Metodologías de Programación I

Alumno: Orellana Maximiliano

Comisión 2.

*Práctica 3.*

*Patrones Factory Method y Observer*

*Respuestas:*

*Ejercicios 1:*

*Recordamos*: ¿Cuál es la única diferencia entre los métodos *llenar*, *llenarPersonas* y *llenarAlumnos* implementados en la práctica 1 (ejercicios 5, 12 y 16)? ¿Qué tuvo que hacer con el método *informar* (ejercicio 6) a medida que ejecutaba los métodos *main* (ejercicios 13 y 17)? ¿Qué sucedería con todos estos métodos si apareciera una nueva clase *Vendedor* la cual se desea comparar por cantidad de ventas realizadas?

Las diferencias entre los métodos *llenar*, *llenarPersonas* y *llenarAlumnos* implementados en la práctica 1 son que crean diferentes tipos de objetos.

Con el método *informar* a medida que ejecutaba los métodos *main tuve que cambiar el tipo de objeto que creaba.*

Si apareciera una nueva clase *Vendedor tendría que implementar el método informar para la clase Vendedor.*

*Ejercicios 7:*

*Para reflexionar*. ¿Qué debería hacer si se quiere tener en el método *main* la opción de almacenar los comparables en una pila, en una cola, en una colección múltiple, en un conjunto o en un diccionario?

Si se quiere tener en el mismo main la opción de almacenar los comparable en una pila, en una cola, en una colección múltiple, en un conjunto o en un diccionario debería implementar el patrón Factory Method para el tipo de colección, además, de implementar en main un menú de opciones para crear el coleccionable deseado.

*Ejercicios 10:*

*Para reflexionar*. ¿Qué tienen en común las fábricas de la clase *Vendedor* y de la clase *Alumno*? ¿Podría ampliarse la jerarquía de clases de las fábricas? ¿Cómo?

Las fábricas de la clase *Vendedor* y de la clase *Alumno* tienen en común que son subclases de *fabricaDeComparables* y que crean personas. Se podría ampliar la jerarquía, haciendo una clase *FabricaDePersonas* sub clase de *fabricaDeComparables* y superclase de las fabricas de *Vendedor* y de *Alumno.*