

Programación Orientada a Objetos

Proyecto de cursada

Integrantes:

Pedro Hoses.

Yamil Chaju.

E-mails:

pedrohoses@gmail.com

yamilchaju@gmail.com

Introducción

En el siguiente informe se describe como desarrollamos e implementamos el trabajo propuesto por la catedra y el porqué de las estructuras y/o métodos que utilizamos.

Análisis y conclusiones

- En la interfaz “Seguro” declaramos la variable “numPoliza” para que todos los seguros tengan un identificador único, sea un seguro compuesto, especial o simple.
- Como estructura para trabajar los conjuntos de seguros filtrados utilizamos ArrayList ya que nos simplificaba la utilización de métodos para “SeguroCompuesto”.
- Utilizamos el método “compareTo” para la comparación de atributos de “SeguroSimple” ya que nos simplificaba mucho a la hora de usar filtros.
- Utilizamos como estructura para “SeguroSimple” un hashtable dado que en este problema los atributos son dinámicos (“...la póliza con su tipo y **demás información que pueda ser relevante...**”).
- Agregamos la funcionalidad getCopia ya que a la hora de filtrar un seguro retornamos una copia de el por si el usuario desea modificarlo sin romper el original.
- En el método “getCosto” de “SeguroEspecial” retorna el monto de los seguros mas el monto especial solo si esta en una fecha valida, sino retorna un valor discernible.
- Dividimos los filtros en dos subclases, los filtros lógicos en el cual se encuentra el filtro OR y AND, y los filtros por características el cual contiene el filtro menor, mayor e igual.
- Cuando agregamos atributos a SeguroSimple exigimos que sean de tipo comparable para que a la hora de filtrarlos podamos usar la sentencia compareTo.