A Student wants to create a game called “Life” , ’life’ is an RPG game in which a player can move up, down, left & nght in order to implement this game, assume that you need to create an absiraction of the player controllers

I. Implement a new player extended from movements when each movement method is executed print out the direction that the player has moved

```
interface Movement {
   void moveUp();
   void moveDown();
   void moveLeft();
   void moveRight();
}

class Player implements Movement {
   @Override
   public void moveUp() {
      System.out.println("Player moved up");
   }

   @Override
   public void moveDown() {
      System.out.println("Player moved down");
   }

   @Override
   public void moveLeft() {
      System.out.println("Player moved left");
   }

   @Override
   public void moveRight() {
      System.out.println("Player moved right");
   }
}


public class Main {
   public static void main(String[] args) {
      Movement player = new Player();
      player.moveUp();
      player.moveDown();
      player.moveLeft();
```

```
        player.moveRight();
    }
}
```

II. Assume that there are three types of players, those who move in regular directions and opposite directions when a movement is performed and others who jump by 5 spaces at a time except when crouched where he moves by only two spaces those who move in the opposite direction move 2 spaces at a time implement separate players and print their movement

```
abstract class Player {
    public abstract void moveUp();
    public abstract void moveDown();
    public abstract void moveLeft();
    public abstract void moveRight();
}

class RegularPlayer extends Player {
    @Override
    public void moveUp() {
        System.out.println("Regular Player moved up");
    }

    @Override
    public void moveDown() {
        System.out.println("Regular Player moved down");
    }

    @Override
    public void moveLeft() {
        System.out.println("Regular Player moved left");
    }

    @Override
    public void moveRight() {
        System.out.println("Regular Player moved right");
    }
}

class OppositePlayer extends Player {
    @Override
    public void moveUp() {
        System.out.println("Opposite Player moved down");
    }
```

```
    @Override
    public void moveDown() {
        System.out.println("Opposite Player moved up");
    }

    @Override
    public void moveLeft() {
        System.out.println("Opposite Player moved right");
    }

    @Override
    public void moveRight() {
        System.out.println("Opposite Player moved left");
    }
}

class JumpingPlayer extends Player {
    @Override
    public void moveUp() {
        System.out.println("Jumping Player moved up by 5 spaces");
    }

    @Override
    public void moveDown() {
        System.out.println("Jumping Player moved down by 5 spaces");
    }

    @Override
    public void moveLeft() {
        System.out.println("Jumping Player moved left by 5 spaces");
    }

    @Override
    public void moveRight() {
        System.out.println("Jumping Player moved right by 5 spaces");
    }
}

class CrouchingPlayer extends JumpingPlayer {
    @Override
    public void moveUp() {
        System.out.println("Crouching Player moved up by 2 spaces");
    }
```

```
    @Override
    public void moveDown() {
        System.out.println("Crouching Player moved down by 2 spaces");
    }

    @Override
    public void moveLeft() {
        System.out.println("Crouching Player moved left by 2 spaces");
    }

    @Override
    public void moveRight() {
        System.out.println("Crouching Player moved right by 2 spaces");
    }
}


public class Main {
    public static void main(String[] args) {
        Player regularPlayer = new RegularPlayer();
        regularPlayer.moveUp();
        regularPlayer.moveDown();
        regularPlayer.moveLeft();
        regularPlayer.moveRight();

        Player oppositePlayer = new OppositePlayer();
        oppositePlayer.moveUp();
        oppositePlayer.moveDown();
        oppositePlayer.moveLeft();
        oppositePlayer.moveRight();

        Player jumpingPlayer = new JumpingPlayer();
        jumpingPlayer.moveUp();
        jumpingPlayer.moveDown();
        jumpingPlayer.moveLeft();
        jumpingPlayer.moveRight();

        Player crouchingPlayer = new CrouchingPlayer();
        crouchingPlayer.moveUp();
        crouchingPlayer.moveDown();
        crouchingPlayer.moveLeft();
        crouchingPlayer.moveRight();
    }
}
```