

TollCalculator

Syfte

Syftet med den här uppgiften är att lära oss att självständigt skapa program i Objektorienterad stil. Ni ska också arbeta i grupp om 2 personer och lära er använda gitbrancher för att organisera ert arbete sinsemellan. Ni ska också skaffa färdigheter i att planera och designa mjukvara på ett mer strukturerat sätt med problem nedbrytning och planering.

Kursmål som täcks av uppgift

Kunskaper

2. Redogöra för de vanligaste biblioteken i Javaplattformen
3. Beskriva hur man analyserar och bryter ner problem till logiska mindre delar
4. Redogör för TDD
5. Redogöra för versionshantering med Git och GitHub
6. Skapa terminalapplikationer med IDE och Java

Vad ni ska leverera

README.md (med instruktioner om kompilering och körning / testning)

Planering.txt (beskrivning av problemnedbrytning)

TollCalculator.java

Vehicle.java

Car.java

MotorBike.java

TollCalculatorTest.java (Klass med main metod som testar TollCalculator)

Uppgift

Vår stad har bestämt att ha Tullavgifter på vägarna i ett försök att skapa mindre flaskhalsar i trafiken. Folk ska få ekonomiska incitament att köra andra tider än vid rusning (8:00 - 9:00) och (16:00-17:00). Vår uppdragsgivare har gett oss konsulter uppdraget att designa ett system för att räkna ut avgiften för en enskild bil som passerar en vägkamera en viss tid på dygnet.

Kravspecifikation

- Avgifter varierar mellan 8 kr - 18 kr
- Det finns 2 fordonstyper: Bilar och Motorcyklar
 - Bilar kostar normalt 11 kr för att passera vägkamera
 - Motorcyklar kostar normalt 8 kr för att passera vägkamera
- Rusningstrafik mellan (8:00 - 9:00) och (16:00-17:00) ger maxtaxa (18 kr)
- Högsta dagliga avgiften får aldrig överskrida 60 kr
- En bil kan bara bli taxerad max en gång i timmen
 - Om en bil passerat en vägtull flera gånger under en timme appliceras den högsta avgiften av dessa.
- Lördagar och Söndagar är avgiftsfria

Planering

Ni ska planera ert arbete i denna uppgift genom att göra en problemnedbrytning (helst innan ni börjar koda, så ni har någon nytta av det). Problemnedbrytning ska dokumenteras i en textfil Planering.txt. Denna kan ha följande format (Sortering exempel) :

1. *Ta lista som input*
2. *Loopa igenom listan från första elementet framåt och spara index över minsta tal*
 1. *Byt plats på första elementet med index på minsta talet*
3. *Upprepa (punkt 2 men från andra elementet)*
4. *Upprepa (punkt 2 men från tredje elementet)*
5. *Upprepa till slutet på listan*
6. *Ta den sorterade listan och returnera.*

Klasser

Klassen TollCalculator

```

import java.util.*;

public class TollCalculator {

    // Huvudmetod som räknar ut dagsavgift för en bil
    public int getTollFee(Vehicle vehicle, Date... dates) {
        Date intervalStart = dates[0];
        int totalFee = 0;
        for (Date date : dates) {

        }
        return totalFee;
    }

    // Räkna ut avgift för en bil en viss tid på dygnet
    public int getTollFee(final Date date, Vehicle vehicle) {
        return 0;
    }

    // Räkna ut avgift för en bil en viss tid på dygnet
    private boolean isTollFreeDate(Date date) {
        return false;
    }

}

```

Interface Vehicle

```

public interface Vehicle {
    int getFee();
}

```

Klassen Car

```

public class Car implements Vehicle {
    public int getFee() {
        return 11;
    }
}

```

Klassen MotorBike

```

public class MotorBike implements Vehicle {
    public int getFee() {
        return 8;
    }
}

```

Ni ska också skriva klassen TollCalculatorTest med main metod som testat er kod. Ni ska verifiera i testerna att samtliga krav (se kravspecifikation) uppfylls av koden.

Betygskrav

Betyg G

Programmet ska uppfylla specifikationen

Projektet ska vara versionshanterat med git, med flera commits som beskriver utvecklingen av programmet (OBS: en commit kommer klassas som fusk och ge IG)

Betyg VG

Alla tester i TollCalculatorTest täcker kravspecifikationen

Ni har gjort en problemnedbrytning som avspeglas i programmets design

Projektet har en gitbranch per person som visar i flera commits vad var och en har gjort i projektet.