**TP1- Repaso y entrada en calor**

1. **Evaluación de protocolos de una clase**

**Opción 2) Por que usa un sistema encapsulado, el cliente no puede observar la estructura interna.**

1. **Delegación**

**Opción 2)**

1. **Polimorfismo**

**Opción 1)**

**No tiene polimorfismo, y al agregar o generar otro tipo de cuenta tendría que modificar la superclase, cosa que no haría falta hacer si tendría jerarquía polimórfica.**

**Opción 2)**

**Al igual que en la opción 1, no tiene polimorfismo.**

**Opción 3)**

**Levantar lógica. Generaría un método abstracto en la clase padre que se utilice en el método extraer, porque los métodos terminan retornando lo mismo, en el método abstracto que generaría cada uno redefiniría la condición.**

**Opción 4)**

**No encuentro ningún defecto, sería la solución o mejora de la opción anterior.**

**Actividad de Lectura #1**

1. **Significa que yo en un método que este implementando en la misma clase de las variables puedo acceder a ellas de manera directa haciendo una auto referencia “this” y llamando a la variable. Por ejemplo: “this.nombreClase”.**
2. **Para acceder a la variable de forma indirecta ya sea para obtener o modificar utilizo getters o setters. Por ejemplo para obtener this.getNombreClase().**

**Actividad de Lectura #2**

**Cuando instancias un objeto**

**Actividad de Lectura #3**

**Se debe proporcionar acceso a las variables mediante métodos. Utilizando el método de Indirect Variable Access, se debe además asegurar que los métodos para poder obtener dichas variables de las colecciones estén protegidos para evitar que el cliente cause resultados indeseables.**

**Actividad de Lectura #4**

**Es necesario por que mejora la comunicación del código resultante. Con estos dos métodos es menor la exposición del estado.**