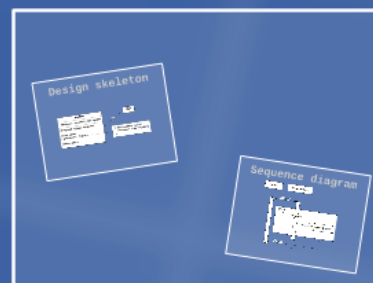


# How many AnimalFactory objects should we create?

Need only *one* instance of factory

Need *global access point* to instance

*Global* factory object?



## Singleton

Prevents *general* instantiation of class

Creates & manages *single* instance

Provides *global access point* to instance

AnimalFactory
- instance : AnimalFactory () static variable
- AnimalFactory() {} private constructor
if static method
+ getAnimalFactoryInstance() : AnimalFactory
+ createAnimal(String) : Animal



# Singleton

Prevents *general* instantiation of class

Creates & manages *single* instance

Provides *global access point* to instance

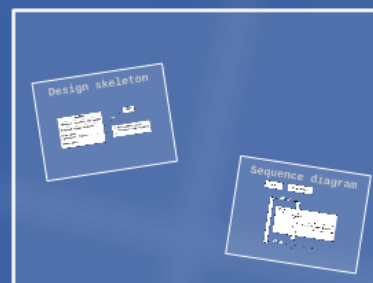
AnimalFactory
- instance : AnimalFactory // static variable
- AnimalFactory() // private constructor
// static method
+ getAnimalFactoryInstance() : AnimalFactory
+ createAnimal(String) : Animal

# How many AnimalFactory objects should we create?

Need only *one* instance of factory

Need *global access point* to instance

*Global* factory object?



## Singleton

Prevents *general* instantiation of class

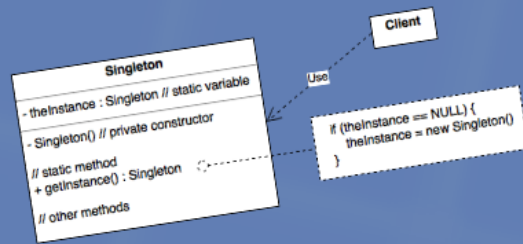
Creates & manages *single* instance

Provides *global access point* to instance

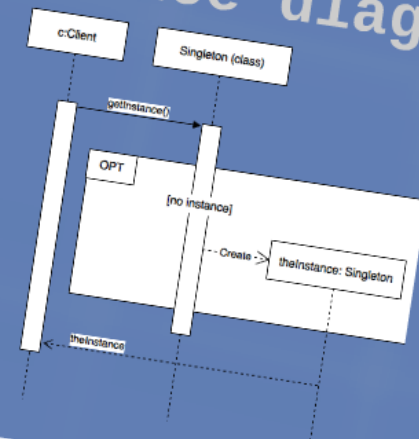
AnimalFactory
- instance : AnimalFactory () static variable
- AnimalFactory() {} private constructor
if static method
+ getAnimalFactoryInstance() : AnimalFactory
+ createAnimal(String) : Animal



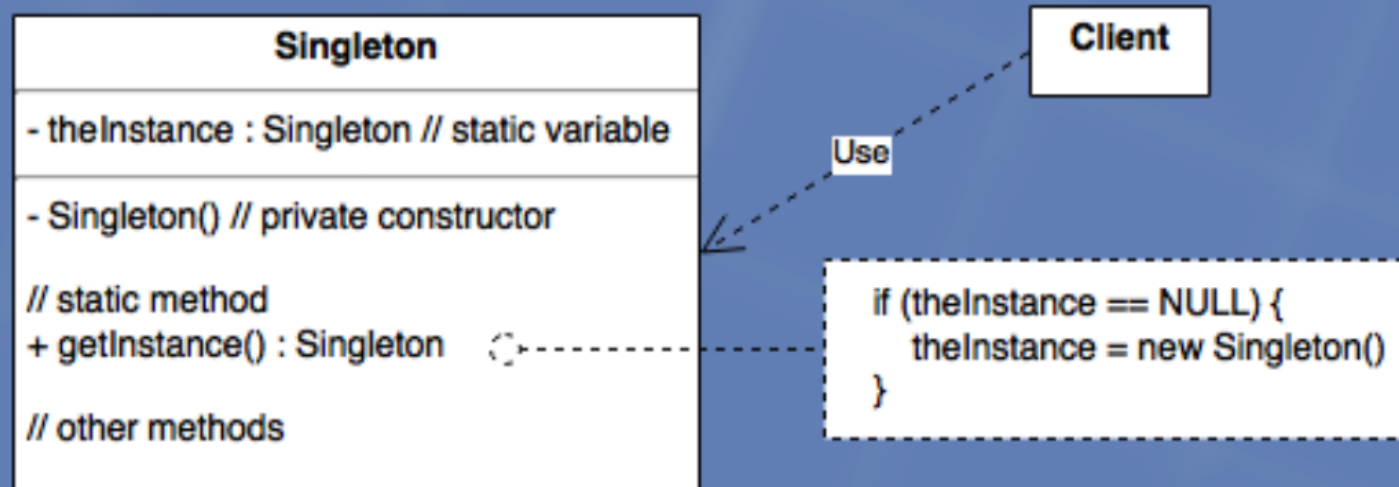
## Design skeleton



## Sequence diagram



# Design skeleton



# Sequence diagram

