

소프트웨어공학 개론 6조

ECoder Passport

노기정, 박주환, 신하영, 이원혁, 정대교





Table of contents

01

System Structure

02

Implementation

03

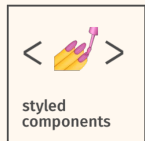
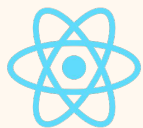
Green Pattern

04

Project Management

System Structure: **Tech Stack**

Frontend



A X I O S

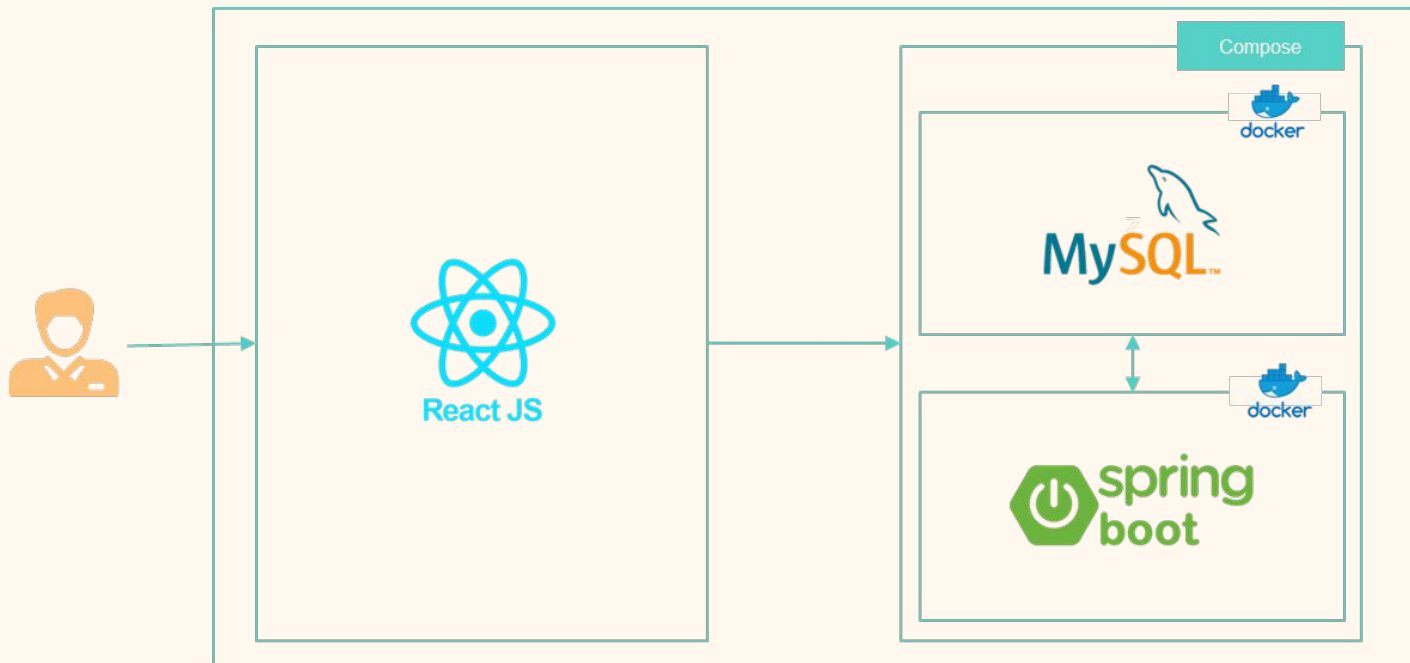
Backend



Deploy



System Structure: **System Architecture**



Implementation: Code Editor

자바 코드 전달 방식 선택

코드 에디터 | Git Repo

ECoder Passport

JAVA 코드만 입력해 주세요

제출

```
1 import java.util.Scanner;
2
3 public class SumOfTwoNumbers {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         // Prompt the user to enter the first number
8         System.out.print("Enter the first number: ");
9         int num1 = scanner.nextInt();
10
11        // Prompt the user to enter the second number
12        System.out.print("Enter the second number: ");
13        int num2 = scanner.nextInt();
14
15        // Calculate the sum of the two numbers
16        int sum = num1 + num2;
17
18        // Display the result
19        System.out.println("The sum of " + num1 + " and " + num2 + " is " + sum);
20
21        // Close the scanner
22        scanner.close();
23    }
24 }
```

에디터에 입력된 자바코드
Byte 실시간 확인

서버 실행 정보

CPU 정보: AMD 라이젠 12382X12/16코어 가용 메모리 크기: 00GB 데이터 센터의 에너지 효율성: 1.5

Editor: 0.74KB

서버정보

Implementation: Code Editor




제출 버튼 클릭

#1: 자바 코드 길이 제한


0 이상 64KB 이하의 자바코드만 입력가능합니다.

제출

 *ECoder Passport*

코드 에디터 | Git Repo

...



```
scanner.close();
```

```
java.lang.NullPointerException
```

#2: 서버 실행중 로딩 표시

#3: 자바코드 런타임 혹은 컴파일 에러

Implementation: Code Editor

실행 완료: 기존 코드와 그린화 패턴 적용 코드 비교

기존코드/그린코드화 패턴 적용 코드

코드다시 입력하기

```
1 import java.util.Scanner;
2
3 public class SumOfTwoNumbers {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         // Prompt the user to enter the first number
8         System.out.print("Enter the first number: ");
9         int num1 = scanner.nextInt();
10
11        // Prompt the user to enter the second number
12        System.out.print("Enter the second number: ");
13        int num2 = scanner.nextInt();
14
15        // Calculate the sum of the two numbers
16        int sum = num1 + num2;
17
18        // Display the result
19        System.out.println("The sum of " + num1 + " and " + num2 + " is " + sum);
20
21        // Close the scanner
22        scanner.close();
23    }
24 }
```

```
1 import java.util.Scanner;
2
3 public class ProductOfTwoNumbers {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         // Prompt the user to enter the first number
8         System.out.print("Enter the first number: ");
9         int num1 = scanner.nextInt();
10
11        // Prompt the user to enter the second number
12        System.out.print("Enter the second number: ");
13        int num2 = scanner.nextInt();
14
15        // Calculate the product of the two numbers
16        int product = num1 * num2;
17
18        // Display the result
19        System.out.println("The product of " + num1 + " and " + num2 + " is " + product);
20
21        // Close the scanner
22        scanner.close();
23    }
24 }
```

서버정보

CPU 정보: AMD 라이젠 12382X12/16코어 가용 메모리 크기: 00GB 데이터 센터의 에너지 효율성: 1.5

Implementation: Code Editor

실행 완료: 기존 코드와 그린화 패턴 적용 코드 실행 결과 확인

코드 실행 결과

기존 코드

Original Code Output

Hello, World!

실행 시간: 0.1s CPU 사용 비율: 0.1% CPU 전력량: 0.1W

탄소 배출량 gCO2e 전력 소모량 kWh

392.90

0.20

비행 거리 gCO2e/km 기차 kWh/km



30.10



234.10

넷플릭스 kWh 구글 검색 kWh



0.10



0.10

그린화 패턴 적용 코드

Green Pattern Applied Code Output

Hello, World!
Hello, World!

실행 시간: 0.1s CPU 사용 비율: 0.1% CPU 전력량: 0.1W

탄소 배출량 gCO2e 전력 소모량 kWh

392.90

0.20

비행 거리 gCO2e/km 기차 kWh/km



0.10



0.10

넷플릭스 kWh 구글 검색 kWh



0.10



0.10

비교 지표

Implementation: **GitHub Repository**

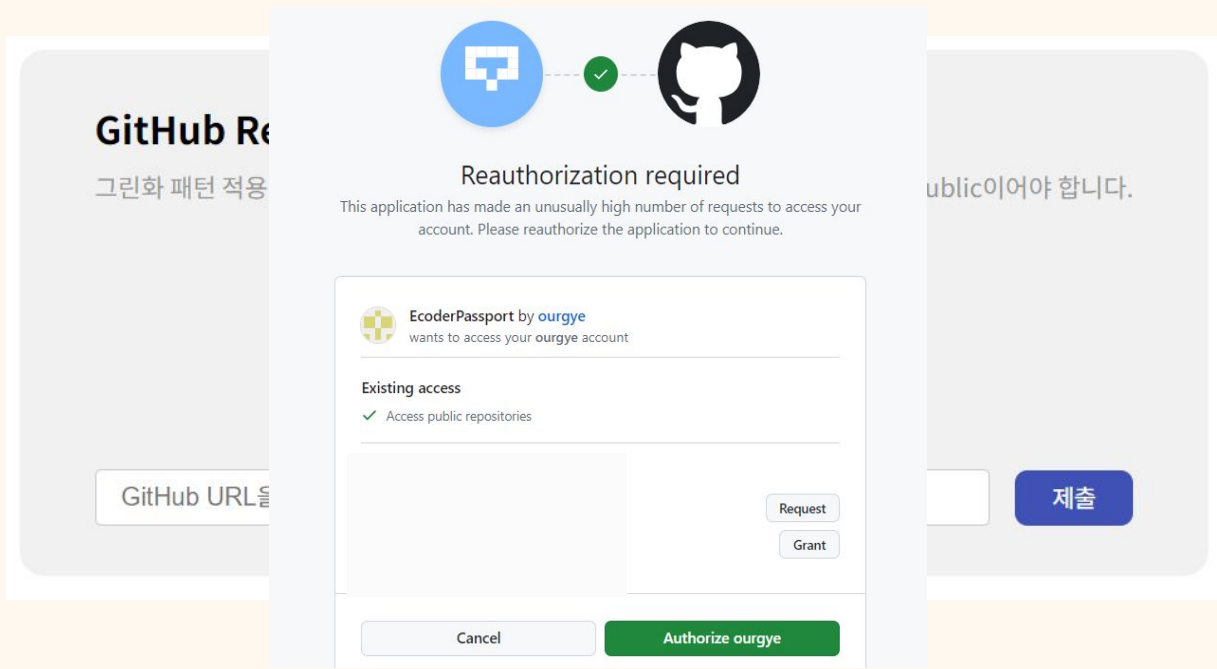
GitHub Repository URL을 입력하세요.

그린화 패턴 적용 후 pull request 해당 repository로 보냅니다. Repository는 Public이어야 합니다.



제출

Implementation: GitHub Repository



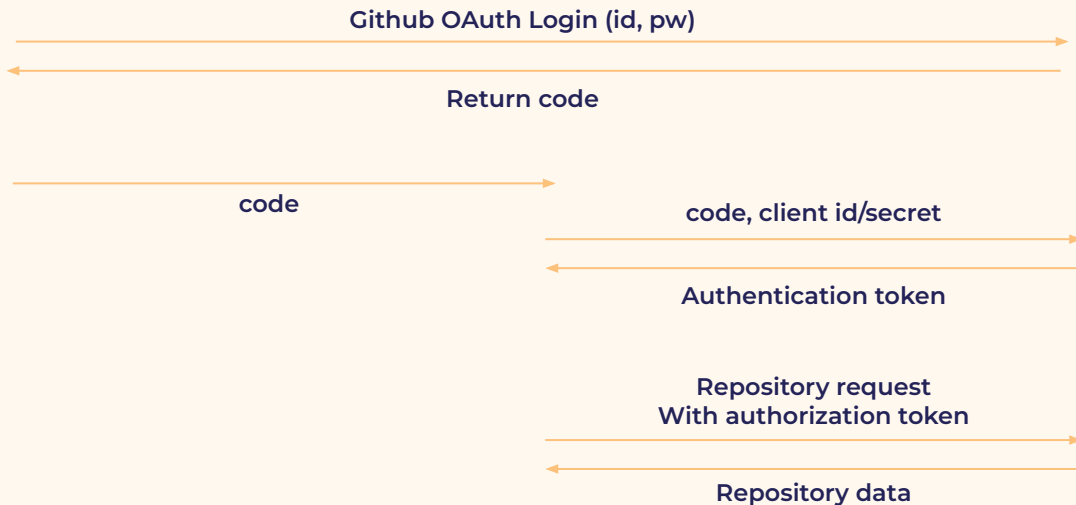
Implementation: **GitHub Repository**

Github OAuth Flow

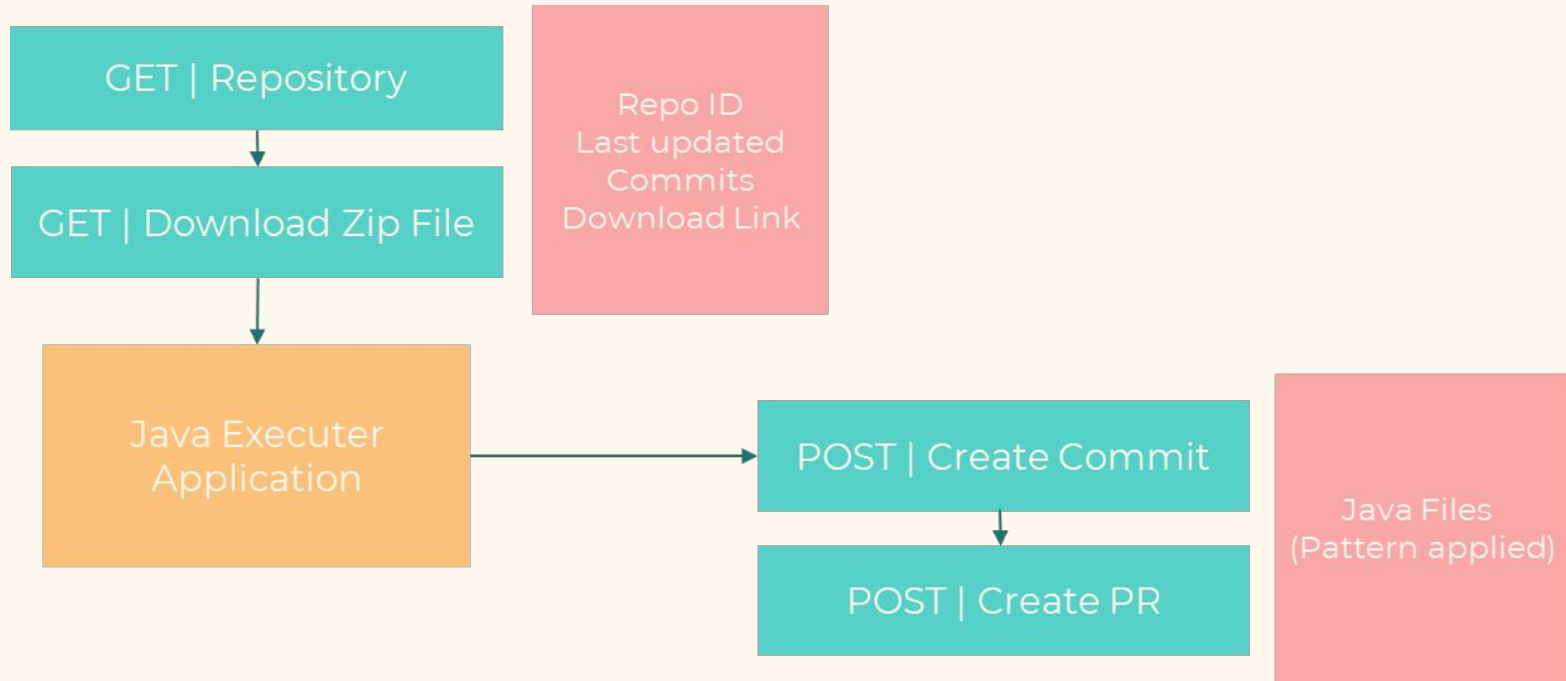
Client(Web)

Server(Backend)

Github



Implementation: **GitHub Repository**



Implementation: **GitHub Repository**

On Success



PR이 완료되었습니다. 결과를 확인하러 가기 🎉



On Failure

에러가 발생했습니다. 다시 시도해주세요.

다시 입력하기

Implementation: **Code Execution**



코드 실행: Spring boot(JAVA)에서 Shell Script 호출



- 요청이 들어올 때마다 새로운 임시 폴더 자동 생성 (ex. java-code16922044790787257624)
- 폴더에 요청으로 받은 JAVA 코드를 Temp.java로 저장
- javac를 사용해 JAVA 파일 컴파일
- 실행 시간: 컴파일 된 JAVA 프로그램 실행 전후의 timestamp 값 차이 계산
- 메모리 사용량: 실행한 java 프로세스의 pid로 VmRSS 값을 조회



Implementation: **Code Execution**

사용 메모리 계산:

```
get_memory_usage() {  
    pid=$1  
    memory_usage=$(awk '/VmRSS/{print $2}' /proc/$pid/status)  
    echo "$memory_usage"  
}
```

실행 시간 계산:

```
start_time=$(date +%s%N)  
  
end_time=$(date +%s%N)
```



Green Pattern: Upgrade



```
public class BuggyTest {  
    public static void main(String[] args) {  
        ArrayList<String> arr = new ArrayList<>();  
        arr.add("a");  
        arr.add("b");  
        arr.add("c");  
        arr.add("d");  
        arr.add("e");  
        arr.add("f");  
        arr.add("g");  
  
        boolean cond1 = checkCond1();  
        boolean cond2 = checkCond2();  
        boolean cond3 = checkCond3();
```

```
        if(cond1) {  
            if(cond2) {  
                if(cond3) {  
                    System.out.println("Triple Nested If");  
                }  
            }  
        }  
    }  
}
```

```
        if(cond1) {  
            if (cond2) {  
                System.out.println("Double Nested If");  
            }  
        }  
    }  
}
```

```
        if(cond1) {  
            if(cond2) {  
                if(cond3) {  
                    System.out.println("Triple Nested If");  
                }  
            }  
        }  
    }  
}
```

```
        for(int i=0; i<arr.size(); i++) {  
            Random r = new Random();  
            System.out.println("Hello");  
        }  
    }  
}
```

Indent 개선

여러 케이스
한번에 처리

2중 중첩 if문 추가

```
public class BuggyTest {  
    public static void main(String[] args) {  
        ArrayList<String> arr = new ArrayList<>();  
        arr.add("a");  
        arr.add("b");  
        arr.add("c");  
        arr.add("d");  
        arr.add("e");  
        arr.add("f");  
        arr.add("g");  
  
        boolean cond1 = checkCond1();  
        boolean cond2 = checkCond2();  
        boolean cond3 = checkCond3();
```

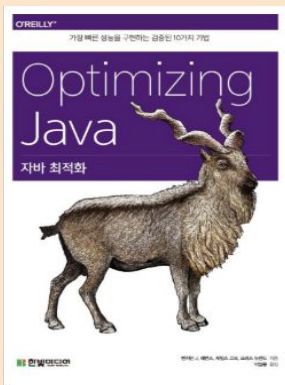
```
        if((cond1 && cond2) && cond3) {  
            System.out.println("Triple Nested If");  
        }  
    }  
}
```

```
        if(cond1 && cond2) {  
            System.out.println("Double Nested If");  
        }  
    }  
}
```

```
        if((cond1 && cond2) && cond3) {  
            System.out.println("Triple Nested If");  
        }  
    }  
}
```

```
        int arrSize = arr.size();  
  
        Random r = new Random();  
        for(int i=0; i<arrSize; i++) {  
            System.out.println("Hello");  
        }  
    }  
}
```


Green Pattern: **Collect**



서적 참고



구글링

BAEKJOON
ONLINE JUDGE

 **programmers**

 **LeetCode**

PS 기법

Green Pattern: **Example**

StringToStringBuilderPattern

* 총 다섯 번 반복 실행하여 평균값을 구함

```
public class Buggy {  
    public static void main(String[] args) {  
        String temp = "";  
  
        for(int i=0; i<10000; i++) {  
            temp += "a";  
            temp += "a";  
            temp += "a";  
            temp += "a";  
            temp += "a";  
        }  
    }  
}
```

Before

Runtime: 0.0012(s)
Memory: 25893(KB)
Emission: 1.9141753E-5(gCO2e)

```
public class Fixed {  
    public static void main(String[] args) {  
        StringBuilder temp = new StringBuilder("");  
  
        for(int i=0; i<10000; i++) {  
            temp.append("a");  
            temp.append("a");  
            temp.append("a");  
            temp.append("a");  
            temp.append("a");  
        }  
    }  
}
```

After

Runtime: 0.00087(s)
Memory: 17024(KB)
Emission: 1.35293385E-5(gCO2e)

Project Management

Frontend



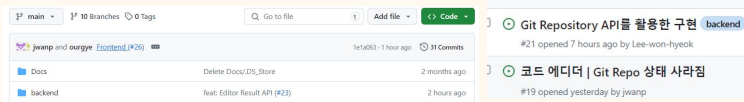
Backend



Both



Branch
Issue/PR
Udacity Commit Convention



소프트웨어공학 개론 6조



감사합니다!