Hung-Kai Sun
ESOF
HW1
9/12/2019

ESOF 322: Software Engineering I **DUE Date: September 12, 2019**

Instructions:

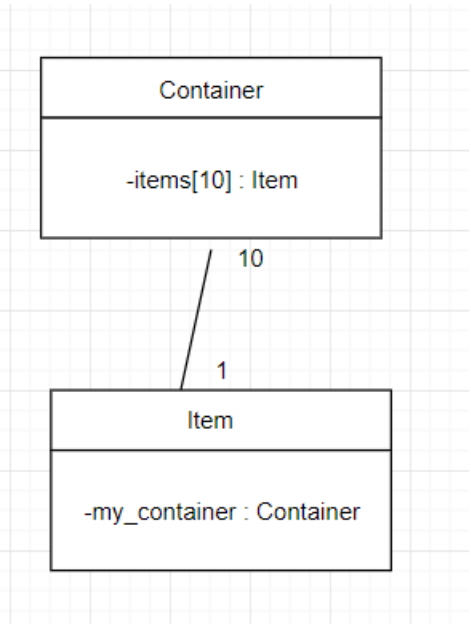
- Do all exercises with your partner.
- Clearly print the names of **all participants** in the first page of your assignment.
- No hand-written answers allowed.
- Absolutely no late assignments.
- Your assignment should be turned in to D2L by **all students in the team**

Exercises Part A (15 pts)

For each of the following (pseudo) code snippets provide the UML **class** diagram.

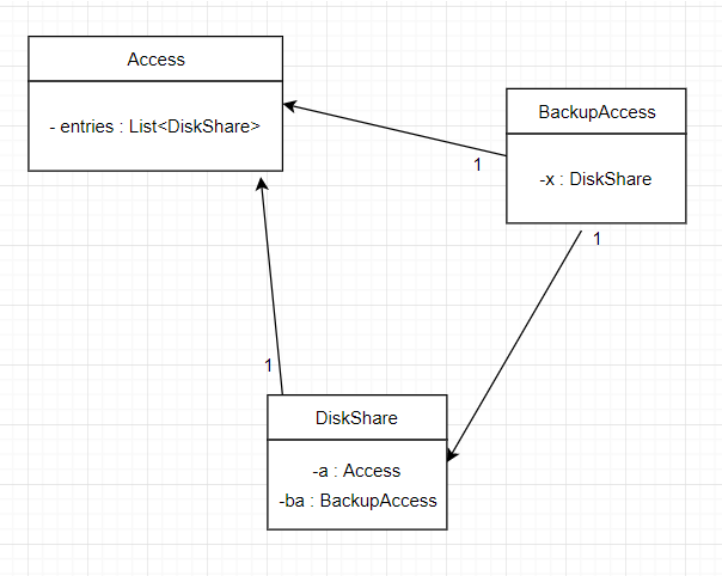
1. (**2pts**)

```
public class Container {
private Item [10] items;
}
public class Item {
private Container my_container;
}
```



For each of the following (pseudo) code snippets provide the UML **class** diagram.

2. **public** class Access { (**3pts**)
private List<DiskShare> entries;
}
public class BackupAccess {
private DiskShare x;
}
public class DiskShare {
private Access a;
private BackupAccess ba;
}

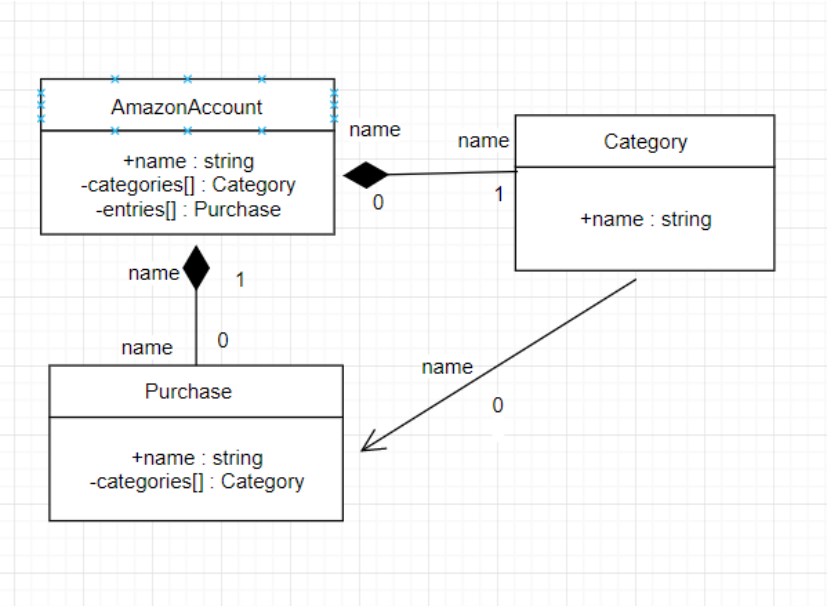


3. **public** class AmazonAccount{ **(5pts)**
private string name;

private Category[] categories;

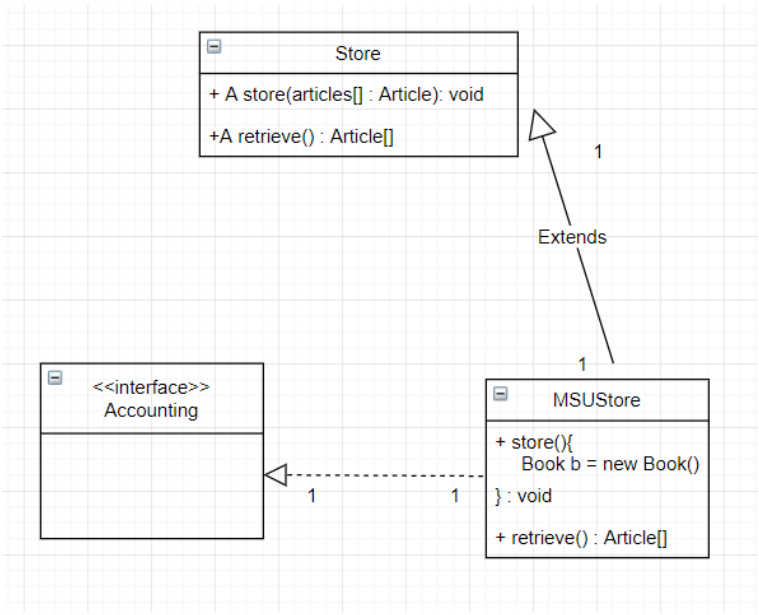
private Purchase[] entries;

}
public class Category {
private string name;
}
public class Purchase{
private string name;
private Category[] categories;
}



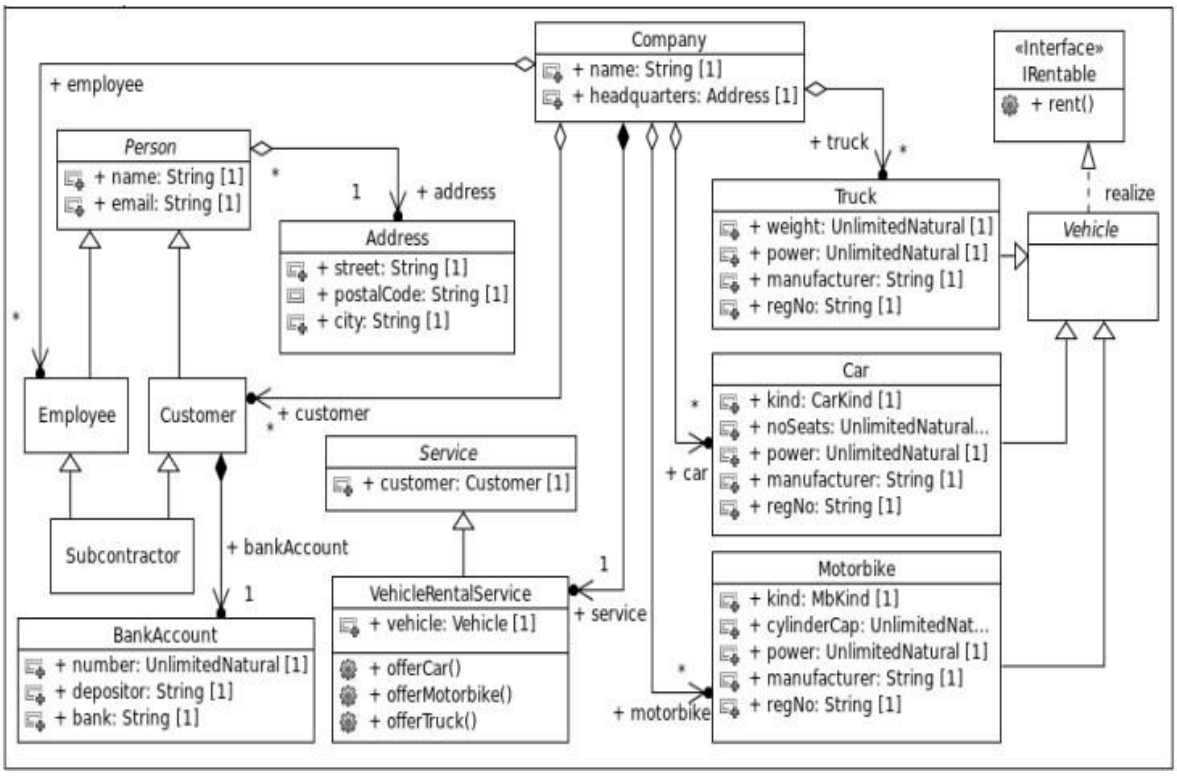
4. **public** abstract class Store{ (**5pts**)

```
    public abstract void store(Article[] articles);
    public abstract Article[] retrieve();
}
public interface Accounting {
    …
}
public class MSUStore extends Store implements Accounting{
    public void store(Article[] articles) { Book b = new Book(); // other code …. }
    public Article[] retrieve() { … }
}
```



Exerciss Part B (15 pts)

Write pseudo code to describe the following UML class diagram:




```
public class Person{
    BankAccount bankaccount;
    String name;
    String email;
    Address address;
    Person(String name, String email, Address address, BankAccount bankaccount){
        this.bankaccount = bankaccount;
        this.name = name;
        this.email = email
        this.address = address
    }
}
public class BankAccount{
```

```
        UnlimitedNatural number;
        String depositor;
        String bank;
        BankAccount(UnlimitedNatural number, String depositor, String bank){
                this.number = number;
                this.depositor = depositor;
                this.bank = bank;
        }
}
public class Address{
        String street;
        String postalCode;
        String city;
        Address(String street, String postalCode, String city){
                this. street = street;
                this.postalCode = postalCode;
                this.city = city;
        }
}
public class Company{
        String name;
        Customer customer
        Address headquarters;
        Company(Customer customer, String name, Address headquarters){
                this.customer = customer;
                this.name = name;
                this.headquarters = headquarters;
        }
}
```

```
interface Rentable{
	public rent();
}
class Vehicle implements Rentable{
	public
	 rent(){
}
class Car extends Vehicle{
	Company company;
	UnlimitedNatural noSeats;
	kind CarKind;
	UnlimitedNatural power;
	String manufacturer ;
	String regNo;
	Car(Company company, UnlimitedNatural noSeats, kind CarKind, UnlimitedNatural power,
String manufacturer, String regNo){
		this.company = company;
		this.noSeats = noSeats;
		this.Carkind = CarKind;
		this.power = power;
		this.manufacturer = manufacturer;
		this.regNo = regNo;
	}
}
class Motorbike extends Vehicle{
	Company company;
	kind MbKind;
	UnlimitedNatural cylinderCap;
	UnlimitedNatural power;
	String manufacturer;
```

```
        String regNo;
        Motorbike(Company company, kind MbKind, UnlimitedNatural cylinderCap, UnlimitedNatural
power, String manufacturer, String regNo){
                this.company = company;
                this.Mbkind = Mbkind;
                this.cylinderCap = cylinderCap;
                this.power = power;
                this.manufacturer = manufacturer;
                this.regNo = regNo;
        }
}
abstract class Service{
        public static final Customer customer;
}
class VehicleRentalService extends Vehicle{
        Vehicle vehicle;
        public Car offerCar(){
                return this.Car;
        }
        public Motorbike offerMotorbike(){
                return this.Motorbike;
        }
        public Truck offerTruck(){
                return this.Truck;
        }
}
class Truck extends Vehicle{
        UnlimitedNatural weight;
        UnlimitedNatural power;
        String manufacturer;
```

```
        Company company;

        String regNo;

        Truck(UnlimitedNatural weight, UnlimitedNatural power, String manufacturer, Company
company, String regNo){

                this.weight = weight;

                this.power = power;

                this.manufacturer = manufacturer;

                this.company = company;

                this.regNo = regNo;

        }

}
```