1. Cuando el método en la superclase y en la subclase tienen distinta signatura de tipo estamos hablando de: **Redefinición**
2. En Java, el tipo de enlace para el polimorfismo en herencia para métodos privados, finales, de clase y atributos es: **Estático**
3. En C++, el tipo de enlace por defecto para el polimorfismo en herencia para métodos virtuales es: **Dinámico**
4. El downcasting en Java siempre se especifica de forma: **Explícita**
5. Cuando el método de la subclase oculta el acceso al método de la superclase, estamos hablando de: **Shadowing**
6. En C++, donde las redefiniciones en las subclases ocultan a las definiciones de las superclases, se usa la estrategia de redefinición denominada: **Shadowing**
7. En Java, el tipo de enlace para el polimorfismo en herencia para métodos de instancia públicos y protegidos es: **Dinámico**
8. En Java, para que una clase no sea polimórfica debemos usar la palabra clave: **Final**
9. Las variables que pueden referenciar más de un tipo de objeto se denominan: **Polimórficas**
10. La sobrecarga de métodos de clase se resuelve en tiempo de: **Compilación**
11. En Java, si el método de la subclase se denomina igual que el de la superclase y tiene la misma signatura, usamos la anotación @Override, y por tanto es una: **Sobrescritura**
12. Las clases que tienen algún método con enlace dinámico se denominan: **Polimórficas**
13. Las llamadas a métodos sobrecargados se resuelven en tiempo de: **Compilación**
14. La sobrecarga de operadores es un ejemplo de sobrecarga basada en ...: **Signatura de tipos**
15. En C++, para que una clase sea polimórfica debe tener al menos un método: **Virtual**
16. Como alternativa a la sobrecarga, en C -p.ej. con la función printf-, o en Java -p.ej con un método en void F(int ... p)-, tenemos las funciones: **Poliádicas**
17. El mecanismo que permite conocer los tipos en tiempo de ejecución se denomina: **RTTI**
18. En Java, el tipo de enlace para el polimorfismo en herencia para métodos de instancia públicos y protegidos es: **Dinámico**
19. En Java, por defecto todas las clases son: **Polimórficas**
20. En Java, esta llamada usa: Object o = new String();: **Upcasting**
21. Las variables que pueden referenciar más de un tipo de objeto se denominan: **Polimórficas**
22. El método toString() en Java es un ejemplo de sobrecarga de métodos basada en ...: **Ámbito**
23. En Java, donde conviven las distintas redefiniciones que encontramos en la jerarquía de herencia, se usa la estrategia de redefinición denominada: **Mezcla**