Singleton

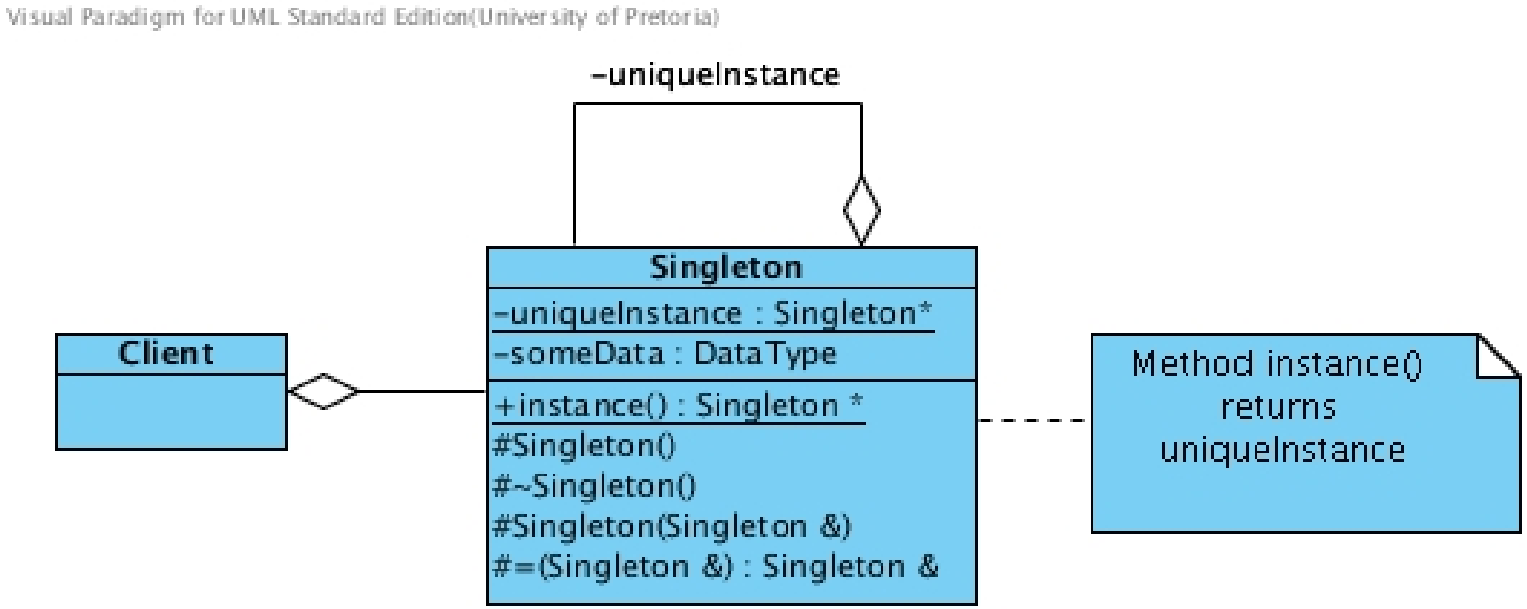
Linda Marshall and Vreda Pieterse

Department of Computer Science
University of Pretoria

29 October 2014

Name and Classification: Singleton (Object Creational)

Intent: "Ensure a class only has one instance, and provide a global point of access to it." GoF(127)



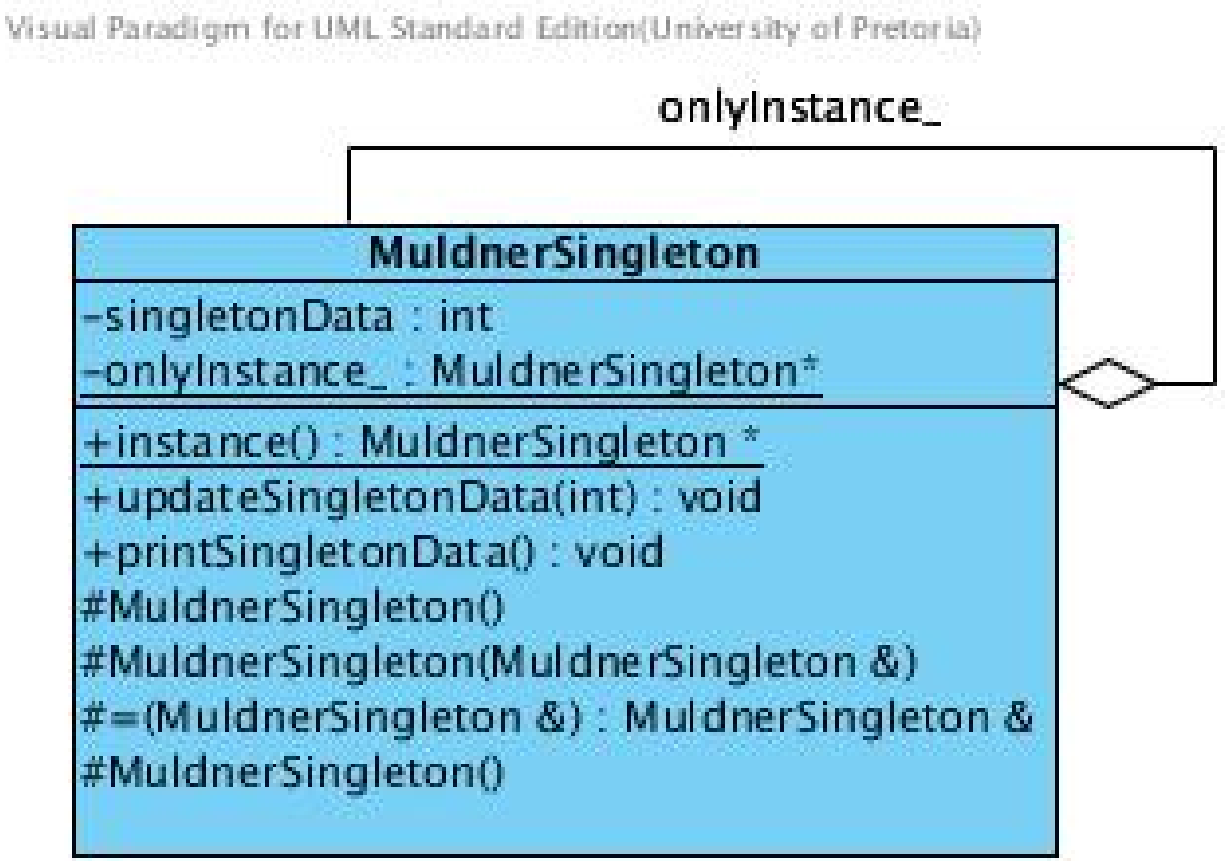
T Muldner, C++ Programming with Design Patterns Revealed, Addison-Wesley,2002

- Note the visibility of the constructor, Copy Constructor, Assignment operator and Destructor

Singleton

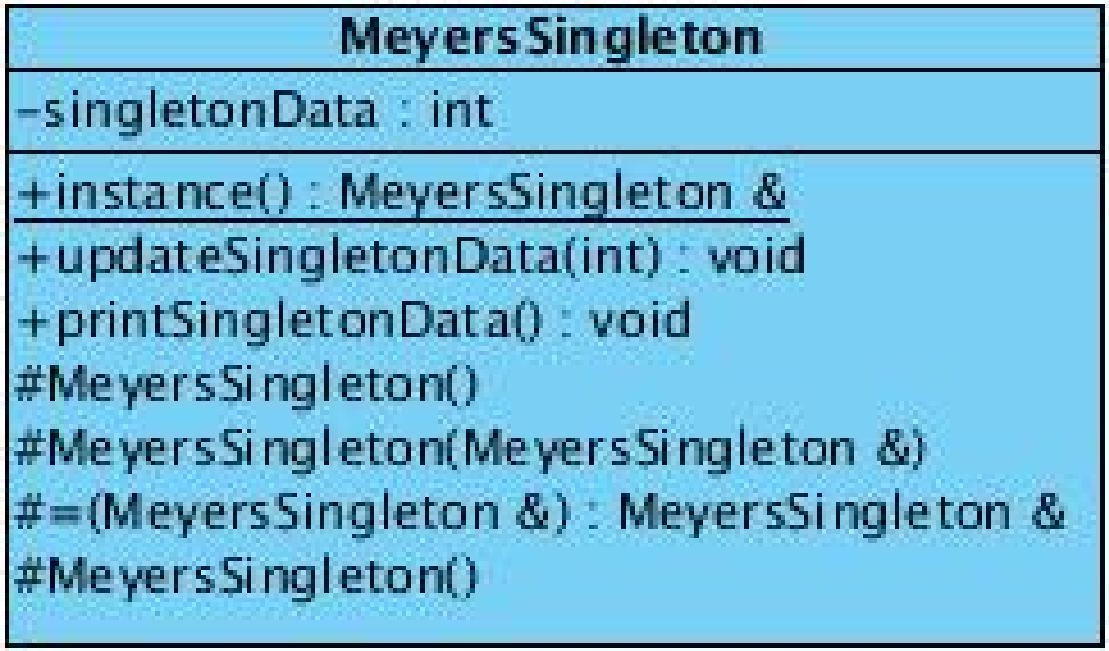
- defines an instance operation and ensures that the object is only constructable via this operation.

- Many patterns can be made a Singleton


Visual Paradigm for UML Standard Edition(University of Pretoria)
onlyInstance_
MuldnerSingleton
-singletonData : int
-onlyInstance_ : MuldnerSingleton*
+instance() : MuldnerSingleton *
+updateSingletonData(int) : void
+printSingletonData() : void
#MuldnerSingleton()
#MuldnerSingleton(MuldnerSingleton &)
#=(MuldnerSingleton &) : MuldnerSingleton &
#MuldnerSingleton()

- Where is the instance variable onlyInstance_ initialised?
- What about the memory leak?

Visual Paradigm for UML Standard Edition(University of Pretoria)



- creates a static Singleton object on the stack and returns a reference to the object
- the Singleton object is created when the instance method is called for the first time

- Why must the Constructor, Copy Constructor, Assignment operator and Destructor be protected or private?
- What are the implications of having the Constructor, Copy Constructor, Assignment operator and Destructor private vs protected?

- Would a virtual destructor in the Muldner example solve the memory leak problem?
- The Meyer solution solves the memory leak problem

- Do I know how many singletons have been requested?
- If not, how do I make sure I know?

- How do I separate concerns? .h and .cpp files!