캡스톤디자인I (CSE4186) 중간고사 리포트

2024-04-24

1. Sequences (20점)

Kotlin Sequence는 Iterator를 통해 각 element에 순차적으로 operation을 수행할 수 있는 type 중 하나로 terminal operation을 통해서만 evaluation이 시작되는 lazy 속성을 가진다.

```
public interface Sequence<out T> {
	public operator fun iterator(): Iterator<T>
}
```

A. `Sequence`의 extension function `map`은 `TransformingSequence`를 리턴하는 intermediate operation이다. `map` function과 `TransformingSequence` class의 소스코드를 예시하여 intermediate operation의 동작방식을 상세 설명하시요.

B. `Sequence` interface의 terminal operation 중 하나인 `first` function 소스코드를 예시하고 intermediate operation과 비교하여 `Sequence`의 lazy evaluation 구현 방식을 구체적으로 기술하시요.

2. Generic (20점)

Kotlin `Collection` 중 generic interface `List`와 `MutableList` 소스코드를 참조하여

A. 각각의 type parameter variance를 확인하고 서로 다른 `variance`를 가지는 이유를 설명하시요.

B. `List`의 method 중 선언된 type parameter variance에 위배되는 function의 사례 2~3 개를 제시하고, compile time error 회피를 위하여 사용된 방법을 설명하시요.

C. B.에서 제시한 method가 variance 위배에도 불구하고 type safe 한 이유를 해당 method의 operation 속성을 예시하여 기술하시요.

3. Scope functions (20점)

아래는 scope function `also`를 이용한 one-line swap 코드이다.[1]

```
var a = 1
var b = 2
a = b.also { b = a }
```

A. 위 코드에서 변수 값 swap이 가능한 이유를 구체적으로 설명하시오.

B. Scope function `apply`, `let`, `run`을 이용하여 변수 추가 없이 동일한 기능을 구현하는 one-line swap 코드를 작성하시요.

C. Scope function `apply`, `apply`, `let`, `run`의 소스코드를 제시하여 각 함수의 동작과 대표적인 use case를 비교하여 설명하시요.

4. Collection (20점)

A. `fold()` 함수를 사용하여 List의 element 개수를 구하는 `size()` 함수를 구현하시요.

```
fun <T> List<T>.size(): Int =
   TODO("fold(0) { ??? }")
```

B. `flatMap()` 함수를 사용하여 `filter()` 함수를 구현하시요.

```
fun <T> List<T>.filter(p: (T) -> Boolean): List<T> =
   flatMap { TODO() }
```

[1] https://kotlinlang.org/docs/idioms.html#swap-two-variables