

S M A _ T 6

1To10 _ CPT Tool

OOPT Demonstaration

One to Ten

CPT 한번에 끝내기



didix@naver.com



010-9633-5320



trello.com//software-modelling

| 손성호 이광제 이용주 황준익

Table of Contents

01 Respond to Static analysis

02 OOPT Review

03 Summary



01

Respond to Static analysis

01 Respond to Static analysis

“ 두 조 모두 PMD, Eclipse Metrics, FindBugs를 이용 했기에
공통적인 문제가 검출되었다.”

01 Respond to Static analysis

PMD

➤➤ 중복된 코드가 많음, 일부 변수 형식의 불일치

```
JLabel lblNewLabel_1 = new JLabel("Select");
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setFont(new Font("", Font.BOLD, 20));
lblNewLabel_1.setForeground(Color.WHITE);
lblNewLabel_1.setOpaque(true);
lblNewLabel_1.setBackground(Icon.makeColorRGBA(64,64,64,255));
lblNewLabel_1.setBounds(319, 49, 105, 25);
this.add(lblNewLabel_1);
```

01 Respond to Static analysis

Metrics

➤➤ 일부 Package에 대한 Cyclomatic Complexity가 높다.

개발자가 재귀보다 루프나 스위치를 좋아하여 블록이 꽤 많았다.

➡ 삼항 연산자 등을 활용하여 코드의 복잡도를 줄이고자 노력
But 알고리즘 상의 Loop이므로 많은 개선은 힘들다.

01 Respond to Static analysis

FindBugs

➤➤ 파일 입출력 관련 null Pointer 처리 확인 요망

```
finally
{
    if(put != null) put.close();
}
finally
{
    if(input != null) input.close();
}
```

파일 입출력에서 예외처리 후 Reader를 close하는 과정에서 null 확인 후 close

01 Respond to Static analysis

FindBugs

➤➤ Calculation과정에서 쓰이지 않는 지역변수

```
public ArrayList<CPT.Category> sort(){  
  
    resultList.clear();  
    resultList_filtered.clear();  
  
    ArrayList<CPT.Property> temporaryPropertyList = new ArrayList<CPT.Property>();  
    ArrayList<CPT.Category> tempCategoryList = new ArrayList<CPT.Category>();  
    //ArrayList<CPT.RepValue> tempRepvList = new ArrayList<CPT.RepValue>();  
}
```

Calculation 과정 중에 사용하지 않는 tempRepvList 지역변수를 제거

01 Respond to Static analysis

FindBugs

➤➤ clone()사용에서의 경고

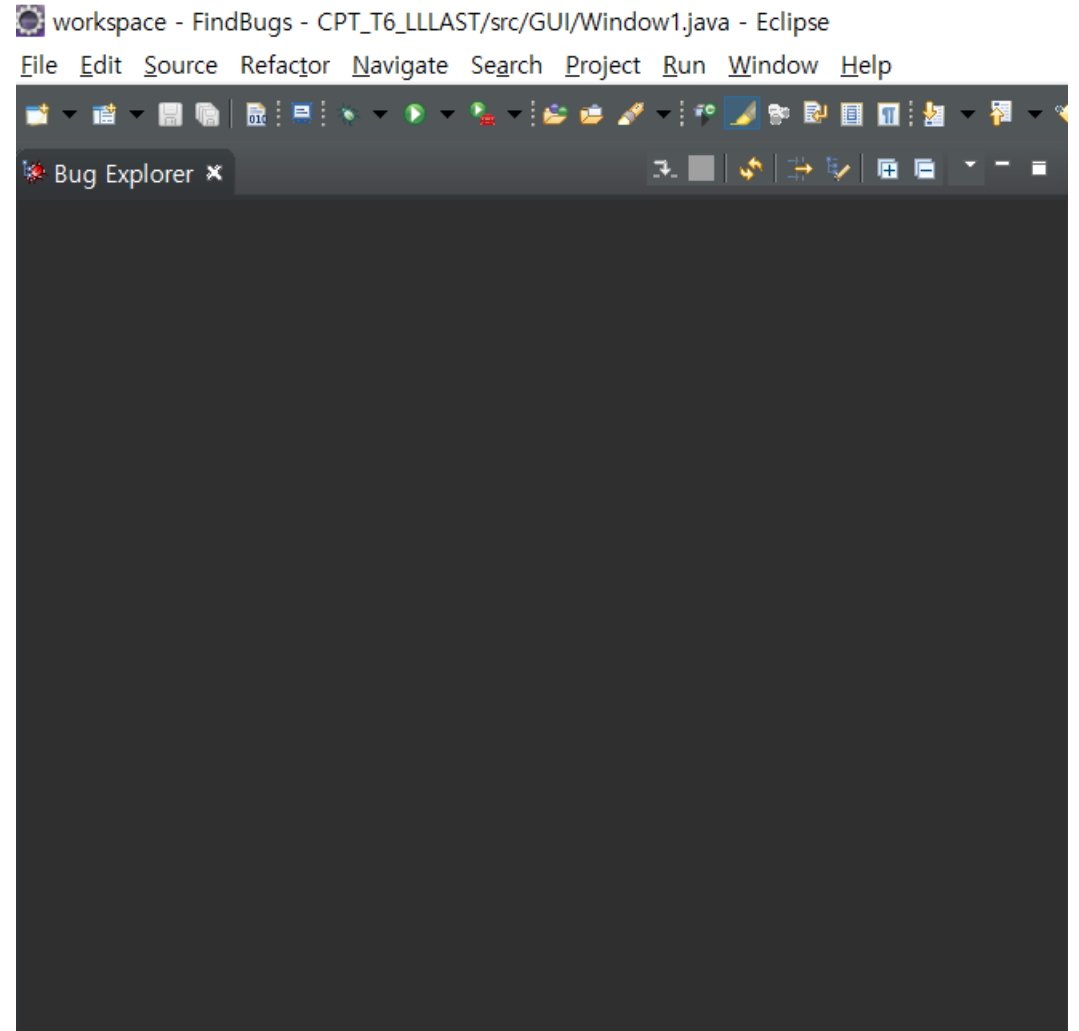
```
public class Result implements Cloneable{  
  
    @Override  
    public Result clone()  
}
```

Result의 clone의 깊은 복사를 위한 인터페이스와 오버라이딩 메소드 구현

01 Respond to Static analysis

FindBugs

위험요인을 모두 수정하여 FindBugs
에서 위험요소가 검출되지 않는다.



02

OOPT Review

02 OOPT Review

Advantage

- ▶▶ 단계별 프로젝트 설계를 하며, iterative하게 진행한다.
Waterfall 모델과 달리 한 번에 모든 사항에 대한 결정을 필요로 하지 않음
 - ▶ 부담이 덜하다.
- ▶▶ 현재 단계에서의 이슈에만 집중해서 프로젝트를 진행.
다음 단계에서의 이슈에 대한 고려를 배제한다.
 - ▶ 단계별 과정을 단순화 가능

02 OOPT Review

Advantage

- ▶▶ 각 단계별로 세부적인 사항을 고정하지 않고 개발 과정을 진행해서 고정된 내 용으로 프로젝트를 진행하지 않고 자유로운 프로젝트 진행이 가능했다.
- ▶▶ 문서를 통해 필요한 정보에 대한 즉각적인 확인이 가능했다.
- ▶▶ 단계별로 프로젝트를 진행하다 보니, 초기에 계획한 대로 프로그램을 완성 할 수 있었다.

02 OOPT Review

Disadvantage

- ▶▶ 단계별로 작성하는 문서로 인해, 문서 작성에 많은 시간이 소요.
- ▶▶ 뒤쪽 단계에서 수정사항이 발생 시 앞쪽에서 작성했던 문서를 모두 수정해야 하는 사태가 발생
- ▶▶ 이번 프로젝트의 경우 규모가 그렇게 크지 않아서 단계별로 나누어 문서를 작성하고 프로세스를 진행하는데 오히려 시간이 더 많이 소요되는 단점이 있었다.

03

Summary

03 Summary

이광제

OOPT를 하면서 가장 크게 느꼈던 점은,
문서가 귀찮아도 내가 생각했던 걸 저장한다는 점에서
굉장히 이롭다는 것을 느꼈다.

03 Summary

손성호

문서를 단계별로 작성하니까, 나중에 모르는 부분이나 기존의 내용과 다른 부분이 있는 경우 **문서를 확인할 수 있기 때문에 프로젝트를 좀 더 효율적으로 진행**할 수 있었다.

또한, 각 단계별로 **다음 단계에 대한 생각을 배제**하고 프로세스를 진행하니까 좀 더 **생각이 간결**해지는 것 같았다.

03 Summary

황준익

문서를 작성하는 데 어려움이 많았고, 프로세스 단계를 진행함에 따라 문서의 수정이 너무 많아서 힘든 부분이 있었지만, 프로그램 개발 단계에 접어들면서부터 문서에 기록했던 내용들이 프로그램에 적용되는 것을 보면서 **단계별로 문서를 작성하는 것이 큰 도움이 됨**을 느꼈다.

03 Summary

이용주

프로그램을 만들 때, 어떤 개발 프로세스를 사용 해 프로젝트를 진행하느냐에 따라서 개발과정이 많이 다를 수 있다는 것을 알 수 있었습니다.

다음 프로젝트를 진행한다면 가장 알맞은 진행 프로세스를 사용하도록 노력하겠습니다.

Q & A