Test on inheritance, abstraction and polymorphism

```
import java.awt.*;

interface Displayable{

   public void display(int x, int y);
   public void display(int x, int y, int i);

}

interface Movable{

   public void moveLeft();
   public void moveRight();
   public void moveUp();
   public void moveDown();
}

abstract class CartoonPart implements Displayable{

   private Image[] images;
   public void display(int x, int y){
   // code which displays the default image
   }

   public void display(int x, int y, int i){
   // code which displays the ith image
   }

}

abstract class CartoonCharacter extends CartoonPart implements Movable{

   abstract public void moveLeft();
   abstract public void moveRight();
   abstract public void moveUp();
   abstract public void moveDown();

   public void jumpUpNDown(){
      moveUp();
      moveDown();
   }

   public void panic(){
      moveLeft();
      moveRight();
      jumpUpNDown();
   }
}
```

```
class House extends CartoonPart{

   public void doorOpen() {
      // show the picture of house with the door open
   }

   public void doorClose() {
      // show the picture of house with the door closed
   }

   public void fallDown() {
      // collapse in  ruins
   }

}

class Duck extends CartoonCharacter{
// code for methods moveLeft(), moveRight(), moveUp(), moveDown()
}

class Rabbit extends CartoonCharacter{
// code for methods moveLeft(), moveRight(), moveUp(), moveDown()
}

class Wolf extends CartoonCharacter{
// code for methods moveLeft(), moveRight(), moveUp(), moveDown()
}

class Hunter extends CartoonCharacter{
// code for methods moveLeft(), moveRight(), moveUp(), moveDown()
}
```

Questions:

1. Does an object of type House have `display(int x, int y)` method?
2. Does an object of type House have `moveLeft()` method?
3. Does an object of type Hunter have `images` field?
4. Does an object of type Hunter have `panic` method?
5. Suppose that a method has the following signature:
   `public static void animation(Displayable[] m)`
   and variable `ducks` is of type `CartoonCharacter[]`. Would the compiler report an error if we pass `ducks` to `animation`, i.e. call `animation(ducks)`?
6. What if in the example above `ducks` is of type `ChartoonPart[]`?
7. What if we have a method
   `public static void parade(CartoonCharacter[] c)`
   and `ducks` is of type `Displayable[]` and we call `parade(ducks)`?
8. What if in the example above `ducks` is of type `Duck[]`?