

ChipWhisperer의 설계 패턴

IT융합공학부 권혁동

Contents

설계 패턴이란

CW의 설계 패턴

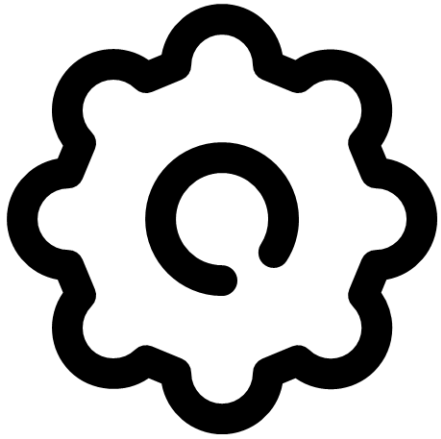


CryptoCraft LAB

설계 패턴이란

- 영어로는 디자인 패턴(Design Pattern)이라 칭함
- 소프트웨어를 작성할 때 ...
 - 자주 마주하는 **상황**
 - 비슷한 유형의 **문제**
- 여러 상황에서 여러 번 **반복 사용**할 수 있는 **솔루션**

설계 패턴이란



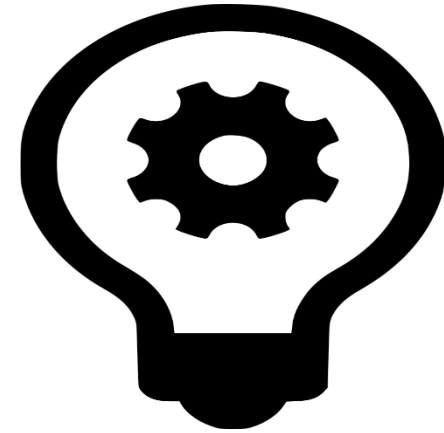
Context

문제가 발생할 수 있는 **상황**



Problem

패턴이 적용되어야 하는 **이슈**



Solution

문제를 해결할 수 있는 **방안**

설계 패턴이란

- 생성 패턴(Creational)
- 객체를 **생성**하는데 관련된 패턴
- **캡슐화**를 사용
- 객체가 생성, 수정되어도 **프로그램 구조에 영향이 없음**

추상 팩토리 / 빌더 / 팩토리 메소드 / 프로토타입 / 싱글턴

설계 패턴이란

- 구조 패턴(Structural)
- 클래스나 객체를 **조합**
- **거대한 구조를 형성**하는데 사용

어댑터 / 브리지 / 컴퍼짓트 / 데코레이터

퍼사드 / 플라이웨이트 / 프록시

설계 패턴이란

- 행위 패턴(Behavioral)
- 객체나 클래스의 알고리즘
- 책임 분배
- 업무를 분배하면서 객체간 결합도를 최소화

책임 연쇄 / 커맨드 / 인터프리터 / 이터레이터 / 미디에이터 / 메멘토
옵저버 / 테이트 / 스트래티지 / 템플릿 메소드 / 비지터

설계 패턴이란

- 타인이 작성한 코드는 이해하기 어려움
 - 다수의 **협력 코드**
 - 기존 **전임자의 코드**
- 특별한 문제에 대해서 **유연한 해결책**을 제공
- 협업 시에 **의사소통** 수단
- 무작정 패턴을 적용하는 것은 비효율적
 - 패턴이 문제에 적합한지 확인
 - 클래스와 객체 구성을 확인

CW의 설계 패턴

```
class CWCaptureGUI(CWMainGUI):  
    self.api.sigNewInputData.connect(self.newTargetData)  
    self.api.sigConnectStatus.connect(self.connectStatusChanged)  
    self.api.sigTraceDone.connect(self.glitchMonitor.traceDone)  
    self.api.sigCampaignStart.connect(self.glitchMonitor.campaignStart)  
    self.api.sigCampaignDone.connect(self.glitchMonitor.campaignDone)
```


• 옵저버 패턴

- 특정 객체의 **상태 변화**를 감지
- 상태 변화에 대응할 수 있도록 **의존 관계 형성**

CW의 설계 패턴

```
class ModelsBase(Parameterized):
```

```
    def leakage(self, pt, ct, guess, bnum, state):  
        pass
```



```
class AESLeakageHelper(object):
```

```
    def leakage(self, pt, ct, key, bnum):  
        raise NotImplementedError("ASKLeakageHelper does not implement leakage")
```

• 템플릿 메소드 패턴

- 템플릿을 제공하되, **메소드만 정의**하도록 함
- 큰 구조는 동일하지만 **세부적인 동작이 다른** 경우
- 상속받는 하위 클래스에서 기능을 구현

CW의 설계 패턴

```
self.newAct = QAction(QIcon('new.png'), '&New', self, shortcut=QKeySequence.New,  
                      statusTip='Create new Project', triggered=self.newProject)  
self.fileMenu.addAction(self.newAct)  
self.openAct = QAction(QIcon('open.png'), '&Open...', self, shortcut=QKeySequence.Open,  
                       statusTip='Open Project File', triggered=self.openProject)
```

```
def newProject(self):  
    self.okToContinue()  
    self.api.newProject()  
    logging.info("New Project Created")
```

```
def openProject(self, fname = None):  
    if not self.okToContinue():  
        return  
    if fname is None:
```

• 스트래티지 패턴

- 행위, 동작을 캡슐화
- 한가지의 **같은 문제**를 **다양한 방식**으로 해결
- 전략을 변경
 - 수행 방식, 규칙, 문제 해결 방법 ...