



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційні системи та технології

## Лабораторна робота №4

# Технології розроблення програмного забезпечення

*Вступ до паттернів проектування. Розібратися в особливостях використання та застосувати в проекті паттерн State.*

Варіант 12 (CI server)

Виконав студент групи ІА-13:

Матусяк М.Ю.

Перевірив:

Мягкий М.Ю.

Київ 2023

**Мета роботи:** Розібратися в особливостях використання та застосувати в проекті паттерн State.

### Хід роботи:

За завданням необхідно реалізувати шаблон State. Цей шаблон дозволяє змінювати логіку роботи об'єктів у випадку зміни їх внутрішнього стану.

Реалізація даного шаблону полягає в наступному: пов'язані зі станом поля, властивості, методи і дії виносяться в окремий загальний інтерфейс(BuildState):

```
interface BuildState {  
    void execute(CIContainerMediator mediator);  
  
    void visit(Visitor v);  
}
```

Кожен стан являє собою окремий клас (RunCompile, RunTest, RunCompileAndTest), які реалізують загальний інтерфейс BuildState:

Клас RunCompile визначає стан проекту при якому він буде тільки компілювати вхідний проект.

```
class RunCompile implements BuildState {  
    Project project;  
  
    public RunCompile(Project project) {this.project = project;}  
  
    @Override  
    public void execute(CIContainerMediator mediator) {  
        mediator.executeCompileCommand(project);  
    }  
  
    @Override  
    public void visit(Visitor v) {v.visit(project);}  
}
```

Клас RunTest визначає стан проекту при якому він буде тільки тестувати вхідний проект.

```
class RunTest implements BuildState {  
    Project project;  
  
    public RunTest(Project project) {this.project = project;}  
  
    @Override  
    public void execute(CIContainerMediator mediator) {  
        mediator.executeTestCommand(project);  
    }  
}
```

```
@Override
public void visit(Visitor v) {v.visit(project);}
}
```

Клас RunCompileAndTest визначає стан проекту при якому він буде компілювати та тестувати вхідний проект.

```
class RunCompileAndTest implements BuildState {
    Project project;

    public RunCompileAndTest(Project project) {this.project = project;}

    @Override
    public void execute(CIContainerMediator mediator) {
        new CompileAndTestDecorator(new RunCompile(project))
            .execute(mediator);
    }

    @Override
    public void visit(Visitor v) {v.visit(project);}
}
```

В кожному класі стану є основний метод execute() який визначає поведінку.

**Висновок:** на цій лабораторній роботі я познайомився з шаблоном State, засвоїв знання на практиці, продовжив розробку проекту.