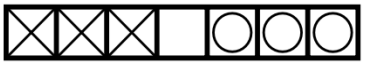
CSED232 Assignment 2

Due Wednesday, March 12

Problem 1: The Hoping Rabbits Game

- Two teams of n rabbits each are placed facing each other in a row with $2n + 1$ positions. The `x` team occupies the first n positions, the `o` team occupies the last n positions, and the middle position is left empty. E.g., the following shows the initial state for $n = 3$.



- Rabbits from the x team can only move to the right, and rabbits from the o team can only move to the left. A rabbit can move to a position if that position is empty. A rabbit can jump over a competitor if the position behind the competitor is empty.
- The goal is to swap the positions of the two teams by moving the rabbits. The class `HopingRabbitsGame` includes several methods, including `move`, `isGoal`, and `isStuck`. See the javadoc comments for more details.

Problem 2: Number Translator

- In this problem, you will implement a number translator that converts numeric values into words based on a given locale. You will implement two language-specific translators: `KoreanNumberTranslator` and `EnglishNumberTranslator`.
- The interface `NumberTranslator` defines the following method to convert a number into words, according to its locale. See the javadoc comments for more details.
  - `String toWords(long number)`
- The class `AbstractNumberTranslator` provides a skeletal implementation. It implements a common translation algorithm in the `toWords` method but delegates locale-specific logic to additional abstract methods. Implement `AbstractNumberTranslator`.
- Implement two classes, `KoreanNumberTranslator` and `EnglishNumberTranslator`, which are subclasses of `AbstractNumberTranslator`. These classes cannot override the `toWords` method, as it is declared `final` in `AbstractNumberTranslator`
- Minimize duplicate code by using the abstract class `AbstractNumberTranslator` (See the comments for more details):
  - In `AbstractNumberTranslator`, define abstract methods for locale-specific logic, and implement the `toWords` method using these abstract methods.
  - In `KoreanNumberTranslator` and `EnglishNumberTranslator`, override the abstract methods to implement locale-specific logic. You cannot override `toWords` here.

Turning In

1. Make your code as easy to read and understand as possible. You can add more (private) fields and methods (e.g., to avoid duplicate code) if you want.
2. The `src/main` directory contains the skeleton code. You should implement all classes and methods with *TODO* in the above classes.
3. The `src/test` directory contains some test cases. You can check them by running the classes in `src/test`. (For grading, we may use additional test cases.)
4. We use Github Classroom to manage homework. Click on the following link, and accept the assignment: `https://classroom.github.com/a/-gcik8bo`
5. Your assignment repository should now be created. Clone the repository, complete your homework, and push the changes to GitHub before the deadline.

Java Reference

- Java Language Specification: `https://docs.oracle.com/javase/specs/`
- Learn Java: `https://dev.java/learn/`
- Core Java, Volume I: Fundamentals 13th by Cay S. Horstmann, Pearson, 2024 (available online at the POSTECH digital library `http://library.postech.ac.kr`)