



FACULTAD DE INGENIERÍA Y ARQUITECTURA

INGENIERÍA EN DESARROLLO DE SOFTWARE

MANEJO DE ESTRUCTURA DE DATOS

Tarea 1: Números Primos

Docente:

Ing. Sergio Alberto Bernal

11 de agosto de 2024

Índice

1. Introducción	3
2. Objetivos	3
3. Instrucciones	3
3.1. 1. Definir Clases para Tipos de Números	3
3.2. 2. Creación de Interfaz Común	3
3.3. 3. Evaluación y Clasificación de Números	3
3.4. 4. Implementación de la Lógica de Clasificación	4
3.5. 5. Salida y Operaciones Específicas	4
4. Puntos a Considerar	4
5. Conclusión	4

1. Introducción

En este ejercicio, implementarás un sistema en Kotlin que clasifica y manipula números enteros dentro de un rango especificado. El objetivo es identificar números primos, pares e impares, y realizar operaciones específicas sobre cada tipo de número utilizando clases e interfaces.

2. Objetivos

- **Entender la clasificación de números:** Desarrollarás un algoritmo que identifique si un número es primo, par o impar.
- **Modelar tipos de datos con clases:** Crearás clases específicas para representar números primos, pares e impares, cada una con sus propias propiedades y métodos.
- **Interfaz común:** Implementarás una interfaz para garantizar que todas las clases puedan manejarse de manera uniforme.
- **Manipulación y evaluación:** Evaluarás un conjunto de números y los clasificarás, realizando operaciones específicas según su tipo.

3. Instrucciones

3.1. 1. Definir Clases para Tipos de Números

- Deberás crear tres clases principales: **PrimeNumber**, **OddNumber** y **EvenNumber**, cada una implementando la interfaz **IBaseNumber**.
- La clase **PrimeNumber** debe representar números primos y sobrescribir un método de la interfaz para imprimir su valor.
- Las clases **OddNumber** y **EvenNumber** deben manejar números impares y pares respectivamente, incluyendo una propiedad para almacenar sus divisores y un método para imprimirlos.

3.2. 2. Creación de Interfaz Común

- Implementa la interfaz **IBaseNumber** que contiene una propiedad **value** de tipo **Int** y un método **printValue()** para imprimir el valor del número.
- Asegúrate de que todas las clases de números (**PrimeNumber**, **OddNumber**, **EvenNumber**) implementen esta interfaz.

3.3. 3. Evaluación y Clasificación de Números

- Desarrolla una clase llamada **PrimeNumberProcessor** que reciba un rango de números y los clasifique en primos, pares e impares.

- Dentro de esta clase, crea un método privado que valide cada número (**validateNumber**) y determine su tipo (**NumberType**) basado en los divisores.
- Asegúrate de que el resultado de esta evaluación sea almacenado en una instancia de **EvaluationResult**, que contiene arreglos de cada tipo de número.

3.4. 4. Implementación de la Lógica de Clasificación

- En la clase **PrimeNumberProcessor**, utiliza estructuras condicionales para clasificar cada número según los resultados de la validación, añadiéndolo a la colección correspondiente.
- Define cómo los números pares e impares deben almacenar sus divisores, y asegúrate de que los números primos se manejen correctamente como un caso especial.

3.5. 5. Salida y Operaciones Específicas

- Al final de la evaluación, itera sobre cada colección de números (primos, pares, impares) y aplica las operaciones definidas en sus clases, como imprimir el valor y los divisores.
- Para los números pares e impares, asegúrate de que se impriman los divisores almacenados en sus propiedades.

4. Puntos a Considerar

- **Optimización:** Piensa en cómo podrías mejorar la eficiencia del método **validateNumber**, especialmente en la identificación de números primos.
- **Extensibilidad:** Considera cómo podrías modificar o extender la solución para manejar otros tipos de números o realizar operaciones adicionales.
- **Seguridad de Tipos:** Asegúrate de que el diseño de tu sistema maneje correctamente los tipos y evite errores comunes, especialmente en la manipulación de números y sus divisores.

5. Conclusión

Este ejercicio te permitirá practicar la implementación de clases e interfaces en Kotlin, aplicando lógica condicional y estructurando un programa para manejar diferentes tipos de números y sus operaciones asociadas.