

Informe Ejercicio 2

PRINCIPIOS DE DISEÑO USADOS

En este ejercicio se están cumpliendo los siguientes principios:

Por un lado, el principio de inversión de la dependencia, ya que al hacer que las clases Equipo y Trabajador hereden de la clase abstracta ComponenteProyecto, y que por tanto estas implementen los métodos abstractos, podremos usar ambas sin causar ningún conflicto de tipos en el programa.

Por otra parte, tenemos el principio de Responsabilidad Única, pues al invocar los métodos de la clase ComponenteProyecto, las responsabilidades se van delegando en los objetos concretos, sobre todo a causa de las llamadas recursivas de métodos tales como totalSalary y totalHours.

Luego tendríamos el principio de “favorece la inmutabilidad” en métodos como getTeams de la clase Proyecto, al devolver su lista de equipos como un objeto no modificable, impidiendo que se pueda alterar su estado interno.

Finalmente encontramos el principio abierto-cerrado, de nuevo en la clase abstracta ComponenteProyecto, porque permitimos su fácil extensión sin comprometer su estado interno a través de modificaciones.

PATRÓN DE DISEÑO EMPLEADO

En este caso se ha utilizado el patrón Composición, ya que tanto Equipo como Trabajadores representan estructuras todo-parte (un equipo tiene varios trabajadores e incluso otros equipos), por lo que el tratamiento de las mismas es mucho más uniforme (al tratar ambas de la misma forma, pese a ser clases diferentes) y se pueden incluir fácilmente otras estructuras en la jerarquía.

Los diagramas (tanto estático como dinámicos) correspondientes a este patrón se incluyen aparte en la carpeta asignada, con el fin de ofrecer una visualización más cómoda de dichos diagramas. Además, cabe aclarar que en los diagramas dinámicos se representan algunos de

los métodos principales del programa.