

TP5

Patrones de Comportamiento

1. Cree una clase bajo el patrón *cadena de responsabilidad* donde los números del 1 al 100 sean pasados a las clases subscriptas en secuencia, aquella que identifique la necesidad de consumir el número lo hará y caso contrario lo pasará al siguiente en la cadena. Implemente una clase que consuma números primos y otra números pares. Puede ocurrir que un número no sea consumido por ninguna clase en cuyo caso se marcará como no consumido.
2. Implemente una clase bajo el patrón *iterator* que almacene una cadena de caracteres y permita recorrerla en sentido directo y reverso.
3. Implemente una clase bajo el patrón *observer* donde una serie de clases están subscriptas, cada clase espera que su propio ID (una secuencia arbitraria de 4 caracteres) sea expuesta y emitirá un mensaje cuando el ID emitido y el propio coinciden. Implemente 4 clases de tal manera que cada una tenga un ID específico. Emita 8 ID asegurándose que al menos cuatro de ellos coincidan con ID para el que tenga una clase implementada.
4. Modifique el programa *IS2_taller_scanner.py* para que además la secuencia de barrido de radios que tiene incluya la sintonía de una serie de frecuencias memorizadas tanto de AM como de FM. Las frecuencias estarán etiquetadas como M1, M2, M3 y M4. Cada memoria podrá corresponder a una radio de AM o de FM en sus respectivas frecuencias específicas. En cada ciclo de barrido se barrerán las cuatro memorias.
5. Modifique el programa *IS2_taller_memory.py* para que la clase tenga la capacidad de almacenar hasta 4 estados en el pasado y pueda recuperar los mismos en cualquier orden de ser necesario. El método *undo* deberá tener un argumento adicional indicando si se desea recuperar el inmediato anterior (0) y los anteriores a el (1,2,3).