Project : Wap Rules

#1 관리자, WapRules class (main)

#2 TopPilar Class

#3 MidFoundation Class

#4 MidPilar Class

#5 BottomFoundation class

#6 BottomSign class

#7 BottomFilar class

#1 축하합니다! 당신은 WapRules Project의 관리자로 선정되셨습니다!

당신이 해야 할 일은 다음과 같습니다.

1. GitHub에 WapRules 라는 (Private) 저장소를 만드세요!
2. 원하는 디렉토리에 WapRules라는 폴더를 만드세요.
3. 그 폴더에 git 저장소를 init 하세요.
4. 그리고 그 폴더에 src 라는 이름의 폴더를 만들고, WapRules.java 파일을 생성하세요.
5. 패키지 이름은 waprules (모두 소문자로 하세요) 라고 짓고, WapRules라는 비어있는 public class를 만들고, Initial Commit 이라는 메시지로 커밋하고 원격 저장소에 Push하세요.
6. 그리고 팀원들의 Github 아이디를 모아서, Github 저장소에서 Setting->Collaborator에 추가하세요.
7. 그 후 세팅에서 master 가지를 protected로 설정하고, **Require pull request reviews before merging**를 체크하고 저장하세요.
8. 팀원들이 Pull Request를 보내고, 해당 Pull Request가 다른 팀원에 의해 Review되었다면, Merge 해 주세요.
9. TopPilar Class, MidFoundation Class, MidPilar Class, BottomFoundation class, BottomSign class, BottomPilar class 모두가 merge 되었다면, 4에서 작성한 WapRules.java에 main함수를 작성하고, 팀원들이 작성한 모든 클래스를 Import합니다.
10. 그리고 다음 명세서를 보고 메인 함수를 작성하세요.

main(String[] args) 함수의 기능 명세

TopPilar, MidFoundation, MidPilar, BottomFoundation, BottomSign, BottomPilar의 객체 생성

생성된 객체들의 메소드를 다음과 같은 순서로 호출

1. TopPilar.build()
2. MidFoundation.build()
3. MidPilar.build()
4. MidPilar.build()
5. BottomFoundation.build()
6. BottomPilar.build()
7. BottomSign.topSign()
8. BottomPilar.build()
9. BottomSign.botSign() , BottomPilar.build(), BottomFoundation.build()

#2 TopPilar class

당신이 해야 할 일은 다음과 같습니다.

1. 당신의 팀의 관리자에게 Github 아이디 알려주기
2. 관리자가 보내주는 WapRules 저장소의 초대 받기
3. WapRules 저장소 clone 하기, 그리고 top\_pilar 라는 branch를 만들고 체크아웃
4. WapRules/src 폴더에 TopPilar.java 파일 만들기
5. Public class TopPilar 를 다음 명세를 보고 구현하기.

Class TopPilar

public void build()

이 함수는 System.out.println 으로 “ 0 “ 문자열을 6번 출력합니다.

“ 0 “는 스페이스 6번, 숫자 0 한번, 그리고 다시 스페이스 6번 입니다.

1. 클래스를 구현했으면 커밋하고 원격 저장소에 top\_pilar 브랜치 push하기
2. Pull Request를 넣고, Pull Request의 코멘트에 위 클래스 명세서를 명시하세요.

#3 MidFoundation class

당신이 해야 할 일은 다음과 같습니다.

1. 당신의 팀의 관리자에게 Github 아이디 알려주기
2. 관리자가 보내주는 WapRules 저장소의 초대 받기
3. WapRules 저장소 clone 하기, 그리고 mid\_foundation branch를 만들고 체크아웃
4. WapRules/src 폴더에 MidFoundation.java 파일 만들기
5. Public class MidFoundation 를 다음 명세를 보고 구현하기.

Class MidFoundation

public void build()

이 함수는 System.out.println 으로 “ 00000 “ 문자열을 출력합니다.

“ 00000 “는 스페이스 4번, 숫자 0 5번, 그리고 다시 스페이스 4번 입니다.

1. 클래스를 구현했으면 원격 저장소에 mid\_foundation 브랜치 push하기
2. Pull Request를 넣고, Pull Request의 코멘트에 위 클래스 명세서를 명시하세요.

#4 MidPilar class

당신이 해야 할 일은 다음과 같습니다.

1. 당신의 팀의 관리자에게 Github 아이디 알려주기
2. 관리자가 보내주는 WapRules 저장소의 초대 받기
3. WapRules 저장소 clone 하기, 그리고 mid\_pilar 라는 branch를 만들고 체크아웃
4. WapRules/src 폴더에 MidPilar.java 파일 만들기
5. Public class MidPilar 를 다음 명세를 보고 구현하기.

Class MidPilar

public void build()

이 함수는 System.out.println 으로 “ 0 0 “ 문자열을 4번 출력합니다.

문자열은 SPACE 4번, 그리고 숫자 0, SPACE 3번, 그리고 숫자 0, 그리고 SPACE 4번입니다.

1. 클래스를 구현했으면 커밋하고 원격 저장소에 mid\_pilar 브랜치 push하기
2. Pull Request를 넣고, Pull Request의 코멘트에 위 클래스 명세서를 명시하세요.

#5 BottomFoundation class

당신이 해야 할 일은 다음과 같습니다.

1. 당신의 팀의 관리자에게 Github 아이디 알려주기
2. 관리자가 보내주는 WapRules 저장소의 초대 받기
3. WapRules 저장소 clone 하기, 그리고 bottom\_foundation 라는 branch를 만들고 체크아웃
4. WapRules/src 폴더에 BottomFoundation.java 파일 만들기
5. Public class BottomFoundation 를 다음 명세를 보고 구현하기.

Class BottomFoundation

public void build()

이 함수는 System.out.println 으로 “0000000000000“ 문자열을 출력합니다.

문자열은 0이 13개 입니다.

1. 클래스를 구현했으면 커밋하고 원격 저장소에 bottom\_foundation 브랜치 push하기
2. Pull Request를 넣고, Pull Request의 코멘트에 위 클래스 명세서를 명시하세요.

#6 BottomPilar class

당신이 해야 할 일은 다음과 같습니다.

1. 당신의 팀의 관리자에게 Github 아이디 알려주기
2. 관리자가 보내주는 WapRules 저장소의 초대 받기
3. WapRules 저장소 clone 하기, 그리고 bottom\_pilar 라는 branch를 만들고 체크아웃
4. WapRules/src 폴더에 BottomPilar.java 파일 만들기
5. Public class BottomPilar 를 다음 명세를 보고 구현하기.

Class BottomPilar

public void build()

이 함수는 System.out.println 으로 “0 0 “ 문자열을 세 번 출력합니다.

문자열은 숫자 0 하나, SPACE 11번, 그리고 다시 숫자 0 입니다

1. 클래스를 구현했으면 커밋하고 원격 저장소에 bottom\_pilar 브랜치 push하기
2. Pull Request를 넣고, Pull Request의 코멘트에 위 클래스 명세서를 명시하세요.

#7 BottomSign class

당신이 해야 할 일은 다음과 같습니다.

1. 당신의 팀의 관리자에게 Github 아이디 알려주기
2. 관리자가 보내주는 WapRules 저장소의 초대 받기
3. WapRules 저장소 clone 하기, 그리고 bottom\_sign 라는 branch를 만들고 체크아웃
4. WapRules/src 폴더에 BottomSign.java 파일 만들기
5. Public class BottomSign 를 다음 명세를 보고 구현하기.

Class BottomSign

public void topSign()

이 함수는 System.out.println 으로 “0 WAP 0 “ 문자열을 출력합니다.

문자열은 숫자 0, SPACE 4번, WAP, SPACE 4번, 숫자 0입니다

Public void botSign()

이 함수는 System.out.println 으로 0 RULES 0 “ 문자열을 출력합니다.

문자열은 숫자 0, SPACE 3번, RULES, SPACE 3번, 숫자 0 입니다

1. 클래스를 구현했으면 커밋하고 원격 저장소에 bottom\_sign브랜치 push하기
2. Pull Request를 넣고, Pull Request의 코멘트에 위 클래스 명세서를 명시하세요.