

Zombie Apocalypse

Assumptions:

The world is represented by $N \times N$ matrix

Zombies and Creatures live within it, both of their positions are represented by x and y coordinates within the world matrix.

Zombies' valid movements are Up, down, Right, Left.

Zombies can break the barriers and form a cyclic motion within rows and columns. i.e it can move come to beginning of the column or row if it reaches the edge of row/column.

Creatures cannot move. If zombie meets creatures on its movement, it destroys the creature and new zombie is formed.

Once zombie completes its movement, the other zombies which was created starts the same movement as its previous and it continues until all the zombies completes.

Once all the zombies are complete, print all the final positions of zombies and creatures.

Inputs:

Dimensions of World– N integer

Initial position of zombie - X, Y coordinates

List of initial position of creatures – List<X, Y> coordinates

List of moves zombie can make – String UDLR

Approach:

Public api -> for setting up the world with zombies and creatures.

```
setUpZombieWorld(int worldSize, Position zombiePos, List<Position> creaturesPos, String moves) {}
```

worldSize cannot be zero or lesser than that

zombiePosition and creaturesPosition should not be greater than world size and should not be lesser than zero

Moves should have valid movement and cannot allow special characters or other characters apart from the pre-defined movement – UDLR.

```
startZombieInfection() {}
```

 - for starting the zombie infection.

Zombies are represented as 1, creatures as 2 and zombie's final positions are represented as 3 in the world.

ZOMBIE (1), CREATURES (2), ZOMBIE_FINALPOSITION (3);

Zombies initial position is inserted into the Queue, when zombie encounters the creature (2) in the world, the new zombie position (X, Y) is inserted to the queue and the queue is polled and iterated for all the movement(string) until all the zombie complete their movement.

Once zombie encounters creatures (2), apart from adding the position to Queue, the creature is removed from list of creatures and the position(x,y) is marked as zombie (1).

After each zombie completes its moves, its final position is marked as 3 in the world.

After all the zombie complete their moves, the final position of zombies is taken from the world and since we already removed the infected creatures from the list, the remaining creatures in the list are the final one's alive.

InputsForZombieApocalypse.java -> is the main program for running the application. Have created one Object of ZombieApocalypse to drive the program. Values can be changed here to simulate various scenarios.

ZombieApocalypseTest.java -> is the test class for testing exceptions and working of the program.

All the output and logs are printed in the console.