Homework 2

Elliott Pryor

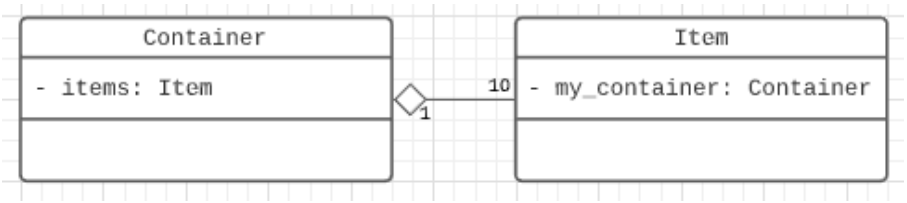
3 Sept 2020

A

Problem 1

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

(a) Psudocode for problem 1 (QUESTION)



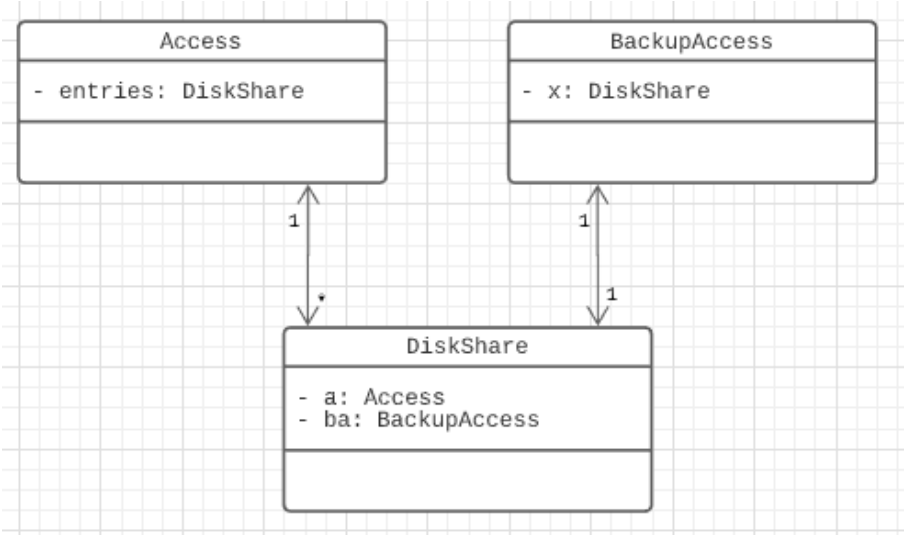
(b) UML Diagram for problem 1 (SOLUTION)

Figure 1: Problem 1 question and solution

Problem 2

```
public class Access {
        private List<DiskShare> entries;
}
public class BackupAccess {
        private DiskShare x;
}
public class DiskShare {
        private Access a;
        private BackupAccess ba;
}
```

(a) Psudocode for problem 2 (QUESTION)



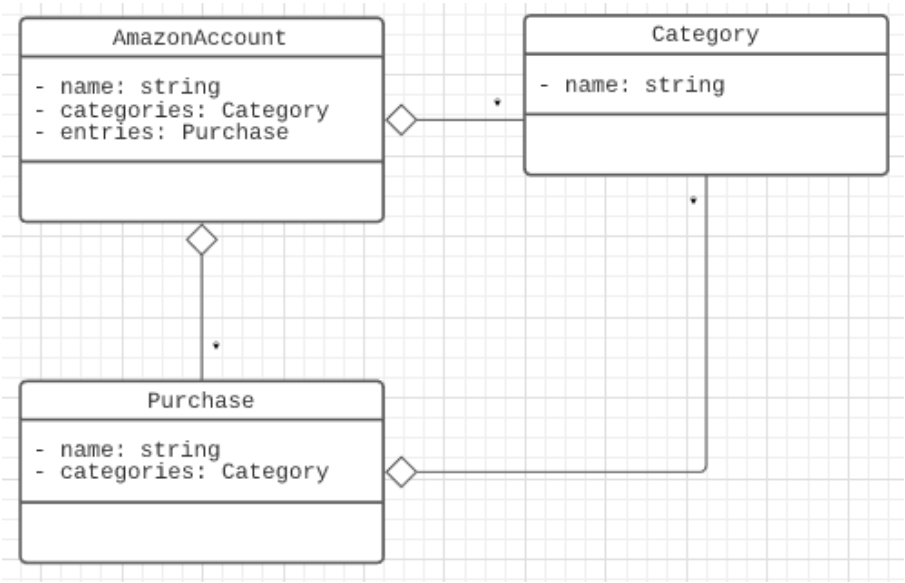
(b) UML Diagram for problem 2 (SOLUTION)

Figure 2: Problem 2 question and solution

Problem 3

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

(a) Psudocode for problem 3 (QUESTION)



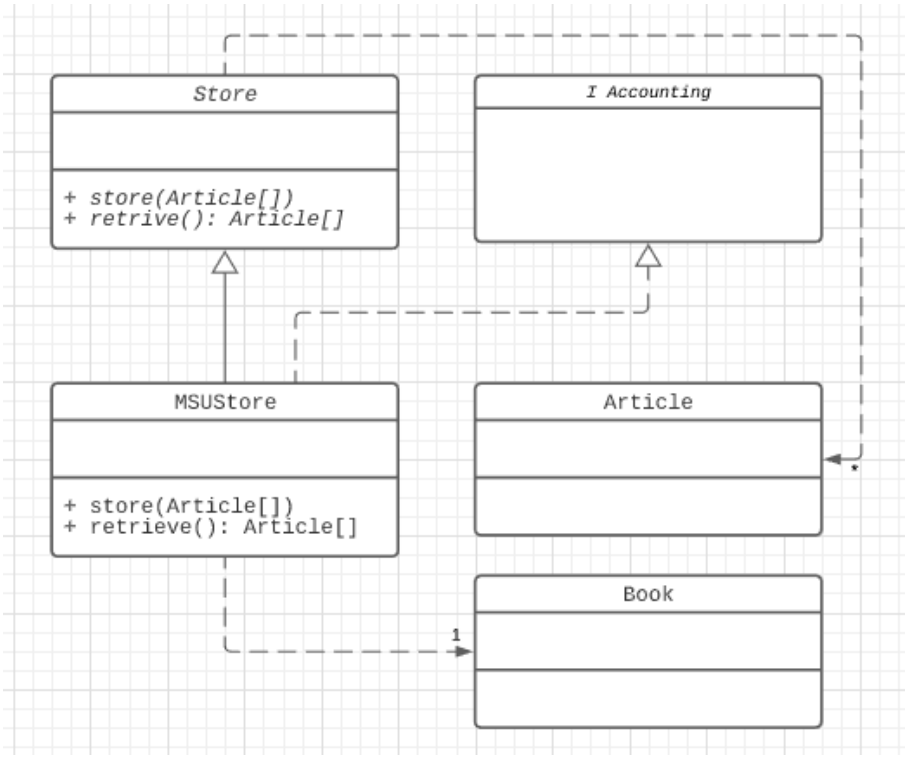
(b) UML Diagram for problem 3 (SOLUTION)

Figure 3: Problem 3 question and solution

Problem 4

```
public abstract class Store {
        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() { … }
}
```

(a) Psudocode for problem 4 (QUESTION)



(b) UML Diagram for problem 4 (SOLUTION)

Figure 4: Problem 4 question and solution

B

```
public class Company{
        public string name;
        public Address headquarters;
        public Employee[] employee;
        public Customer[] customer;
        public VehicleRentalService service;
        public Truck[] truck;
        public Car[] car;
        public Motorbike[] motorbike;
}

public abstract class Person{
        public string name;
        public string email;
        public Address address;
}

public class Employee extends Person{
}

public class Customer extends Person{
        public BankAccount bankAccount;
}

public class Subcontractor extends Employee, Person{
}

public class Address{
        public string name;
        public string postalCode;
        public string city;
}

public class BankAccount{
        public int number;
        public string depositor;
        public string bank;
}

public abstract class Service{
        public Customer customer;
```

```
}

public class VehicleRentalService extends Service{
        public Vehicle vehicle;

        public void offerCar(){}
        public void offerMotorbike(){}
        public void offerTruck(){}
}

public interface Rentable{
        public void rent();
}

public abstract class Vehicle implements Rentable{

        public abstract void rent(); // push implementation to subclasses

}

public class Truck extends Vehicle{
        public int weight;
        public int power;
        public string manufacturer;
        public string regNo;

        public void rent(){}

}

public class Car extends Vehicle{
        public CarKind kind;
        public int noSeats;
        public int power;
        public string manufacturer;
        public string regNo;

        public void rent(){}
}

public class Motorbike extends Vehicle{
        public MbKind kind;
        public int cylinderCap;
        public int power;
```

```
        public string manufacturer;
        public string regNo;

        public void rent(){}
}
```