LISKOV SUBSTITUDE PRINCIPLE

Java Clean code

Outline

Lesson 6.

The Single Responsibility Principle

Lesson 7.

The Open Closed Principle

Lesson 8.

The Liskov Substitution Principle

Lesson 9.

The Interface Segregation Principle

Lesson 10.

The Dependency Inversion Principle

Any fool can write code that a **computer can understand.** *Good programmers* **write code** *that humans* **can understand.**
Martin Fowler

LSP written by Barbara Liskov 1988

in a computer program, if S is a subtype of T, then objects of type T may be replaced with objects of type S (i.e., objects of type S may substitute objects of type T) without altering any of the desirable properties of that program (correctness, task performed, etc.)

LSP

- Subclass objects can replace superclass objects

- Methods should have same arguments

- Same return type

- Subclass can never do less than the parent class but can do more.

Scope

Variable scope length and name length should

be proportional

- The longer the scope the longer the length

- Opposite with methods and classes

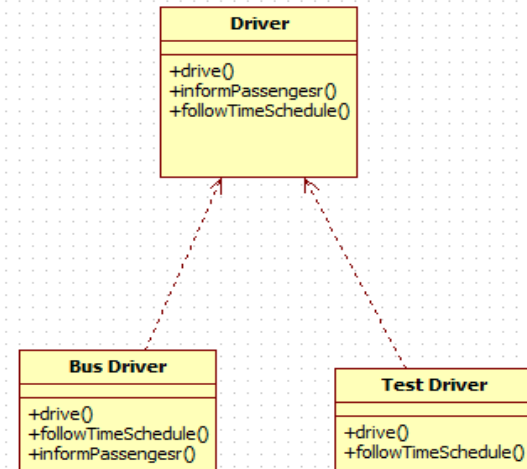- Private methods tend to be longer and more

  descriptive

Check List

- **No new exception should be thrown in derived class**

- **Pre-conditions cannot be strengthened**

- **Post-conditions cannot be weakened**

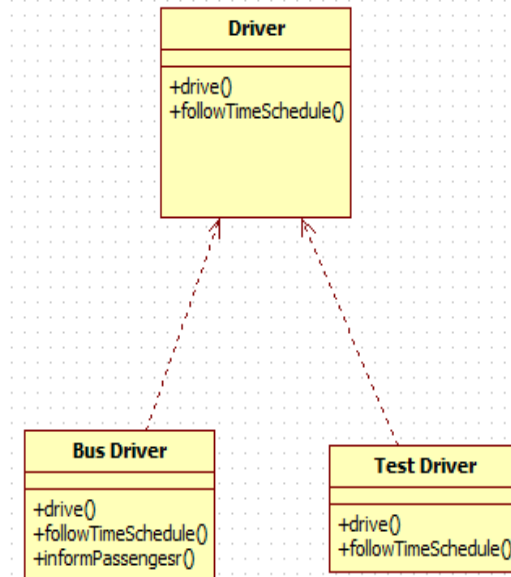- **Invariants must be preserved**

- **History constraint**

Example

Example

Driver

+drive()
+followTimeSchedule()

Bus Driver

+drive()
+followTimeSchedule()
+informPassengesr()

Test Driver

+drive()
+followTimeSchedule()

```php
interface Database
{
    public function selectQuery(string $sql): array;
}

class SQLiteDatabase implements Database
{
    public function selectQuery(string $sql): array
    {
        // sqlite specific code

        return $result;
    }
}

class MySQLDatabase implements Database
{
    public function selectQuery(string $sql): array
    {
        // mysql specific code

        return $result;
    }
}
```

Real world example

Behavior remains unchanged regardless of the implementation we choose to use.

```php
interface Database
{
    public function selectQuery(string $sql, int number): array; // This violates LSP !
}

class SQLiteDatabase implements Database
{
    public function selectQuery(string $sql, int number): array
    {
        // sqlite specific code

        return $result;
    }
}

class MySQLDatabase implements Database
{
    public function selectQuery(string $sql): array
    {
        // mysql specific code

        return $result;
    }
}
```

Good example gone bad

Now the subtypes cannot be used the same way since they don't produce the same result anymore.

Liskov substitute Principle Summary

- Subclass objects can replace superclass objects

- Arguments && return type

- Subclass can never do less than the parent class but can do more.

Course Progress

Lesson 6	Lesson 7	Lesson 8	Lesson 9	Lesson 10
The Single Responsibility Principle	The Open Closed Principle	The Liskov Substitution Principle	The Interface Segregation Principle	The Dependency Inversion Principle

THANK YOU!

- https://drive.google.com/file/d/0BwhCYaYDn8EgNzAzZjA5ZmItNjU3NS00MzQ5LTkwYjMtMDJhNDU5ZTM0MTlh/view