

Clases Abstractas e Interfaces

Ernesto Rodriguez

Universidad del Itsmo

erodriguez@unis.edu.gt

Clases Abstractas: Motivación

- Muchas veces tenemos una familia de objetos que comparten características en comun.
- Sin embargo, no tiene sentido o utilidad crear instancias de un objeto arriba en la jerarquia de clases.
- Esto podria resultar en una clase que nunca sera instanciada directamente.
- Muchas veces, esa clase va tener metodos vacios, con el proposito que las sub-clases los cambien.

Solución: *Clases abstractas*

Clases Abstractas

- Clases cuyo proposito no es crear instancias, sino permitir que otras clases compartan propiedades y metodos.
- No es possible crear una instancia de una clase abstracta.
- Sin embargo, una clase abstracta puede utilizarse como tipo, ya sea de una propiedad o variable.
- La clase abstracta puede definir *metodos abstractos*.
- Se declaran mediante la palabra reservada `abstract`.

- Un tipo especial de metodo que solo pede existir en una clase abstracta.
- El metodo no tiene cuerpo, sin embargo tiene una *firma*.
- La *clase abstracta* puede hacer uso de sus *metodos abstractos*. Incluso, a pesar que el metodo no tiene cuerop.
- Toda clase ordinaria que herede de una clase abstracta **debe** definir un cuerpo para todos los *metodos abstractos*.
- Una clase abstracta puede heredar de otra clase abstracta **sin** necesidad de implementar sus *metodos abstractos*.
- Tambien se definen mediante la palabra reservada `abstract`.

Interfaces: Moticación

- En C#, una clase solo puede heredar de otra clase (El problema del diamante).
- Sin embargo, existen ocasiones en que dos objetos completamente diferentes comparten algunas operaciones:
 - Biblioteca y Bosque: ámbos son collecciones.
 - Vehiculo y Animal: ámbos tienen una posición y pueden moverse
- A menudo, queremos escribir un método que pueda utilizar ambos objetos, pero ¿que *firma* tendria el metodo?

Solución: Interfaces

- Definen un contrato, que debe cumplir toda clase que implemente la interfaz
- Son una lista de metodos y propiedades, sin definir cuerpo
- No se pueden crear instancias de interfaces, sin embargo, pueden utilizarse como tipo de propiedades y variables
- Se definen mediante la palabra reservada `interface`
- Una clase declara que implementa una interfaz mediante la palabra reservada `implements`

Interfaces: Aplicaciones

- Una herramienta que permite crear abstracciones poderosas.
- Permite escribir metodos que pueden utilizarse con una variedad de objetos.
- Permite crear bibliotecas, donde el programador simplemente debe implementar una serie de interfaces para poder utilizar todas las funciones.
- Permiten que un objeto imite a otro objeto para propósitos de evaluación o demostración.