

## **-DESIGN PATTERN(S)-**

I selected adapter design pattern in this project to solve my matters about enemy robots.

**Why** did i choosed adapter pattern?Because adapter pattern works as a bridge between two incompatible interfaces.This pattern involves a single class which is responsible to join functionalities of independent or incompatible interfaces.Both EnemyRobot and EnemyTank are enemies.They do some familiar things but also some things which are totally different.So they are incompatible.I accept them all enemy as i said before so i do not want to know about tons of methods to call.I want to use just enemy methods for both instances.

**How** did i employ the adapter pattern?First of all,i created an interface which named as EnemyAttacker(Target).I assumed there are methods which i expect from enemy attackers.I wrote them in target interface.Then i created EnemyRobot(Adaptee) and EnemyTank classes.Robots and tanks are working differently.Enemy tank class is implemented from EnemyAttacker.But Enemy robot was kind of independent class.So i should made it adapted as enemy attacker.To adapt enemy robot as enemy attacker i created a adapter class which name is EnemyRobotAdapter and made it adapted here.In the final step,i created main class to test my code.