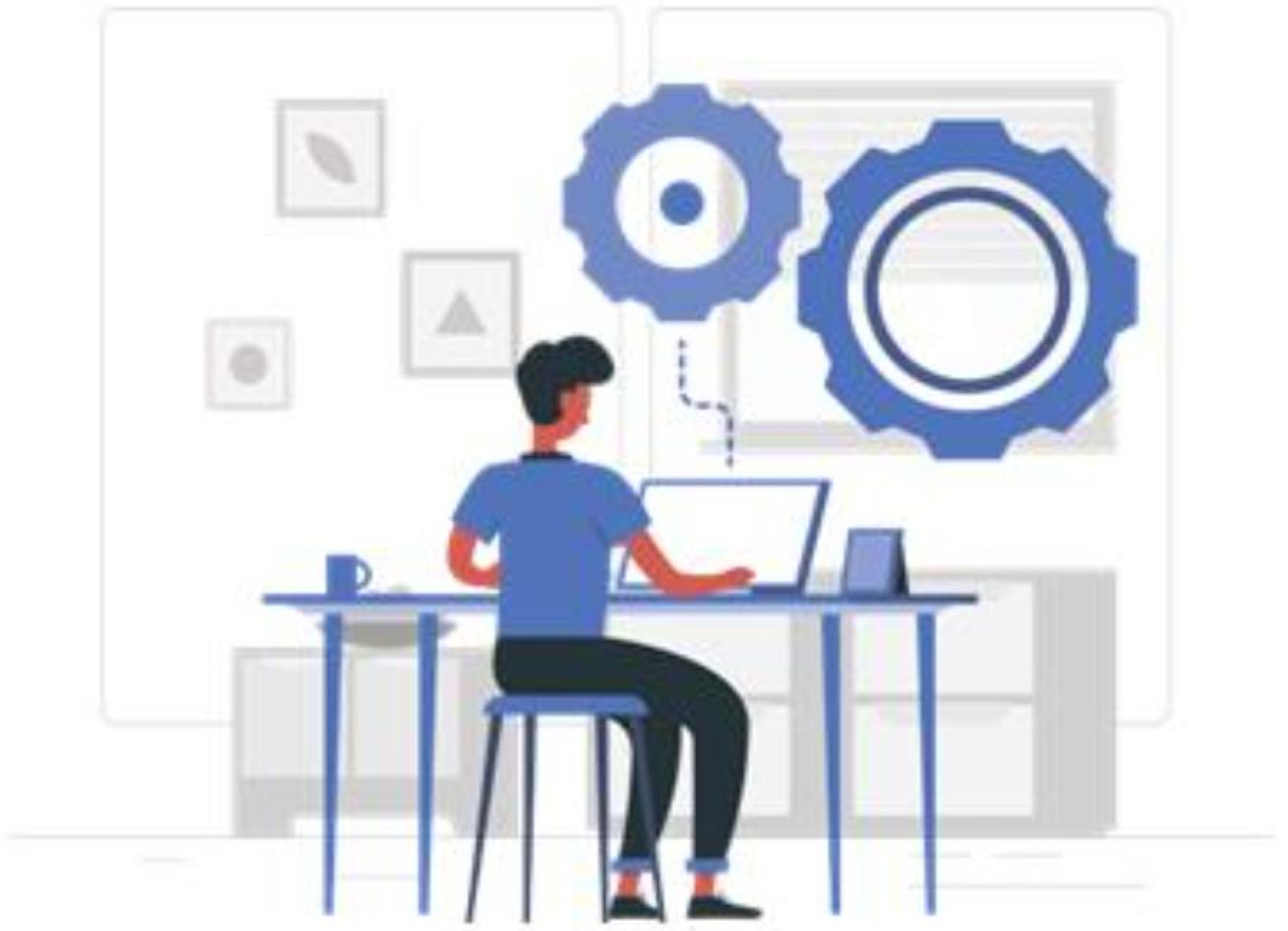


Practice

Implement Polymorphism

Exercise

- Practice 1: Students' Grades
- Practice 2: Vehicle

PRACTICE

Practice 1: Students' Grades

The total grades of all 12th-grade students, regardless of their chosen pathway, must be calculated. The subjects vary depending on their choice of pathway. For example,

A student who has opted for the medical pathway will get grades for math, physics, chemistry, and biology.
A student who has opted for the non-medical stream will get grades for math, physics, and chemistry.
A student who has chosen the business pathway will get grades for business studies, finance, and accounting.

Tasks

- Write a program with overloaded methods that will help calculate the grades of of 12th grade students in different pathways.
- Create a class named `StudentMarks` inside the package `com.marks`
- Define `calculateMarks`() methods which will calculate the total marks for medical pathway students and return the sum of the total marks.

```
public int calculateMarks(int math, int physics, int chemistry ,
int biology ){
return sum;
}
```

- Define `calculateMarks`() methods which will calculate the total marks for non-medical pathway students and return the sum of the total marks.
- Define `calculateMarks`() methods which will calculate the total marks for business pathway students and return the sum of the total marks.

Tasks (cont'd)

- Create the implementation class `StudentImpl` inside the package com.marks
- Declare and initialize Objects of the `StudentMarks` class inside the main method of the `StudentImpl` class.
- Call all the overloaded `calculateMarks`() method and pass the respective parameters.
- Display the value returned from the `calculateMarks()` method inside the main method.

PRACTICE

Practice 2: Vehicle

John is new to Java and is working on polymorphism. He wants to create a `Bike` class and a `Car` class. In the future, he may want to create a few more vehicles, so he wants some features to be overridden from the abstract class `VehicleManufacturer` and from the interface `Vehicle`.

Help John achieve this task.

Tasks

- Create an abstract class `VehicleManufacturer` inside the package com.vehicles
- Declare private instance variable `vehicleName, vehicleModelName` and `vehicletype` with appropriate datatype.
- Create default and Parameterized constructor.
- Declare getters and setters for the instance variable.
- Define abstract method `getManufacturerInformation`() having String as a return type.

Tasks (cont'd)

- Declare `Vehicle` as an interface inside the package defined, with an abstract method

  ```
  int maxSpeed(String vehicleType).
  ```

- Create `Bike` class inside the package defined, that will extend `VehicleManufacturer` and implement `Vehicle` interface and will override all the `abstract` methods.
- Create parameterized constructor to initialize all the super class variables.
- Inside the `Bike` class `maxSpeed`() method should return maximum speed depending upon their types:
  - If vehicleType is equal to sportsBike then return speed as 300kmh
  - If vehicleType is equal to cruiser, then return speed as 170kmh
- Inside the `Bike` class `getManufacturerInformation`() method should return output in the format : `Bike{Manufacturer name:'name',Model Name:'modelName',Type:'type'}`

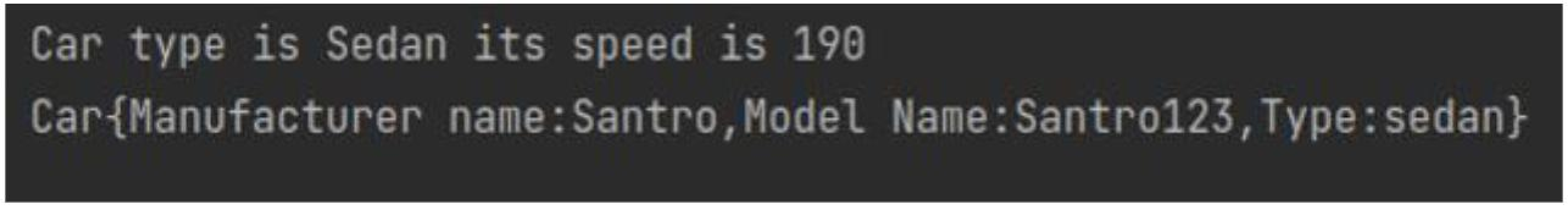
Tasks (cont'd)

- Create a `Car` class that will extend `VehicleManufacturer`, implement the Vehicle interface, and override all the abstract methods.
- Create a parameterized constructor to initialize all the super class variables.
- Inside the `Car` class, the `maxSpeed()` method should return the maximum speed depending on the type:
  - If vehicleType is equal to SportsCar, then the return speed is 250 km/h.
  - If vehicleType is equal to Sedan, then the return speed is 170 km/h.
- Inside the Car class, the `getManufacturerInformation()` method should return output in the following format:

```
Car{Manufacturer name:'name',Model Name:'modelName',Type:'type'}.
```

Tasks (cont'd)

- Create the `VehicleService` class inside the package com.vehicle.
- Declare the main method and inside the main method:
  - Create object of `Bike` class by calling parameterized constructor and passing all the parameters value.
  - Call the `maxSpeed`() mathod and print the int value returned by the method.
  - Call the `getManufacturerInformation`() method and print the String value returned by the method.
- Sample Output for Car Object:



```
Car type is Sedan its speed is 190
Car{Manufacturer name:Santro,Model Name:Santro123,Type:sedan}
```