Programming in Java

Coursework One

See the Moodle site for the due date

Introduction

Several students have commented that they found some difficulties understanding how data structures are set in memory. The following three exercises are given as additional practice and preparation for the first coursework which is the fourth exercise.

You should solve all the questions using pen and paper. The objective is to read the code, understand it, and produce diagrams that explain the state of the variables (both simple and complex types) in memory at every numbered point (as shown in the comments). The points are not numbered in any particular order, and the program may go through each point more than once.

If you still find them confusing, or just want to confirm that your understanding is correct, then you are encouraged to show your proposed solution(s) to the teaching team to receive feedback; this applies only to the first three exercises (obviously).

Submission

You should submit your solution to question 4 on paper. Pen-drawing is fine as long as it is clear; there is no need (or desire) for us to receive fancy computer-generated diagrams.

To submit, hand your diagrams to any member of the teaching team. You should write your name and degree programme on all of the pages you submit.

1. A *Car dealing* practice example:

```
class Car {
    String Model;
    double price;
}

void trade(Car newCar, Car oldCar, int myFund) {
    if (newCar.price <= oldCar.price + myFund) {
        println("we have a deal");
    } else {
        println("forget about it...");
    }

    println(" now we try to be naughty");
    Car temp = new Car();
    temp = newCar;
    newCar = oldCar;
    oldCar = temp;
    // Point 1
}

Car myOldFord = new Car();
myOldFord.Model = "Ka";
myOldFord.price = 2000;
Car fancyRacer = new Car();
fancyRacer.Model = "911";
fancyRacer.price = 300000;
int myBudget = 10000;
// Point 2


println("Lets exchange cars! Deal?");
trade(fancyRacer, myOldFord, myBudget);
// Point 3

println("The new car has become " + fancyRacer.model);
println("And the old car has become " + myOldFord.model);
println("The deal has failed");
```

2. A *Film rating* practice example:

```
class Film {
    String title;
    int reviewer = 0;
    double totalStar = 0;

    void review(int stars) {
        reviewer++;
        totalStar += stars;
    }

    double getRating() {
        if (reviewer == 0) {
            return 0;
        } else {
            return totalStar / reviewer;
        }
    }
}

void rate(Film f, int stars) {
    println("Viewer voting being processed...");
    f.review(stars);
    f = null;
    stars++;
    println("The value of star inside this method is: " + stars);
    // Point 1
}

Film helpMeChoose(Film f1, Film f2) {
    return f1.getRating() > f2.getRating() ? f1 : f2;
}

Film film1 = new Film();
film1.title = "Prometheus";
Film film2 = new Film();
film2.title = "Twilight";
print(film1.reviewer + " reviewers have given " + film1.title);
println(" an average rating of " + film1.getRating() + " stars.");
print(film2.reviewer + " reviewers have given " + film2.title);
println(" an average rating of " + film2.getRating() + " stars.");
// Point 2

int stars = 5;
```

```
rate(film1, stars);
println("The value of star here is now: " + stars);

stars = 4;
rate(film1,stars);
println("The value of star here is now: " + stars);

stars = 2;
rate(film2,stars);
println("The value of star here is now: " + stars);

stars = 3;
rate(film2,stars);
println("The value of star here is now: " + stars);
// Point 3

print(film1.reviewer + " reviewers have given " + film1.title);
println(" an average rating of " + film1.rating + " stars.");
print(film2.reviewer + " reviewers have given " + film2.title);
println(" an average rating of " + film2.rating + " stars.");
Film finalChoice = helpMeChoose(film1,film2);
println("And we officially recommend: " + finalChoice.title);
// Point 4
```

3. A practice example using *People*:

```
class Person {
    String firstname;
    String surname;

    Person father;
    Person mother;

    String personDetails() {
        String result="";
        result+="firstname: "+firstname;
        result+="\n";
        result+="surname: "+surname;
        return result;
    }

    String familyDetails() {
        String result="";
        result+="father\n";
        result+="------\n";
        result+=father.personDetails();
        result+="\n";
        result+="mother\n";
        result+="------\n";
        result+=mother.personDetails();
        return result;
    }


    void showDetails() {
        println(personDetails());
        println(familyDetails());
    }
}

Person p1=new Person();
p1.firstname="Adam";
p1.surname="Taylor";
// Point 1

Person p2=new Person();
p2.firstname="Eve";
p2.surname="Smith";
// Point 2
```

```
Person p3=new Person();
p3.firstname="Daniel";
p3.surname="Taylor";
p3.father=p1;
p3.mother=p2;
p3.showDetails();
// Point 3
```

4. The coursework assignment which is to be handed in for grading:

```
class Travelcard {
    String type;
    int zone;
}

class OysterCard {
    boolean autotopup;
    boolean registered;
    double amount;
    Travelcard travelcardAttached;
}

boolean yellowReader(OysterCard card, int zone) {
    double fare = getFare(zone);
    double cost = 0.0;
    if(card.travelcardAttached != null) {
        if(card.travelcardAttached.zone >= zone) {
            return true;
        }
    }
    if (card.amount > fare) {
        card.amount -= fare;
        return true;
    }
    return false;
}

double getFare(int zone) {
    double fare=0.0;
    if (zone <= 2) {
        fare = 2.3;
    } else if (zone <= 4) {
        fare = 4.6;
    } else {
        fare = 6.2;
    }
    return fare
}

OysterCard card1 = new OysterCard();
card1.autotopup = true;
card1.registered = false;
card1.amount = 50.0;
```

```
card1.travelcardAttached = null;

Travelcard tcard = new Travelcard();
tcard.type = "week";
tcard.zone = 2;

OysterCard card2 = new OysterCard();
card2.autotopup = false;
card2.registered = true;
card2.amount = 6.3;
card2.travelcardAttached = tcard;
// Point 1

println(yellowReader(card1, 2)? "pass" : "problem with card");
println(yellowReader(card1, 6)? "pass" : "problem with card");
println(yellowReader(card2, 4)? "pass" : "problem with card");
println(yellowReader(card2, 5)? "pass" : "problem with card");
// Point 2
```