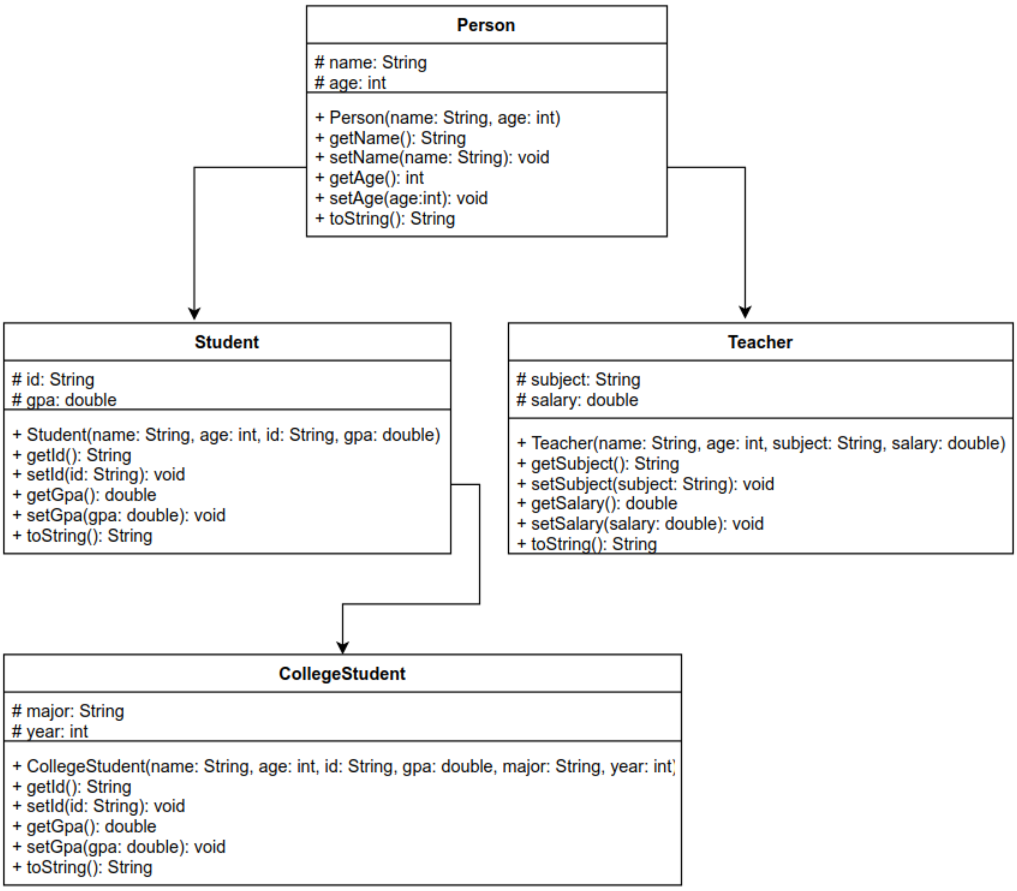
CSC 230: Elementary Data Structures and Algorithms

Fall 2022

Assignment 7

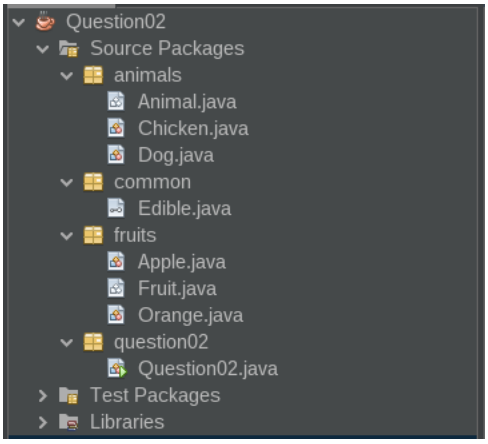
- General programming guidelines:
  - Create a separate NetBeans project with the name `QuestionXX`, where `XX` is the question number.
  - Do not forget necessary javadoc comments before classes and methods.
  - You should use *single line* or *multiline* comments, **if it is required**. Do not put unnecessary comments (Do not state the obvious!!!).
  - Use meaningful identifier.
  - Don't forget to check the parameters of your methods and throw appropriate exceptions as necessary.
  - **Creating correct NetBeans projects, zipping your final assignment folder, and testing it before uploading are your responsibilities. If any of these steps fails, you will receive a grade of zero.**
- Academic integrity policy
  - You are **not** allowed to use any online resources EXCEPT Revel, class lecture notes and Java documentation.
  - All programs/ code must be your own work.
  - You should be able to clearly explain every line of your code, if instructor requests you to do so.
  - Any violation of these policies will be considered as plagiarism and dealt accordingly.

Question 1 (50 points) In this problem we represent people in a school using inheritance. We have `Person` super class and two sub classes derived from it: `Student` and `Teacher`. A `Teacher` is a `Person` but will have additional properties such as salary (the amount the teacher earns) and subject (e.g."Computer Science", "Chemistry", "English", etc…). The `CollegeStudent` class will extend the `Student` class by adding a year (current level in college) and major (e.g. "Electrical Engineering", "Communications", "Undeclared". etc…). Other fields in the following UML diagram are self explanatory.

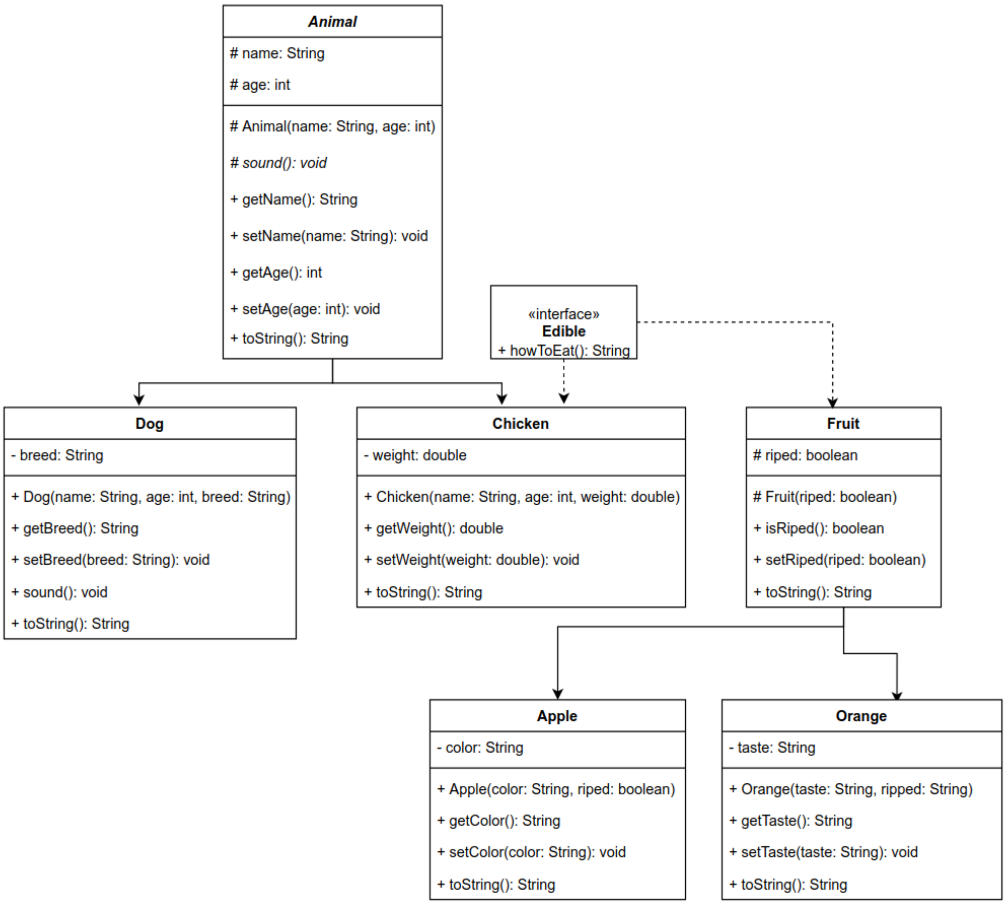

Person
# name: String
# age: int
+ Person(name: String, age: int)
+ getName(): String
+ setName(name: String): void
+ getAge(): int
+ setAge(age:int): void
+ toString(): String
Student
# id: String
# gpa: double
+ Student(name: String, age: int, id: String, gpa: double)
+ getId(): String
+ setId(id: String): void
+ getGpa(): double
+ setGpa(gpa: double): void
+ toString(): String
Teacher
# subject: String
# salary: double
+ Teacher(name: String, age: int, subject: String, salary: double)
+ getSubject(): String
+ setSubject(subject: String): void
+ getSalary(): double
+ setSalary(salary: double): void
+ toString(): String
CollegeStudent
# major: String
# year: int
+ CollegeStudent(name: String, age: int, id: String, gpa: double, major: String, year: int)
+ getId(): String
+ setId(id: String): void
+ getGpa(): double
+ setGpa(gpa: double): void
+ toString(): String

Question 2 (50 points) Start by creating a NetBeans project with the name "Question02". Create classes/ interfaces for the following UML diagram. You should use the given content of the `main` method and it's output as a guidance.

- You need to have the following package structure (Take a look at the image shown):
  - The `animals` package: `Animal`, `Chicken` and `Dog` classes.
  - The `common` package: `Edible` interface.
  - The `fruits` package: `Fruit`, `Apple` and `Orange` classes.



Question02
Source Packages
animals
Animal.java
Chicken.java
Dog.java
common
Edible.java
fruits
Apple.java
Fruit.java
Orange.java
question02
Question02.java
Test Packages
Libraries



Sample content for the `main` method:

```
Dog lily = new Dog("Shih Tzu", "Lily", 3);

System.out.println(lily.toString());
System.out.println("The dog’s name: " + lily.getName());
System.out.println("The dog’s age: " + lily.getAge());
System.out.println();

Chicken chicken = new Chicken(2, "No Name", 4);

System.out.println(chicken.toString());
System.out.println("The chicken’s name: " + chicken.getName());
System.out.println("The chicken’s age: " + chicken.getAge());
System.out.println("How to eat it? " + chicken.howToEat());
System.out.println();

Orange orange = new Orange("Sweet", true);

System.out.println(orange.toString());
System.out.println("The taste of the orange: " + orange.getTaste());
System.out.println("How to eat it? " + orange.howToEat());
System.out.println();


Apple apple = new Apple("red", true);
```

```
System.out.println(apple.toString());
System.out.println("The color of the apple: " + apple.getColor());
System.out.println("How to eat? " + apple.howToEat());
```

Here is the expected output:

```
Dog{name=Lily, age=3, breed=Shih Tzu}
The dog’s name: Lily
The dog’s age: 3

Chicken{name= No Name, age=4, weight=2.0}
The chicken’s name: No Name
The chicken’s age: 4
How to eat it? Fried or curry :-)!!!

Orange{riped=true, taste=Sweet}
The taste of the orange: Sweet
How to eat it? Make sure to peel before you eat!!!

Apple{riped=true, color=red}
The color of the apple: red
How to eat? You can eat raw!!
```