Lösningsförslag övning 1.

```
public class Swapper {
	private Robot robot;
	public static void main(String[] args) {
		Swapper swapper = new Swapper();
		swapper.createEnviroment();
		swapper.swapAll();
	}//main

	public void createEnviroment() {
		RobotWorld world = RobotWorld.load("swap.txt");
		robot = new Robot(world.getNumRows() - 2, 2, Robot.NORTH, world);
		robot.setDelay(100);
	}//createEnviroment

	//swapping colours on all across cells in the corridor
	//before: the robot is located in beginning of the corridor, facing the corridor
	//after: the robot has the same location and facing the same direction
	public void swapAll() {
		boolean finished = false;
		while (!finished) {
			swapTwoCells();
			if (robot.frontIsClear())
				robot.move();
			else
				finished = true;
		}
		returnToStartPosition();
	}//swapAll

	//swapping colours of two across cells
	//before: robot in the corridor facing the corridor
	//after: robot in the corridor facing the corridor
	private void swapTwoCells() {
		if (leftCellIsDark() != rightCellIsDark())
			changeColors();
	}//swapTwoCells

	//return: true if the cell on left side is dark, otherwise false
	//before: robot in the corridor facing the corridor
	//after: robot in the corridor facing the corridor
	private boolean leftCellIsDark() {
		robot.turnLeft();
		robot.move();
		boolean isDark = robot.onDark();
		turnAround();
		robot.move();
		robot.turnLeft();
		return isDark;
	}// leftCellIsDark
```

```
//return: true if the cell on right side is dark, otherwise false
//before: robot in the corridor facing the corridor
//after: robot in the corridor facing the corridor
private boolean rightCellIsDark() {
      turnRight();
      robot.move();
      boolean isDark = robot.onDark();
      turnAround();
      robot.move();
      turnRight();
      return isDark;
}//rightCellIsDark

//change colour of the cell on left side and of the cell on right side
//before: robot in the corridor facing the corridor
//after: robot in the corridor facing the corridor
private void changeColors() {
      changeColourOfLeftCell();
      changeColourOfRightCell();
}//changeColors;

//change colour of the cell on left side
//before: robot in the corridor facing the corridor
//after: robot in the corridor facing the corridor
private void changeColourOfLeftCell() {
      robot.turnLeft();
      robot.move();
      switchColour();
      turnAround();
      robot.move();
      robot.turnLeft();
}//changeColourOfLeftCell

//change colour of the cell on right side
//before: robot in the corridor facing the corridor
//after: robot in the corridor facing the corridor
private void changeColourOfRightCell() {
      turnRight();
      robot.move();
      switchColour();
      turnAround();
      robot.move();
      turnRight();
}//changeColourOfRightCell;

//switch colour of the cell
//before: none
//after: if the cell was dark it has become light, and if the cell was light it has become dark
private void switchColour() {
      if (robot.onDark())
            robot.makeLight();
      else
            robot.makeDark();
}//switchColour
```

```
    //before: robot in end of the corridor, facing the wall
    //after: robot in beginning of the corridor, facing the corridor
    private void returnToStartPosition() {
        turnAround();
        while (robot.frontIsClear()) {
            robot.move();
        }
        turnAround();
    }//returnToStartPosition

    //before: none
    //after: robot has turned 90 degree to right
    public void turnRight() {
        turnAround();
        robot.turnLeft();
    }//turnRight

    //before: none
    //after: robot is facing the opposite direction
    private void turnAround() {
        robot.turnLeft();
        robot.turnLeft();
    }//turnAround
}//Swapper
```