

The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on the left side are several concentric circles and arcs. Some of these arcs have degree markings, including 40, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, and 260. There are also curved arrows indicating a clockwise direction.

Singleton 패턴 실습 문제

요구 사항

- 다른 방법으로 싱글턴 패턴 구현하기
 - (1) Bill Pugh Singleton: inner helper 클래스 이용
 - (2) enum Singleton: 실행 시 모든 Enum 값들은 하나만 생성된다는 사실을 이용
- Captain 클래스에 싱글턴 패턴 적용하기
 - `getCaptain()`: 유일한 Captain 객체를 리턴하는 메소드
 - Bill Pugh Singleton 적용
- CaptainEnum 클래스에 싱글턴 패턴 적용하기
 - `getCaptain()`: 유일한 Captain 객체를 리턴하는 메소드
 - enum Singleton 적용

요구 사항

- Client 클래스(main())
 - Captain 객체를 2개 생성하여 객체의 주소가 동일한 지 확인
 - CaptainEnum 객체를 2개 생성하여 객체의 주소가 동일한 지 확인

출력 예

```
***Singleton Pattern Demo: Bill Pugh ***
```

```
Caption: hello!
```

```
Trying to make another captain:
```

```
Caption: hello!
```

```
c1 and c2 are the same instance.
```

```
***Singleton Pattern Demo: enum ***
```

```
Caption: hello!
```

```
Trying to make another captain:
```

```
Caption: hello!
```

```
c3 and c4 are the same instance.
```