Patrón de creación Prototype



Estudiantes: Claudio Alfaro, Ignacio Taibo

Docente: Delia Ibacache.

Asignatura: CIF 6558 Ingeniería de software

Introducción

En este trabajo hablaremos sobre el patrón de diseño Prototype, cuando se usa, como se usa y la aplicación que este tiene en código.

Patrón Prototype

Este patrón se clasifica dentro de los patrones de creación, este tiene como finalidad crear objetos clonando instancias las cuales son creadas con anterioridad.

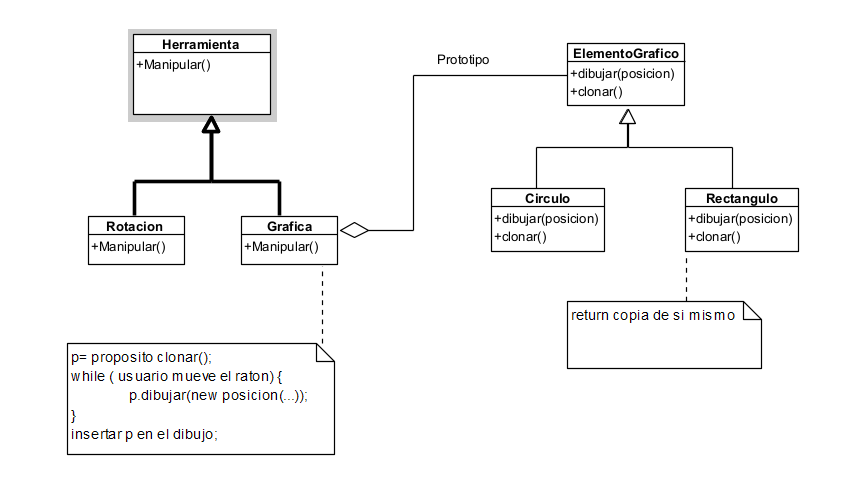
La idea de este patrón es usarlo cuando el proceso de crear un nuevo objeto sea complejo y costoso, esto independiente de si se requieren muchos pasos para crearlo, al usarlo debemos tener en cuenta de que nuestro código no dependa de clases concretas las cuales van a ser copiadas.

Esto tiene diferentes ventajas y desventajas las cuales son descritas a continuación:

Ventajas: Crea y elimina objetos en tiempo de ejecución, esto permite definir un comportamiento a través de la composición de objetos.

Desventajas: Es una manera muy fuerte de solucionar la creación de objetos costosos si no se crean muchos objetos de manera dinámica, oculta las dependencias, todas las clases deben implementar un método el cual se llama “CLONE ()” el cuál se usa para poder crear un objeto nuevo.

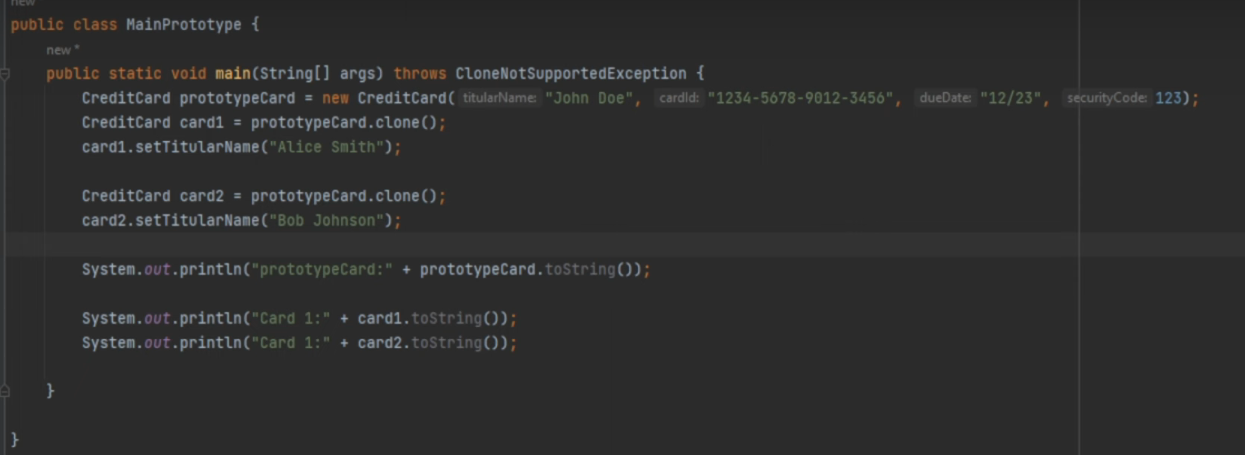
|  |  |
| --- | --- |
| Nombre | Prototype. |
| Clasificación | Patrón de creación. |
| Propósito | Su finalidad es crear objetos clonando instancias creadas con anterioridad. |
| Aplicabilidad | Este patrón se usa cuando el proceso de creación de un nuevo objeto es complejo y costoso, ya sea porque el objeto es complicado o se requieren muchos pasos para crearlo.  El criterio a usar es cuando nuestro código no dependa de las clases concretas de los objetos que vamos a copiar. |
| Estructura | Ver figura 1.1 |
| Consecuencias | Ventajas: Crea y elimina objetos en tiempo de ejecución, esto permite definir un nuevo comportamiento a través de la composición de objetos.  Desventajas: Es una manera muy fuerte de solucionar la creación de objetos costosos si no se crean muchos objetos de manera dinámica, oculta las dependencias, todas las clases tienen que implementar un método “CLONE ()” para poder crear un nuevo objeto |
| Implementación |  |

Figura 1.1

Implementación

Para la implementación de un patrón Prototype lo explicaremos con un ejemplo basado en tarjetas de crédito o débito.

Cada tarjeta tiene un número, fecha de vencimiento, nombre del titular y código de seguridad, el poder crear una tarjeta de crédito nueva puede llegar a ser costoso y con aprobación de múltiples sistemas, pero mucha información de cada tarjeta es similar, por lo que el patrón Prototype puede entrar en acción al clonar las tarjetas con información relativamente modificada.

A continuación, se muestra un código de ejemplo para el patrón y el ejemplo mencionado anteriormente.

