オブジェクト指向応用
終了後テスト

テストの注意点
・所要時間は30分です。問題2問、ボーナス問題1問です。
・IntelliJを使用してください。
・intellij-project内に「_3test-object-oriented2」というプロジェクトを作成し、その中にプログラムを書いてください。
・テスト中はインターネットやテキストなどを閲覧してはいけません。
・問題の意図、意味がわからなければ講師に質問してください。

Ver.202504
作成者：伊賀　将之

第1問：(配点50点)
Car(車)クラスの作成

- 第1問目は「**exam1パッケージ**」の中にクラスを作成してください
- 以下の要素を持つCarクラスを作成してください
- カプセル化も意識してアクセス修飾子やGetter/Setterも記述してください

要素	変数名・メソッド名	属性・操作の意味
属性	String name	名前 (サブクラスで利用します)
操作	void putOnGas()	給油する (どの車にもある共通機能) ・「【nameの値】 ＋ "に給油します」と表示する
	void run()	走る ・「ブーン！車が走ります」と表示する

第1問：サブクラスの作成

- 以下の要素を持つ**Carクラスのサブクラスを２つ**作成してください
- SuperCarクラス

要素	変数名・メソッド名	属性・操作の意味
操作	void run()	走る (オーバーライドした独自機能) ・「ブオーン！【nameの値】が走ります」と表示する

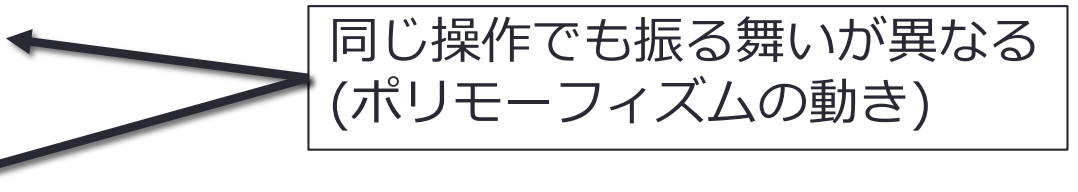
- EcoCarクラス

要素	変数名・メソッド名	属性・操作の意味
操作	void run()	走る (オーバーライドした独自機能) ・「シーン！【nameの値】が走ります」と表示する

第1問：実行用クラスの作成

- 実行用のクラスであるExam1クラスを作成してください
- 実行したら以下の結果になるようにmain()メソッドを作成してください(車の名前はnameのSetterでセットします)
- この際、SuperCarオブジェクトとEcoCarオブジェクトは1つのCar型のcarという変数に代入し(carという変数は1度だけ宣言をして使いまわすという意味)、同じ操作(car.run()という記述)で異なる振る舞いをするポリモーフィズムの動きを実現してください
- (もし変数を使い回すという意味がわからない場合は、気にせずいつも通りのインスタンス化の方法でプログラムを書いてください。)

実行結果==========
フェラーリに給油します
ブォーン！フェラーリが走ります
プリウスに給油します
シーン！プリウスが走ります
==========



第2問：(配点50点)
ポリモーフィズム動物園

- 第2問目は「**exam2パッケージ**」の中にクラスやインターフェースを作成してください
- 以下の要素を持つAnimalインターフェースを作成してください

要素	変数名・メソッド名	属性・操作の意味
操作	void cry()	泣く　抽象メソッド

- 以下のクラス群を作成してください(全てのクラスはAnimalインターフェースを実装します)

クラス名	cry()の実装でやること
Sheep	「baa」と泣く(表示する)
Horse	「whinny」と泣く(表示する)
Goat	「bleat」と泣く(表示する)

第2問：実行用クラスの作成

- main()メソッドを含む実行用のZooクラスを作成してください
- main()メソッドでは以下を実施してください
  - Animal型の変数が3個ある配列(animals)を用意する
  - この配列に羊、馬、やぎを1匹ずつを入れる
    ※それぞれの動物はAnimalインターフェースを実装しているので
    Animal型の配列に入れることができます
  - for文または拡張for文で配列の要素の数だけループさせ、一つ一つの動物を取り出す
  - ループの中で取りだした動物のcry()メソッドを呼び泣かせる

```
実行結果
====================
baa
whinny
bleat
====================
```

ボーナス問題：(配点20点)
３つのインターフェース作成

- ボーナス問題は「**exambonusパッケージ**」の中にクラスやインターフェースを作成してください
- 以下の３つのインターフェースを作成してください
- Carインターフェース

要素	変数名・メソッド名	属性・操作の意味
操作	void run()	走る 抽象メソッド

- Airplaneインターフェース

要素	変数名・メソッド名	属性・操作の意味
操作	void fly()	飛ぶ 抽象メソッド

- TimeMachineインターフェース

要素	変数名・メソッド名	属性・操作の意味
操作	void timeTravel()	時間移動 抽象メソッド

ボーナス問題：空飛ぶタイムマシン車(デロリアン)の作成

- 先ほど作った３つのインターフェースを実装する以下の要素を持つDeLoreanクラスを作成してください。

要素	変数名・メソッド名	属性・操作の意味
操作	void run()	走る ・「デロリアンが走る！」と表示する
	void fly()	飛ぶ ・「デロリアンが飛ぶ！」と表示する
	void timeTravel()	時間移動 ・「デロリアンがタイムテレポート！」と表示する
	void engineStart()	エンジン起動 ・このクラスにあるrun()メソッドを呼ぶ ・次にfly()メソッドを呼ぶ ・次にtimeTravel()メソッドを呼ぶ

ボーナス問題：実行用クラスの作成

- 実行用のクラスであるExamBonusクラスを作成してください
- main()メソッドを作成し、以下の実行結果になるようプログラムを書いてください
- この際、DeLoreanオブジェクト生成した後、**3つのメソッドを呼んで表示させるのではなく、エンジンを起動する１つのメソッドを呼んで表示させてください**

実行結果＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝
デロリアンが走る！
デロリアンが飛ぶ！
デロリアンがタイムテレポート！
＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝