BufferedReader br): void
+teacherScreen(Teacher teacher, WaitingList[] wait, ArrayList<Subject> subjects, BufferedReader br): void
+getSubjectID(Teacher teacher, BufferedReader br): int
+getSubjectID(BufferedReader br): int
+manualVerification(WaitingList[] wait, BufferedReader br, ArrayList<Subject> subjects, int idx): void
+autoVerification(WaitingList[] wait, BufferedReader br, ArrayList<Subject> subjects, int idx): void
+mainMenuTeacher(ArrayList<Tutored> tutores, ArrayList<Tutor> tuteurs, int idx, BufferedReader br, CalculAffectation<Student> calcul, ArrayList<Subject> subjects): CalculAffectation<Student>
+giveAffectation(Tutored tutored, WaitingList[] wait, ArrayList<Subject> subjects, BufferedReader br): void
+giveAffectation(Tutor tutor, WaitingList[] wait, ArrayList<Subject> subjects, BufferedReader br): void
main::LogInManagement
-logged: Person
+LogInManagement(Person logged): ctor
+LogInManagement(): ctor
+toString(): String
-getUserPwd(ArrayList<Person> p, BufferedReader br): Person
+connect(ArrayList<Person> p, BufferedReader br): Person
+isLogged(): boolean
+getLogged(): Person
main.Users::Person
#lastName: String
#name: String
#login: String
#password: String
+Person(String lastName, String name, String login, String password): ctor
+toString(): String
+getLastName(): String
+getName(): String
+getLogin(): String
+getPassword(): String
+connect(String login, String pwd): boolean
+isTeacher(): boolean
+isStudent(): boolean
main.Users::Teacher
-subjects: ArrayList<Subject>
+Teacher(String lastName, String name, String login, String password, ArrayList<Subject> subjects): ctor
+Teacher(String lastName, String name, String login, String password): ctor
+Teacher(String lastName, String name, String password): ctor
+toString(): String
+addSubjects(Subject s): void
+removeSubjects(Subject s): void
+isInCharge(Subject s): boolean
+getSubjects(): ArrayList<Subject>
0..1
-inCharge
-subjects
0..n
main.Users::Student
-score: double[]
-LEVEL: Level
-fixed: boolean[]
-tmpSub: int
-modifier: int
+Student(String nom, String prenom, String login, String password, double[] moyenne, String annee): ctor
+Student(String nom, String prenom, String password, double[] moyenne, String annee): ctor
+Student(String nom, String prenom, String login, String password, double[] moyenne, String annee, String modifier): ctor
+getLastName(): String
+setLastName(String lastName): void
+getModifier(): int
+setTmp(int tmp): void
+getName(): String
+setName(String name): void
+getScore(): double[]
+setScore(double[] score): void
+getLevel(): Level
+getFixed(): boolean[]
+setFixed(boolean fixed, int subjectID): void
+toString(): String
+toString(int subjectId): String
+compareTo(Student d): int
+equals(Object obj): boolean
-LEVEL
1..1
«enum»
main.Users::Level
FIRST
SECOND
THIRD
main::Subject
-MAX_STUDENT: int
-NAME: String
-inCharge: Teacher
-tutoredList: ArrayList<Tutored>
-tutorList: ArrayList<Tutor>
-id: int
-calcul: CalculAffectation<Student>
+Subject(int MAX_STUDENT, String NAME, int id): ctor
+getId(): int
+getInCharge(): Teacher
+setInCharge(Teacher inCharge): void
+getMAX_STUDENT(): int
+getNAME(): String
+getTutoredList(): ArrayList<Tutored>
+getTutorList(): ArrayList<Tutor>
+contains(Tutored t): boolean
+contains(Tutor t): boolean
+getCalcul(): CalculAffectation<Student>
+setCalcul(CalculAffectation<Student> calcul): void
+contains(Student t): boolean
+toString(): String
+getAffectation(Tutored tutored): Tutor
+getAffectation(Tutor tutor): ArrayList<Tutored>
main.Users::Tutored
-tutor: Map<Subject, Tutor>
+Tutored(String nom, String prenom, String login, String password, double[] moyenne, String annee, Map<Subject, Tutor> tutor): ctor
+Tutored(String nom, String prenom, String password, double[] moyenne, String annee, Map<Subject, Tutor> tutor): ctor
+Tutored(String nom, String prenom, String password, double[] moyenne, String annee): ctor
+getTutor(): Map<Subject, Tutor>
+toString(): String
0..n
tutoredList
main.Users::Tutor
-tutored: Map<Subject, ArrayList<Tutored»
+Tutor(String nom, String prenom, String login, String password, double[] moyenne, String annee, Map<Subject, ArrayList<Tutored» tutored): ctor
+Tutor(String nom, String prenom, String password, double[] moyenne, String annee, Map<Subject, ArrayList<Tutored» tutored): ctor
+Tutor(String nom, String prenom, String password, double[] moyenne, String annee): ctor
+getTutored(): Map<Subject, ArrayList<Tutored»
+addSubject(Subject s): boolean
+clone(): Tutor
+toString(): String
0..n
-tutorList
main::WaitingList
-subject: Subject
-tutor: ArrayList<Tutor>
-tutored: ArrayList<Tutored>
+WaitingList(Subject subject): ctor
+WaitingList(Subject subject, ArrayList<Tutored> tutored, ArrayList<Tutor> tutor): ctor
+addTutor(Tutor t): void
+addTutored(Tutored t): void
+contains(Tutored t): boolean
+contains(Tutor t): boolean
+contains(Student t): boolean
+getSubject(): Subject
+setSubject(Subject subject): void
+getTutor(): ArrayList<Tutor>
+getTutored(): ArrayList<Tutored>
+toString(): String