COMP90041 Programming and Software Development
Final Project
UML Class Diagram

Ebru Soezbir 1135390 nsoezbir | June 25, 2020

package ethicalengine

<<enumeration>> Gender

FEMALE
MALE
UNKNOWN

<<enumeration>> BodyType

AVERAGE
ATHLETIC
OVERWEIGHT
UNSPECIFIED

Character

- age: int
- gender: Gender
- bodyType: BodyType
...

+ Character()
+ Character(age: int, gender: Gender, bodyType: BodyType)
+ Character(otherCharacter: Character)
+ getAge(): int
+ getGender(): Gender
+ getBodyType(): BodyType
+ setAge(age: int): void
+ setGender(gender: Gender): void
+ setBodyType(bodyType: BodyType): void

Person

- isPregnant: boolean
- isYou: boolean
- profession: Profession
- ageCategory: AgeCategory
...

+ Person()
+ Person(age: int, gender: Gender, bodyType: BodyType)
+ Person(age: int, profession: Profession, gender: Gender, bodyType: BodyType, isPregnant: boolean)
+ Person(otherPerson: Person)
+ isPregnant(): boolean
+ setPregnant(pregnant: boolean): void
+ isYou(): boolean
+ setAsYou(isYou: boolean): void
+ getAgeCategory(age: int): AgeCategory
+ getAgeCategory(): AgeCategory
+ getProfession(): Profession
+ toString(): String

<<enumeration>> Profession

DOCTOR
CEO
CRIMINAL
HOMELESS
UNEMPLOYED
TEACHER
STUDENT
UNKNOWN
NONE
PROFESSOR

<<enumeration>> AgeCategory

BABY
CHILD
ADULT
SENIOR

Animal

- species: String
- isPet: boolean
...

+ Animal()
+ Animal(species: String)
+ Animal(species: String, isPet: boolean)
+ Animal(age: int, gender: Gender, bodyType: BodyType, species: String, isPet: boolean)
+ Animal(otherAnimal: Animal)
+ getSpecies(): String
+ setSpecies(species: String): void
+ isPet(): boolean
+ setPet(isPet: boolean): void
+ toString(): String

Scenario

- passengers: ArrayList<Character>
- pedestrians: ArrayList<Character>
- isLegalCrossing: boolean

+ hasYouInCar(): boolean
+ hasYouInLane(): boolean
+ getPassengers(): Character[]
+ getPedestrians(): Character[]
+ isLegalCrossing(): boolean
+ setLegalCrossing(isLegalCrossing: boolean): void
+ getPassengerCount(): int
+ getPedestrianCount(): int
+ toString(): String

ScenarioGenerator

- random: Random
- passengerCountMinimum: int
- passengerCountMaximum: int
- pedestrianCountMinimum: int
- pedestrianCountMaximum: int
...

+ ScenarioGenerator()
+ ScenarioGenerator(seed: long)
+ ScenarioGenerator(seed: long, passengerCountMinimum: int, passengerCountMaximum: int, pedestrianCountMinimum: int, pedestrianCountMaximum: int)
+ setPassengerCountMin(min: int): void
+ setPassengerCountMax(max: int): void
+ setPedestrianCountMin(min: int): void
+ setPedestrianCountMax(max: int): void
+ getRandomPerson(): Person
+ getRandomAnimal(): Animal
+ generate(): Scenario

MyPair

- name: String
- value: double

+ MyPair(name: String, value: double)
+ getName(): String
+ getValue(): double
+ compareTo(otherPair: MyPair)

<<enumeration>> Decision

PEDESTRIANS
PASSANGERS

EthicalEngine

...

+ decide(scenario: Scenario): Decision
+ extractPerson(dataRow: String[], lineCounter: int): Person
+ extractAnimal(dataRow: String[], lineCounter: int): Animal
+ readConfigFile(filepath: String): ArrayList<Scenario>
...

Audit

- auditType: String
- runs: int
- scenarioList: ArrayList<Scenario>
- survivorsReport: HashMap<String, Integer>
- deathsReport: HashMap<String, Integer>
...

+ Audit()
+ Audit(scenarioList: ArrayList<Scenario>)
+ setAuditType(auditTypeName: String): void
+ getAuditType(): String
+ run(newRuns: int): void
+ run(): void
+ toString(): String
+ printStatistic(): void
+ printToFile(filepath: String): void

contains

contains

contains

generates

generates

runs

decides

0..*

0..*

0..*

0..*

1..*

1..*

1..*

1..*