**ماشین حساب عوارض**

**هدف:**

هدف از این کار این است که به ما بیاموزد که به طور مستقل برنامه هایی در سبک Object Orientead ایجاد کنیم. شما همچنین در یک گروه 2 نفره کار خواهید کرد و یاد خواهید گرفت که از gitbrancher استفاده کنید تا کار خود را بین خود سازماندهی کنید. شما همچنین مهارت های برنامه ریزی و طراحی نرم افزار را به روشی ساختارمندتر با کاهش مشکلات و برنامه ریزی به دست خواهید آورد.

**اهداف دوره تحت پوشش تکالیف**

**دانش**

1. رایج ترین کتابخانه ها در پلتفرم جاوا را شرح دهید
2. نحوه تجزیه و تحلیل و تجزیه مسائل را به بخش های منطقی کوچکتر شرح دهید
3. TDD را توضیح دهید
4. کنترل نسخه را با Git و GitHub توضیح دهید
5. برنامه های ترمینال را با IDE و جاوا ایجاد کنید

**چیزی که قرار است تحویل دهید**

1. README.md (با دستورالعمل هایی در مورد تدوین و رانندگی / تست)
2. Planning.txt (توضیح تفکیک مشکل)
3. TollCalculator.java
4. Vehicle.java
5. Car.java
6. MotorBike.java
7. TollCalculatorTest.java (کلاس با روش اصلی که TollCalculator را آزمایش می کند)

**وظیفه**

شهر ما تصمیم گرفته است تا در جاده ها عوارض گمرکی داشته باشد تا گلوگاه های کوچک تری در ترافیک ایجاد کند. به مردم باید مشوق های مالی داده شود تا در زمان هایی غیر از ساعات شلوغی (8:00 تا 9:00) و (16:00-17:00) رانندگی کنند. مشتری ما به ما مشاوران وظیفه طراحی سیستمی را برای محاسبه کارمزد یک خودروی فردی که در ساعت معینی از روز از دوربین جاده عبور می کند، داده است.

**مشخصات مورد نیاز**

* هزینه ها بین 8 تا 18 کرون متفاوت است.
* دو نوع وسیله نقلیه وجود دارد: اتومبیل و موتور سیکلت.
  + اتومبیل معمولاً 11 کرون برای عبور از دوربین جاده هزینه دارد.
  + موتورسیکلت معمولاً 8 کرون برای عبور از دوربین جاده هزینه دارد.
* ترافیک ساعت شلوغی بین (8:00 - 9:00) و (16:00-17:00) حداکثر کرایه (18 کرون) را نشان می دهد.
* حداکثر هزینه روزانه نباید از 60 کرون تجاوز کند.
* یک خودرو فقط حداکثر یک بار در ساعت می تواند مشمول مالیات شود.
  + اگر خودرویی در یک ساعت چندین بار از عوارض عبور کرده باشد، بالاترین شارژ از این عوارض اعمال می شود.
* شنبه و یکشنبه رایگان است.

**برنامه ریزی**

شما باید کار خود را در این کار با انجام یک تفکیک مشکل (ترجیحاً قبل از شروع کدنویسی) برنامه ریزی کنید تا از آن بهره مند شوید. تفکیک مشکل باید در یک فایل متنی Planning.txt مستند شود. این ممکن است موارد زیر را داشته باشد

قالب (مثال مرتب سازی):

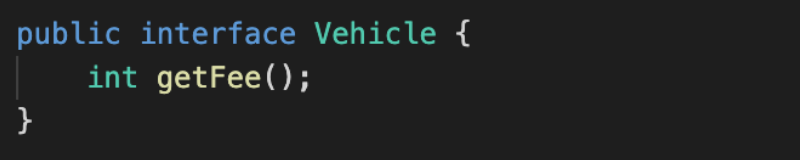
1. فهرست را به عنوان ورودی در نظر بگیرید.
2. فهرست را از اولین عنصر به جلو مرور کنید و فهرست کوچکترین عدد را ذخیره کنید.
   1. جای اولین عنصر را با شاخص کوچکترین عدد عوض کنید.
3. تکرار کنید (نقطه 2 اما از عنصر دوم).
4. تکرار کنید (نقطه 2 اما از عنصر سوم).
5. .... تا انتهای لیست تکرار کنید.
6. فهرست مرتب شده را بردارید و برگردید.

**کلاس ها**

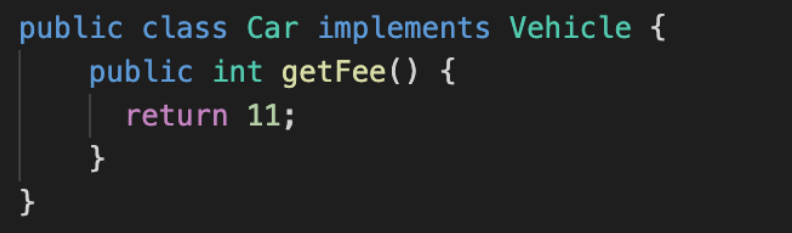
کلاس های TollCalculator



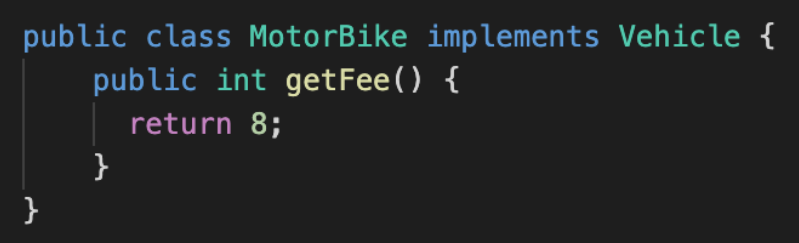
وسیله نقلیه رابط



کلاس ماشین



کلاس های موتورسیکلت



همچنین باید کلاس TollCalculatorTest را با روش اصلی که کد شما را آزمایش می کند بنویسید. شما باید در آزمایشات تأیید کنید که همه الزامات (مشخصات الزامات را ببینید) با کد مطابقت دارد.

**الزامات درجه بندی**

**رتبه G**

برنامه باید مشخصات زیر را داشته باشد:

پروژه باید با نسخه مدیریت شده با git، با چندین commit که توسعه برنامه را توصیف می کند (توجه: یک commit به عنوان تقلب طبقه بندی می شود و IG می دهد)

**رتبه بندی VG**

تمام تست های TollCalculatorTest مشخصات مورد نیاز را پوشش می دهند.

شما یک مشکل ایجاد کرده اید که در طراحی برنامه منعکس شده است.

این پروژه دارای یک gitbranch برای هر نفر است که در چندین commit نشان می دهد که هر کدام در پروژه چه کاری انجام داده اند.