**컴퓨터프로그래밍및실습**

과제 2 HKarel 문제 풀이 1

**202200423 김건우**

**목차**

1. 개조된 Karel 클래스 소개
   1. AdvancedMoveable 인터페이스 소개
   2. GeonuKarel 클래스 및 멤버 소개
   3. GeonuKarelMk2 클래스 및 멤버 소개
2. 문제 풀이 설명
   1. 문제 1 코드 및 설명
   2. 문제 2 코드 및 설명
   3. 문제 3 코드 및 설명
   4. 문제 4 코드 및 설명
3. 결과
4. 코드 원본 첨부
5. **개조된 Karel 클래스 소개**
   1. AdvancedMovable.java  
      텍스트이(가) 표시된 사진

      자동 생성된 설명  
      void move(int distance) 메소드와 void moveToWall() 메소드를  
      추상 메소드로 갖는다.
   2. GeonuKarel.java  
      GeonuKarel 클래스는 과제 풀이에 도움이 되는 필드와 메소드를 포함하며  
      AdvancedMoveable 인터페이스를 구현한다.  
      텍스트이(가) 표시된 사진

      자동 생성된 설명  
      - boolean isOnRightSide: Karel의 위치가 월드의 우측 끝인지 여부를 저장한다.  
      - 생성자에서는 Karel의 Beeper 개수를 9999개로 세팅한다.  
      - void move(int distance): int형 변수를 인자로 받아 그 횟수만큼 움직인다.  
      - void moveToWall(): 벽을 만날 때까지 움직인다.
   3. GeonuKarelMk2.java  
      GeonuKarel에서 문제 3을 해결하기 위한 개량 버전이다.  
      텍스트이(가) 표시된 사진

      자동 생성된 설명  
      - GeonuKarelMk2는 한 칸 움직일 때마다 Beeper를 놓고 놓지 않기를 반복한다.
6. **문제 풀이 설명**
   1. 문제 1 코드 및 설명  
      KarelAssignment1\_1.java  
      텍스트이(가) 표시된 사진

      자동 생성된 설명  
      - Karel이 목적지에 가서 Beeper를 집고 원래 자리로 돌아온다.
   2. 문제 2 코드 및 설명  
      KarelAssignment1\_2.java  
      텍스트이(가) 표시된 사진

      자동 생성된 설명  
      - 기둥이 하나인 경우에도 수리하기 위해 do-while문을 사용하였다.  
      - 반복문 처음 시행 시에는 전진하지 않으며, 두번째 시행 부터는 4칸씩 전진한다.
   3. 문제 3 코드 및 설명  
      KarelAssignment1\_3.java  
      텍스트이(가) 표시된 사진

      자동 생성된 설명  
      - GeonuKarelMk2 객체를 생성하여 사용한다.  
      - 이동 중 자동으로 Beeper가 놓여지므로, 월드를 순회하는 코드를 설계하였다.
   4. 문제 4 코드 및 설명  
      KarelAssignmant1\_4.java  
      텍스트이(가) 표시된 사진

      자동 생성된 설명  
      - Beeper가 있는 곳에서 돌아서서 한 칸 전진하고 Beeper를 놓는다  
      - 위를 반복하면 Karel은 가운데보다 한 칸 왼쪽에 위치하게 된다.  
      - 이때 한 칸 전진하여 Beeper를 놓으면 가운데 자리만 2개의 Beeper가 놓이게 된다.  
      - Beeper를 하나씩 제거하면 가운데 자리에만 Beeper가 놓이게 된다
7. **결과**
   1. 모든 경우에서 프로그램이 의도대로 동작한다.  
      - 방의 폭이 한 칸인 경우  
      - 방의 폭이 두 칸 이상인 경우
8. **코드 원본 첨부**

AdvancedMoveable.java

package *bluebenny.karel*;

*public* *interface* AdvancedMoveable {

*public* *void* move(*int* *distance*);

*public* *void* moveToWall();

}

GeonuKarel.java

package *bluebenny.karel*;

import *hufs.karel.HKarel*;

*public* *class* GeonuKarel *extends* *HKarel* *implements* *AdvancedMoveable* {

*boolean* isOnRightSide = false;

*public* GeonuKarel(*String* *name*) {

*super*(name);

setBeepersInBag(9999);

}

@*Override*

*public* *void* move(*int* *distance*) {

for(*int* i = 0; i < distance; i++)

*this*.move();

}

@*Override*

*public* *void* moveToWall() {

while(frontIsClear())

move();

*// 벽에 닿았을 때 오른쪽을 보고 있다 == karel이 오른쪽 끝에 와있다.*

isOnRightSide = facingEast();

}

}

GeonuKarelMk2.java

package *bluebenny.karel*;

*public* *class* GeonuKarelMk2 *extends* *GeonuKarel* {

*static* *boolean* beeperTurn = true;

*public* GeonuKarelMk2(*String* *name*) {

*super*(name);

}

@*Override*

*public* *void* move() {

if(beeperTurn) putBeeper();

*super*.move();

beeperTurn = !beeperTurn;

}

}

\KarelAssignment1\_1.java

package *bluebenny.karel*;

import *hufs.karel.HufsProgram*;

*public* *class* KarelAssignment1\_1 *extends* *HufsProgram* {

*GeonuKarel* karel;

@*Override*

*protected* *void* onInit() {

karel = (GeonuKarel)getHObject("건우");

}

@*Override*

*protected* *void* onStart() {

karel.move(2);

karel.turnRight();

karel.move();

karel.turnLeft();

karel.move();

karel.putBeeper();

karel.turnAround();

karel.move(3);

karel.turnRight();

karel.move();

karel.turnRight();

}

*public* *static* *void* main(*String*[] *args*) {

HufsProgram.main(args, new GeonuKarel("건우"));

}

}

KarelAssignment1\_2.java

package *bluebenny.karel*;

import *hufs.karel.HufsProgram*;

*public* *class* KarelAssignment1\_2 *extends* *HufsProgram* {

*GeonuKarel* karel;

@*Override*

*protected* *void* onInit() {

karel = (GeonuKarel)getHObject("건우");

}

@*Override*

*protected* *void* onStart() {

*boolean* mark = false;

do {

if(!mark) mark = true;

else karel.move(4);

karel.turnLeft();

if(karel.noBeepersPresent())

karel.putBeeper();

while(karel.frontIsClear()) {

karel.move();

if(karel.noBeepersPresent())

karel.putBeeper();

}

karel.turnAround();

karel.moveToWall();

karel.turnLeft();

} while(karel.frontIsClear());

}

*public* *static* *void* main(*String*[] *args*) {

HufsProgram.main(args, new GeonuKarel("건우"));

}

}

KarelAssignment1\_3.java

package *bluebenny.karel*;

import *hufs.karel.HufsProgram*;

*public* *class* KarelAssignment1\_3 *extends* *HufsProgram* {

*GeonuKarel* karel;

@*Override*

*protected* *void* onInit() {

karel = (GeonuKarel)getHObject("건우2");

}

@*Override*

*protected* *void* onStart() {

while(true) {

karel.moveToWall();

if(karel.isOnRightSide)

karel.turnLeft();

else

karel.turnRight();

if(karel.frontIsBlocked()) break;

karel.move();

if(karel.isOnRightSide)

karel.turnLeft();

else

karel.turnRight();

}

}

*public* *static* *void* main(*String*[] *args*) {

HufsProgram.main(args, new GeonuKarelMk2("건우2"));

}

}

KarelAssignment1\_4.java

package *bluebenny.karel*;

import *hufs.karel.HufsProgram*;

*public* *class* KarelAssignment1\_4 *extends* *HufsProgram* {

*GeonuKarel* karel;

@*Override*

*protected* *void* onInit() {

karel = (GeonuKarel)getHObject("건우");

}

@*Override*

*protected* *void* onStart() {

karel.putBeeper();

if(karel.frontIsBlocked()) return;

karel.moveToWall();

karel.putBeeper();

karel.turnAround();

karel.move();

while(karel.noBeepersPresent()) {

while(karel.noBeepersPresent())

karel.move();

karel.turnAround();

karel.move();

karel.putBeeper();

karel.move();

}

karel.turnAround();

karel.move();

karel.putBeeper();

karel.moveToWall();

karel.turnAround();

karel.pickBeeper();

while(karel.frontIsClear()){

karel.move();

karel.pickBeeper();

}

}

*public* *static* *void* main(*String*[] *args*) {

HufsProgram.main(args, new GeonuKarel("건우"));

}

}