Assignment 1

Tags	
Created time	@December 14, 2024 12:44 AM
Last edited time	@December 26, 2024 8:37 PM
Status	Not started

Task Goal

Divide a given map into 4 equal chambers. If the map can't be equally divided into 4 chambers, divide it into the largest possible equal number of chambers (e.g., 3, 2, or 1).

Constraints

- Use beepers to divide the map
- You can use double line of beepers if you need
- Use API to setup beepers to 1,000
- Use only function given in Karel reference

Measurements

- Use less beepers
- Lowest number of moves
- Minimize number of lines by writing reusable functions

Functions

- Moves Counter

Instruction

- Print moves count in each step (optional print used beepers)
  - We can override move function to do this job
  - we need variable to count number of steps and number of beepers

Workflow Steps

1. Get Map dimensions (width and height)
    a. if the map is square just get one of them
        i. go to the most right or most top
    b. if the map is not square we need to get both

> We'll assume that the map is rectangle (General Case) Karel always start from bottom left and his face to right so we'll assume this point as (0,0)

2. Decide to how many champers and how many line of beepers
3. Start divide

Map Cases

1×1 → 1 → can't divided

1×2 → 2 → can't divided

1×3 → 3 → divide into 2

1×4 → 4 → divide into 2 using two beepers

1×5 → 5 → divide into 3

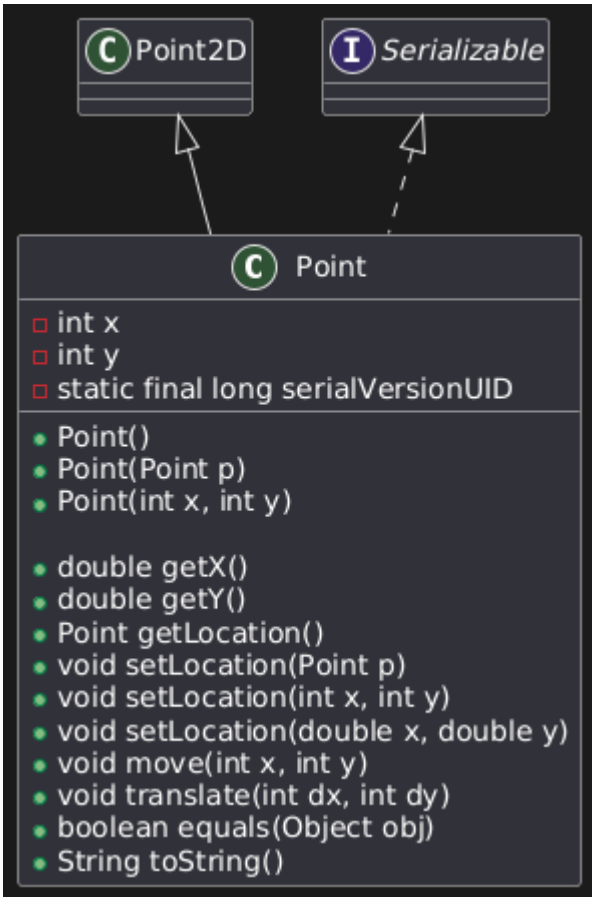
2×2 → 4 → can't divided

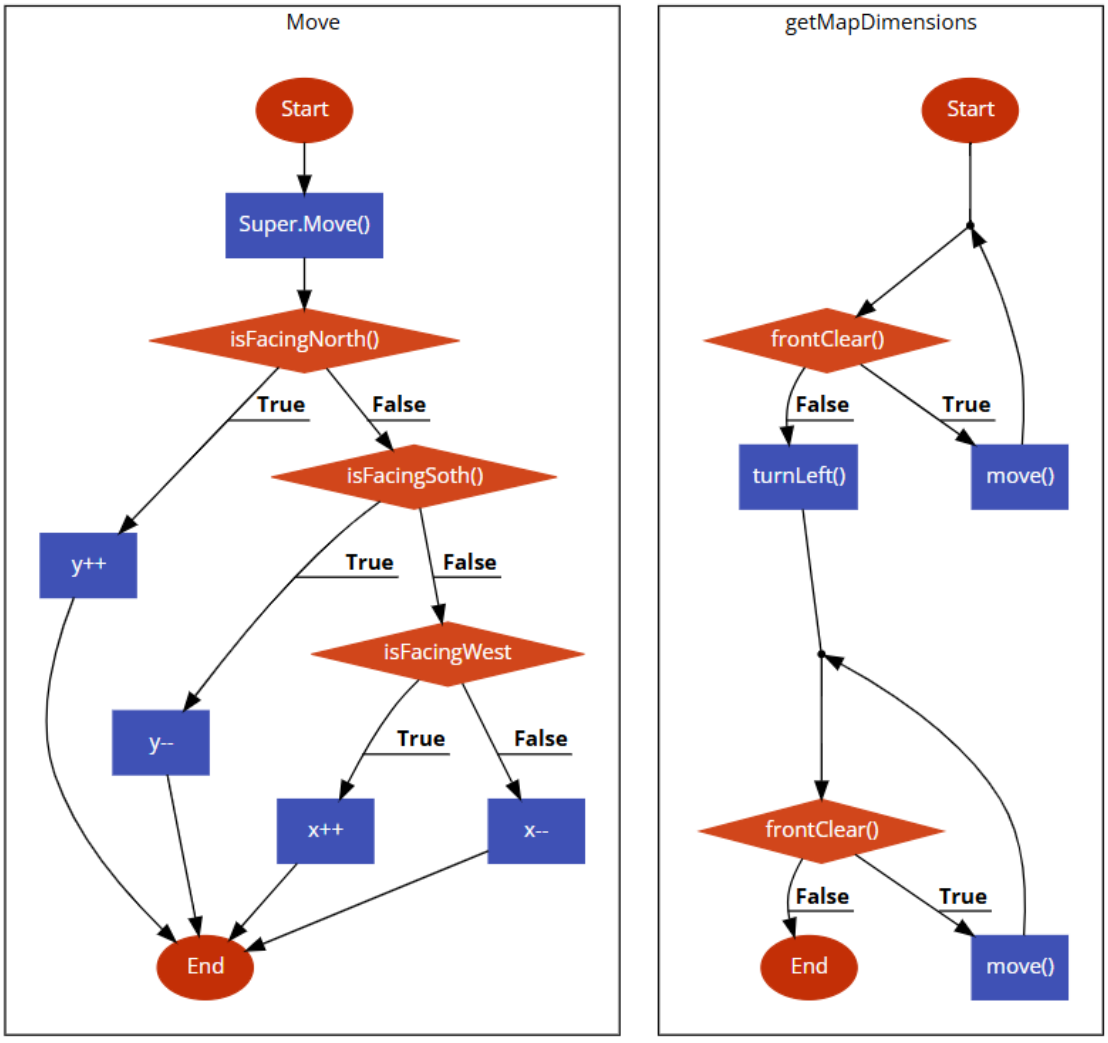
2×3 → 6 → into 2

r → rows

c → columns

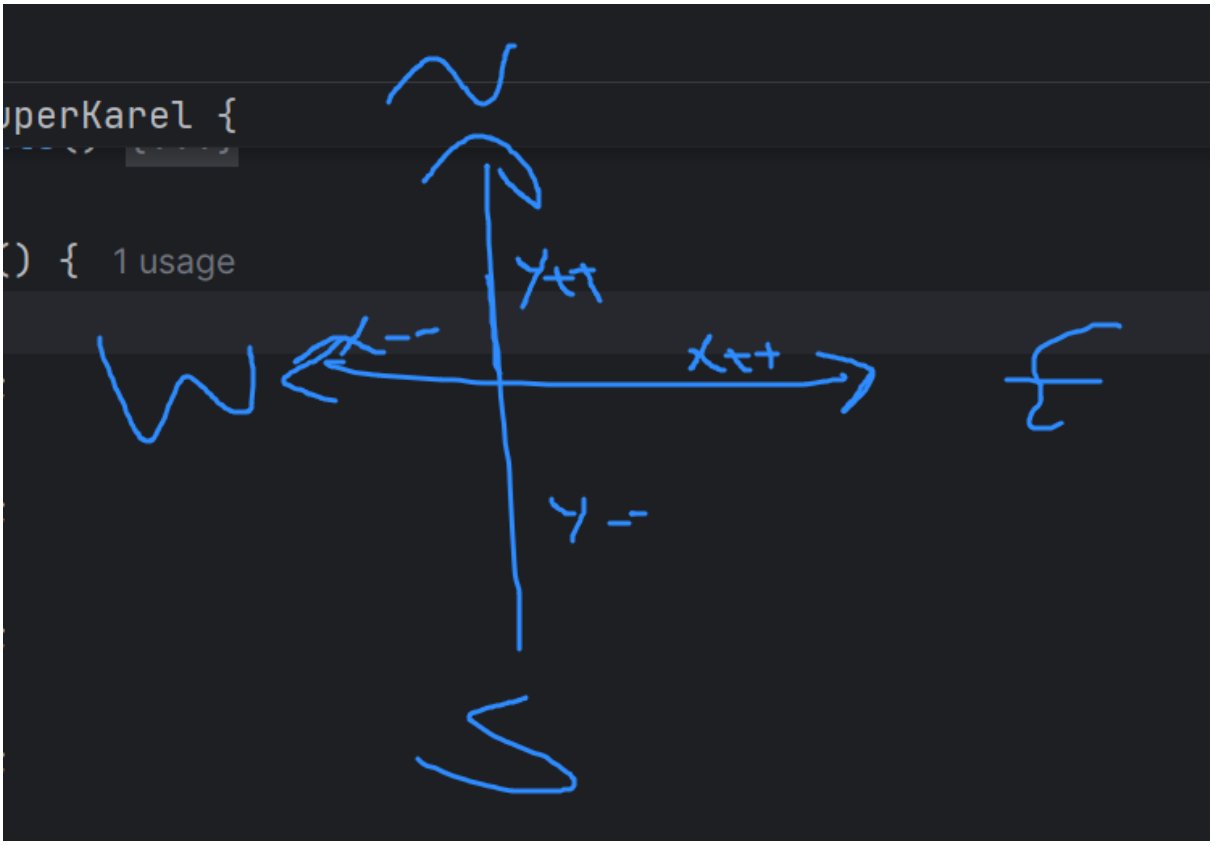
Point Class


Point2D
Serializable
Point
int x
int y
static final long serialVersionUID
Point()
Point(Point p)
Point(int x, int y)
double getX()
double getY()
Point getLocation()
void setLocation(Point p)
void setLocation(int x, int y)
void setLocation(double x, double y)
void move(int x, int y)
void translate(int dx, int dy)
boolean equals(Object obj)
String toString()



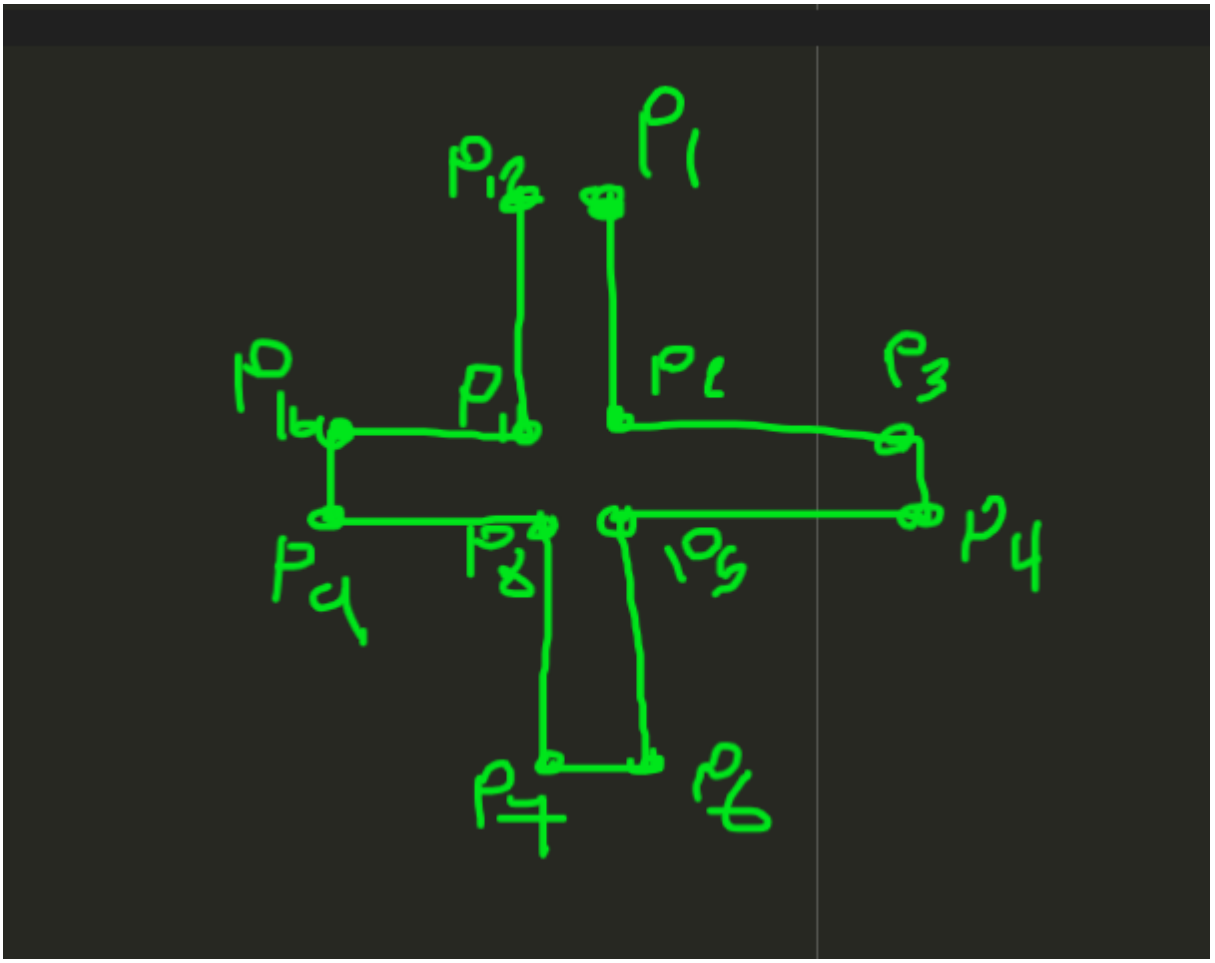
Double line of beepers → when the width or height is even

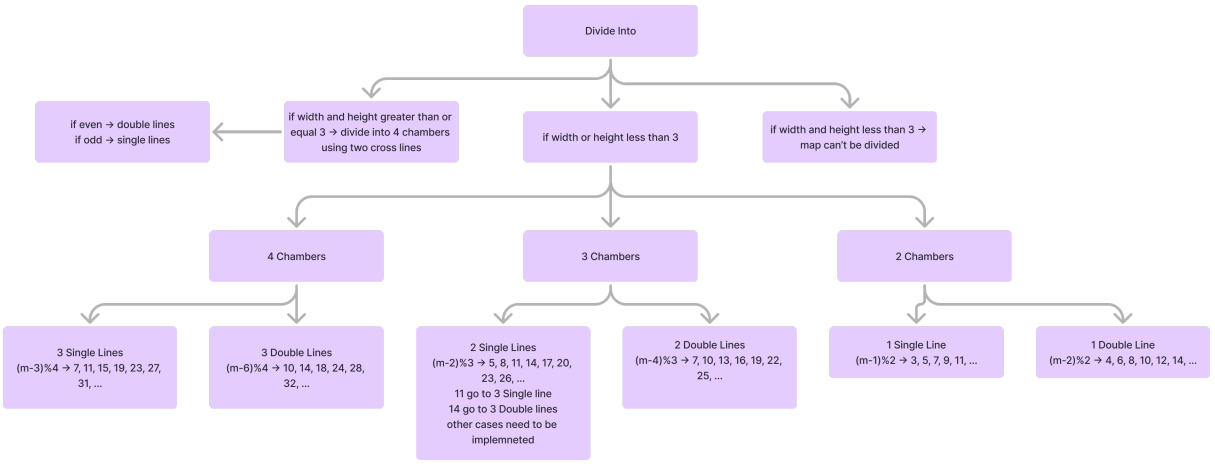
Update Location



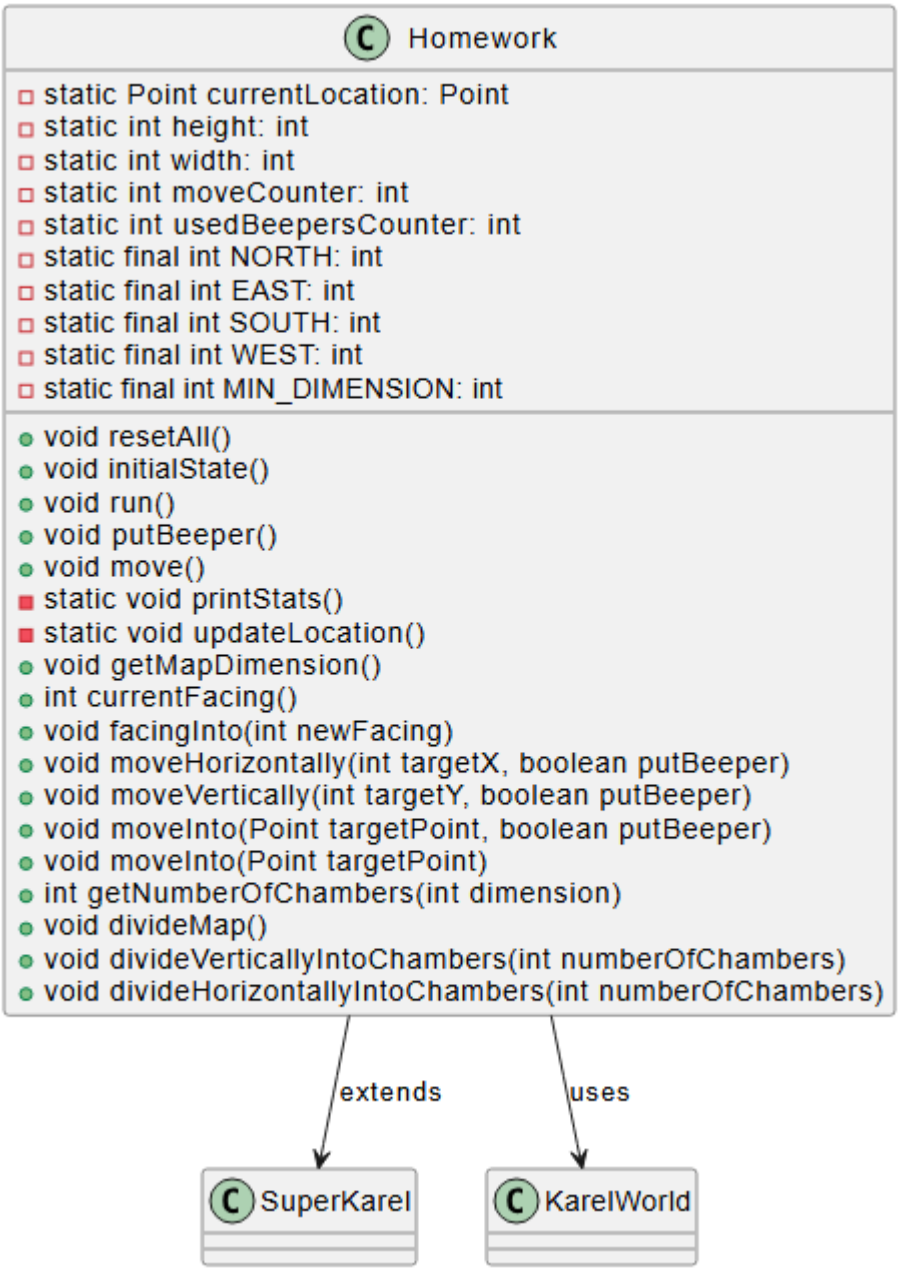

```
function updateLocation(){
  Start;
  if (facingNorth())
    currentLocation.y++;
  else if(facingSouth())
    currentLocation.y--;
  else if (facingEast)
    currentLocation.x++;
  else
    currentLocation.x--;
  End;
}

function move(){
  Start;
  Super.Move();
  call updateLocation();
```

```
  End;
}


function getMapDimensions(){
  Start;
  while(frontClear()){
    call move();
  }
  turnLeft();
  while(frontClear()){
    call move();
  }
  End;
}

function run(){
  Start;
  call getMapDimensions();
  End;
}
```





Class Diagram


Homework
static Point currentLocation: Point
static int height: int
static int width: int
static int moveCounter: int
static int usedBeepersCounter: int
static final int NORTH: int
static final int EAST: int
static final int SOUTH: int
static final int WEST: int
static final int MIN_DIMENSION: int
void resetAll()
void initialState()
void run()
void putBeeper()
void move()
static void printStats()
static void updateLocation()
void getMapDimension()
int currentFacing()
void facingInto(int newFacing)
void moveHorizontally(int targetX, boolean putBeeper)
void moveVertically(int targetY, boolean putBeeper)
void moveInto(Point targetPoint, boolean putBeeper)
void moveInto(Point targetPoint)
int getNumberOfChambers(int dimension)
void divideMap()
void divideVerticallyIntoChambers(int numberOfChambers)
void divideHorizontallyIntoChambers(int numberOfChambers)
extends
uses
SuperKarel
KarelWorld