**CLASES ABSTRACTAS**

* Una clase abstracta puede contener o no métodos abstractos

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| Si una subclase no implementa todos o ninguno de los métodos de una superclase abstracta, la subclase debe definirse como abstracta.    abstract classes can extend other abstract classes and are not required to provide implementations for any of the abstract methods. | A la sub clase se le pueden agregar métodos con implementación y métodos abstractos. |

|  |  |
| --- | --- |
| It follows, then, that a concrete class that extends an abstract class must implement all inherited abstract methods.        The class Lion is not marked as abstract, and as the first concrete subclass, it must implement all inherited abstract methods not defined in a parent class. | There is one exception to the rule for abstract methods and concrete classes: a concrete subclass is not required to provide an implementation for an abstract method if an intermediate abstract class provides the implementation.        Si Lion implementa getName, entonces sobre escribe el método de BigCat |

|  |  |
| --- | --- |
|  | if an intermediate class provides an implementation for an abstract method, that method is inherited by subclasses as a concrete method, not  as an abstract one.  In other words, the subclasses do not consider it an inherited abstract method because it is no longer abstract by the time it reaches the subclasses. |

|  |  |
| --- | --- |
| Los métodos abstractos no son estáticos    Error: | Los métodos normales de una clase abstracta si pueden ser estáticos |

|  |  |
| --- | --- |
| Una subclase puede ser abstracta incluso si su superclase (clase normal, clase concreta) no lo es | Una clase abstracta no puede ser final o privada, ya que habrá sub clases que la tienen que extender    **Error: The class can be either abstract or final, not both.**  Un método abstracto no puede ser marcado como **final** porque será implementado en una subclase.    **The abstract method metodoG in type ClaseAbstracta can only set a visibility modifier, one of public or protected**  Un método abstracto no puede ser marcado como privado, ya que la sub clases no tendrían acceso.    **The abstract method metodoG in type ClaseAbstracta can only set a visibility modifier, one of public or protected** |

|  |  |
| --- | --- |
| Una subclase no puede reducir la visibilidad del método que sobre escribe.  Modificador default    Modificador protected    Modificador public | En cualquiera de los tres casos, marca el error:    **Cannot reduce the visibility of the inherited method from ClaseAbstracta**  Ya que **metodoG** es declarado como **default/public/protected** en la super clase, debe ser declarado con el mismo nivel en la subclase o como public. |

|  |  |
| --- | --- |
| Una clase abstracta puede tener un método main    Abstract just means you can't instantiate the class directly.  Loading a class is not the same as creating an instance of the class. And there's no need to create an instance of the class to call main(), because it's static. So there's no problem. | La sub clase anterior no implementa todos los métodos de su super clase, por lo tanto, marcará error. Esta sub clase es una clase concreta.  Una clase concreta/no abstracta puede extender a otra clase concreta/no abstracta.    La clave es que la clase debe implementar todos los métodos abstractos que su super clase concreta no implementa.  Aunque la sub clase (clase concreta) **Flamingo** implementa el método abstracto **getName(),** su super clase **Ave** no implementa este método, por lo tanto no compilará. |

|  |  |
| --- | --- |
|  |  |

* with abstract classes, you can declare fields that are not static and final
* and define public, protected, and private concrete methods.
* An abstract class may have static fields and static methods.
* ¿Cuál es el modificador por defecto de una variable?
* Clase abstracta privada