Deep vs Shallow Copy

- **Shallow Copy**

```
public class Company implements Cloneable {
   private String name;
   private List<Employee> employees = new ArrayList<>();

   public Object clone() {
      try {
         Company c = (Company) super.clone();
         return c;
      } catch (CloneNotSupportedException e) {
         throw new InternalError(e.getMessage());
      }
   }
}
```

Deep vs Shallow Copy

- **Level1-Deep-Copy**

```
public class Company implements Cloneable {
   private String name;
   private List<Employee> employees = new ArrayList<>();

   public Object clone() {
      try {
         Company c = (Company) super.clone();
         c.employees = new ArrayList<>(employees);
         return c;
      } catch (CloneNotSupportedException e) {
         throw new InternalError(e.getMessage());
      }
   }
}
```

Deep vs Shallow Copy

- **Level2-Deep-Copy**

```
public class Company implements Cloneable {
   private String name;
   private List<Employee> employees = new ArrayList<>();

   public Object clone() {
      try {
         Company c = (Company) super.clone();
         c.employees = new ArrayList<>();
         for(Employee e : employees)
            c.employees.add(e.clone());
         return c;
      } catch (CloneNotSupportedException e) {
         throw new InternalError(e.getMessage());
      }
   }
}
```

Deep vs Shallow Copy

- **Level2-Deep-Copy (cont)**

```
public class Employee implements Cloneable {
   private String name;
   private int yearOfBirth;

   public Object clone() {
      try {
         return super.clone();
      } catch (CloneNotSupportedException e) {
         throw new InternalError(e.getMessage());
      }
   }
}
```