

※概要欄から提供プログラムをダウンロード！ ↓

▶ ポイント

1. 抽象クラス、抽象メソッド
2. インタフェース、実装
3. パッケージ、インポート
4. 参照型の型変換



▶ 問題

1

タクシーを管理する Taxi クラスと電車を管理する Train クラスがあります。Taxi.java と Train.java は提供プログラムを使うものとします。

Taxi クラスと Train クラスがもつメンバ変数とメソッドは以下の表のとおりです。

Taxi クラス	
メンバ変数	
private int crewNum	タクシーの乗客人数
メソッド	
public Taxi(int crewNum)	変数 crewNum に引数を設定する
public void showCrewNum()	変数 crewNum の値を表示する

Train クラス	
メンバ変数	
private int crewNum	電車の乗客人数
メソッド	
public Train(int crewNum)	変数 crewNum に引数を設定する
public void showCrewNum()	変数 crewNum の値を表示する

Taxi.java

```
1 | public class Taxi{
2 |     private int crewNum;
3 |
4 |     public Taxi(int crewNum){
5 |         this.crewNum = crewNum;
6 |     }
7 |     void showCrewNum(){
8 |         System.out.println("タクシーの乗客：" + crewNum + "名");
9 |     }
10 | }
```

Train.java

```
1 | public class Train{
2 |     private int crewNum;
3 |
4 |     public Train(int crewNum){
5 |         this.crewNum = crewNum;
6 |     }
7 |     void showCrewNum(){
8 |         System.out.println("電車の乗客：" + crewNum + "名");
9 |     }
10 | }
```

この2つのクラスの共通部分をまとめて、乗り物を管理する抽象クラスとして Vehicle クラスを定義しましょう。その後、Vehicle クラスをもとにしてバスを管理する Bus クラスを定義します。

Vehicle クラスと Bus クラスがもつメンバ変数とメソッドは以下の表のとおりです。
Bus クラスは Vehicle クラスを継承するものとします。

Vehicle クラス	
メンバ変数	
<code>protected int crewNum</code>	乗客人数
メソッド	
<code>public abstract void showCrewNum()</code>	変数 <code>crewNum</code> の値を表示する (抽象メソッド)

Bus クラス	
メソッド	
<code>public Bus(int crewNum)</code>	変数 <code>crewNum</code> に引数を設定する
<code>public void showCrewNum()</code>	変数 <code>crewNum</code> の値を表示する

Taxi クラスと Train クラスも Vehicle クラスを継承するように修正します。また、メンバ変数 `crewNum` は Vehicle で定義したため、Taxi クラスと Train クラスからは削除します。

2

乗客が乗り物を止めるための処理（stop メソッド）を持つ Stopable インタフェースを定義します。Stopable インタフェースがもつメソッドは以下の表のとおりです。

Stopable インタフェース	
メソッド	
<code>void stop()</code>	停止するメッセージを表示する (抽象メソッド)

Taxi クラスと Bus クラスに Stopable インタフェースを実装します。問題 1 で作成した Taxi.java と Bus.java を修正してください。

Taxi クラスの stop メソッドでは「ここで降ります」と表示し、Bus クラスの stop メソッドでは「次止まります」と表示してください。

3

問題1と問題2で使ったクラスとインタフェース（Vehicle クラス、Stopable インタフェース、Taxi クラス、Train クラス、Bus クラス）をすべて mypack パッケージに格納します。問題2までに作成したファイルを修正してください。

パッケージ化の記述を追加したら、実行用クラス UseVehicle を実行します。UseVehicle.java は提供プログラムにインポートの記述を追加して使います。

UseVehicle.java

```
1  //インポート
2
3
4  public class UseVehicle{
5      public static void main(String[] args){
6          Vehicle[] v = new Vehicle[3];
7          v[0] = new Taxi(4);
8          v[1] = new Train(160);
9          v[2] = new Bus(80);
10
11         for(int i = 0; i < v.length; i++){
12             v[i].showCrewNum();
13             if(v[i] instanceof Stopable){
14                 Stopable s = (Stopable)v[i];
15                 s.stop();
16             }
17         }
18     }
19 }
```

コンパイル・実行例

```
>javac -d . *.java
```

```
>java UseVehicle
```

タクシーの乗客：4 名

ここで降ります

電車の乗客：160 名

バスの乗客：80 名

次止まります



＼フリーラーニング（無料で学べる場）をもっと広げたい！／
チャンネル登録や拡散よろしくお願いします！



NEXT DOOR

