

Ejercicio 6:

Explicar salidas y errores del siguiente código:

```
public static void main(String[] args) {  
    // Using the subclasses  
    Cat cat1 = new Cat();  
    cat1.greeting();  
    Dog dog1 = new Dog();  
    dog1.greeting();  
    BigDog bigDog1 = new BigDog();  
    bigDog1.greeting();  
    // Using Polymorphism  
    Animal animal1 = new Cat();  
    animal1.greeting();  
    Animal animal2 = new Dog();  
    animal2.greeting();  
    Animal animal3 = new BigDog();  
    animal3.greeting();  
    Animal animal4 = new Animal();  
    // Downcast  
    Dog dog2 = (Dog) animal2;  
    BigDog bigDog2 = (BigDog) animal3;  
    Dog dog3 = (Dog) animal3;  
    Cat cat2 = (Cat) animal2;  
    dog2.greeting(dog3);  
    dog3.greeting(dog2);  
    dog2.greeting(bigDog2);  
    bigDog2.greeting(dog2);  
    bigDog2.greeting(bigDog1);  
}
```

1. Desde antes de correr el programa nos damos cuenta que ya marca un error en “Animal”. Esto es porque está intentando declarar un objeto de instancia Animal, lo cual no se puede porque la clase Animal es abstracta.
2. Las primeras salidas es un saludo de los primeros animales declarados al inicio.

```
public static void main(String[] args) {  
    // Using the subclasses  
    Cat cat1 = new Cat();  
    cat1.greeting();  
    Dog dog1 = new Dog();  
    dog1.greeting();  
    BigDog bigDog1 = new BigDog();  
    bigDog1.greeting();  
    // Using Polymorphism  
    Animal animal1 = new Cat();  
    animal1.greeting();  
    Animal animal2 = new Dog();  
    animal2.greeting();  
    Animal animal3 = new BigDog();  
    animal3.greeting();  
    Animal animal4 = new Animal();  
    // Downcast  
    Dog dog2 = (Dog) animal2;  
    BigDog bigDog2 = (BigDog) animal3;  
    Dog dog3 = (Dog) animal3;  
    Cat cat2 = (Cat) animal2;  
    dog2.greeting(dog3);  
    dog3.greeting(dog2);  
    dog2.greeting(bigDog2);  
    bigDog2.greeting(dog2);  
    bigDog2.greeting(bigDog1);  
}
```

run:
Meow!
Woof!
Woof!
Meow!
Woof!
Woof!

Exception in thread "main" java.lang.ClassCastException: Ejercicio05.TestAnimal.main(TestA
C:\Users\ARTURO.000\AppData\Local\NetBeans\C
BUILD FAILED (total time: 0 seconds)

3. Las siguientes 3 salidas son los mismos saludos de animales pero que ahora fueron declarados usando polimorfismo.

```

public static void main(String[] args) {
    // Using the subclasses
    Cat cat1 = new Cat();
    cat1.greeting();
    Dog dog1 = new Dog();
    dog1.greeting();
    BigDog bigDog1 = new BigDog();
    bigDog1.greeting();
    // Using Polymorphism
    Animal animal1 = new Cat();
    animal1.greeting();
    Animal animal2 = new Dog();
    animal2.greeting();
    Animal animal3 = new BigDog();
    animal3.greeting();
    Animal animal4 = new Animal();
    // Downcast
    Dog dog2 = (Dog)animal1;
    BigDog dog2 = (BigDog)animal3;
    Dog dog3 = (Dog)animal3;
    Cat cat2 = (Cat)animal2;
    dog2.greeting(dog3);
    dog3.greeting(dog2);
    dog2.greeting(bigDog2);
    bigDog2.greeting(dog2);
    bigDog2.greeting(bigDog1);
}

```

```

EUN!
Woof!
Woof!
Woof!
Woof!
Woof!
Exception in thread "main" java.lang.ClassCast
    at Ejercicio08.TestAnimal.main(TestA
    Ca/Users/ARTURO_000/AppData/Local/NetBeans/C
BUILD FAILED (total time: 0 seconds)

```

4. Al final marca una excepción porque se está intentando hacer un downcast de gato a perro, pero esas clases no están conectadas entre ellas, ni son del mismo tipo, es por eso que marca una excepción.

```

public static void main(String[] args) {
    // Using the subclasses
    Cat cat1 = new Cat();
    cat1.greeting();
    Dog dog1 = new Dog();
    dog1.greeting();
    BigDog bigDog1 = new BigDog();
    bigDog1.greeting();
    // Using Polymorphism
    Animal animal1 = new Cat();
    animal1.greeting();
    Animal animal2 = new Dog();
    animal2.greeting();
    Animal animal3 = new BigDog();
    animal3.greeting();
    Animal animal4 = new Animal();
    // Downcast
    Dog dog2 = (Dog)animal1;
    BigDog bigDog2 = (BigDog)animal3;
    Dog dog3 = (Dog)animal3;
    Cat cat2 = (Cat)animal2;
    dog2.greeting(dog3);
    dog3.greeting(dog2);
    dog2.greeting(bigDog2);
    bigDog2.greeting(dog2);
    bigDog2.greeting(bigDog1);
}

```

```

run:
Meow!
Woof!
Woof!
Meow!
Woof!
Woof!
Exception in thread "main" java.lang.ClassCastException:
    at Ejercicio05.TestAnimal.main[TestC
C:\Users\ARTURO_000\AppData\Local\NetBeans\
BUILD FAILED (total time: 0 seconds)

```

5. Si quitamos esa línea, dejará de marcar la excepción y podremos ver las siguientes salidas, las cuales son saludos entre perros después de haber hecho el downcast de animales a perros específicos.

```

public static void main(String[] args) {
    // Using the subclasses
    Cat cat1 = new Cat();
    cat1.greeting();
    Dog dog1 = new Dog();
    dog1.greeting();
    BigDog bigDog1 = new BigDog();
    bigDog1.greeting();
    // Using Polymorphism
    Animal animal1 = new Cat();
    animal1.greeting();
    Animal animal2 = new Dog();
    animal2.greeting();
    Animal animal3 = new BigDog();
    animal3.greeting();
    Animal animal4 = new Animal();
    // Downcast
    Dog dog2 = (Dog)animal2;
    BigDog bigDog2 = (BigDog)animal3;
    Dog dog3 = (Dog)animal3;
    Cat cat2 = (Cat)animal2;
    dog2.greeting(dog3);
    dog3.greeting(dog2);
    dog2.greeting(bigDog2);
    bigDog2.greeting(dog2);
    bigDog2.greeting(bigDog1);
}

```

```

run:
Meow!
Woof!
Woof!
Meow!
Woof!
Woof!
Woowooooof!
Woowoooooww!
Woowooooof!
Woowoooooww!
Woowoooooww!
Woowoooooww!
BUILD SUCCESSFUL (total time: 0 seconds)

```