stackoverflow.com

what is the difference between loose coupling and tight coupling in object oriented paradigm?

1 min read • original

Tight coupling is when a group of classes are highly dependent on one another.

This scenario arises when a class assumes too many responsibilities, or when one concern is spread over many classes rather than having its own class.

Loose coupling is achieved by means of a design that promotes single-responsibility and separation of concerns.

A loosely-coupled class can be consumed and tested independently of other (concrete) classes.

Interfaces are a powerful tool to use for decoupling. Classes can communicate through interfaces rather than other concrete classes, and any class can be on the other end of that communication simply by implementing the interface.

Example of tight coupling:

```
 class CustomerRepository
{
    private readonly Database database;

    public CustomerRepository(Database database)
    {
        this.database = database;
    }
    public void Add(string CustomerName)
    {
        database.AddRow("Customer", CustomerName);
    }
}

class Database
{
    public void AddRow(string Table, string Value)
    {
    }
}
```

Example of loose coupling:

```
 class CustomerRepository
{
    private readonly IDatabase database;

    public CustomerRepository(IDatabase database)
    {
        this.database = database;
    }

    public void Add(string CustomerName)
    {
        database.AddRow("Customer", CustomerName);
    }
}

interface IDatabase
{
    void AddRow(string Table, string Value);
}

class Database : IDatabase
{
    public void AddRow(string Table, string Value)
    {
    }
}
```

Another example here.

---

http://stackoverflow.com/questions/2832017/what-is-the-difference-between-loose-coupling-and-tight-coupling-in-object-orien/2832047#2832047