Prototype ID

Student:

Izabela Kuźniar

Teacher:

Andrea Corradini

Course:

Software Design Patterns

# Name and category

Prototype is a creational pattern.

# Intent:

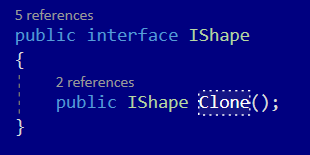
It creates new objects by coping/cloning other.

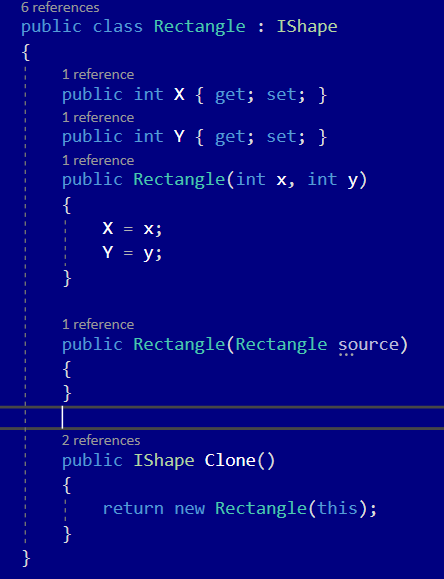
# Motivation:

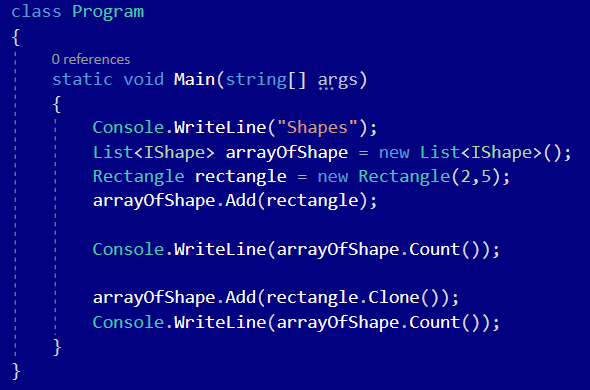
It is more efficient/fast to create a new object by cloning it

# Structure as a UML class diagram

# Implementation:







# Consequences:

Benefits:

* Cloning objects without coupling to their concrete classes
* Getting rid of repeated initialization code in favor of cloning pre-built prototypes
* Producing complex objects more conveniently
* An alternative to inheritance when dealing with configuration presets for complex objects

Drawbacks:

* Cloning complex objects that have circular references might be very tricky

# Known uses

* When the creation of the object is time consuming or complex
* When object creation should be decoupled from system behavior

# Related patterns

1. Prototype Pattern is evolved version of Factory Method Pattern.
2. Prototype Pattern can be used to compose the methods of Abstract Factory classes.
3. Prototype Pattern helps when you need to save copies of Commands into history.
4. Heavy use of Composite and Decorator can often benefit from using Prototype Pattern.
5. You can use Abstract Factory along with Bridge.
6. Prototype Pattern can be implemented as Singleton.