Нові інструкції

- Використання базових інструкцій добре.
- Які інструкції вміє виконувати наш робот?
- Але як доповнити можливості нашого робота?
- Яку інструкцію напишемо першою, чого нам не вистачає?
- Сподіваюсь повернутися направо ☺️

Перша інструкція.

- Нам потрібен синтаксис для створення нової інструкції.
- Нехай інструкція буде мати назву turnRight, яку ми зможемо викликати коли нам потрібно повернути робота направо.
- Відкриємо нашу першу програму FirstProgram.
- В нас вже є написаний нами метод (інструкція).
- Що це за метод?
- run()

Перший метод

- Давайте напишемо наш новий метод.
- private void turnRight(){
- …
- }

- Питання?
- Які ще команди ми можемо придумати?

SuperKarel

- Тепер давайте почнемо робити, щось складніше.
- Виявляється в нас є SuperKarel.
- Це той же робот але з якого зняті тренувальні колеса.
- В нього вже існують методи turnRight, turnAround і багато інших.

SuperKarel

- Ура, ми знаємо всі команди.
- Ми вміємо створювати нові методи.
- Ми СУПЕР програмісти ☺️.
- Давайте трохи пофілософствуємо тепер.
- Чи є якісь дії які ви робите кожен день?

Повторювані дії

- Що нам дає виконання однакових дій (повторення)?
- Ви робите одні й ті ж самі дії (в комплексі).
- Повернемося до робота.
- Припустимо нам потрібно зробити наступні дії:
  - зробити три кроки вперед
- Ми можемо написати три інструкції.
- А потім знову …
- А якщо нам потрібно повторити дії 20 разів?
- Що писати 20 викликів?

Цикл for

- Для цих цілей в нас є цикл for.
- Давайте розглянемо синтаксис.
  - for (int i=0;i<3;i++){
    - turnLeft();
  - }
- І що зробить наш цикл?
- Питання?

Цикл While

- Гарна ситуація, коли ми знаємо скільки разів ми маємо виконати певні дії.
- Але, що робити коли ми цього не знаємо?
- Ви можете сказати, що ви завжди знаєте скільки разів і що ви маєте зробити.
- Але давайте розглянемо ситуацію з нашим Керолом і стіною…
- Інколи нам потрібна конструкція – “Роби щось поки якась умова істинна”
- У випадку з роботом … “Йди прямо поки не наштовхнешся на стіну”

Цикл While

- І як же написати таку умову?
- В нас для цього існує цикл while.
- Синтаксис виглядає наступним чином:
  - while(умова){
  - …
  - }
- Питання?
- Гарне питання звідки взяти умову?
- Якщо ми розглядаємо нашого робота, то в нього є сенсори.
- Що ми можемо одразу перевірити, frontIsClear()
- Давайте розберемо програму SecondProgram і її роботу в трьох світах.

Цикли

- Питання?
- Давайте оформимо метод йти до стіни.
- Я думаю в вас має бути питання, які всі умови, що я можу перевіряти, де вони всі знаходяться?

Сенсори

Сенсор	Протилежний сенсор	Що перевіряється
frontIsClear()	frontIsBlocked()	Чи вільно попереду?
leftIsClear()	leftIsBlocked()	Чи вільно зліва?
rightIsClear()	rightIsBlocked()	Чи вільно справа?
beepersPresent()	noBeepersPresent()	Чи присутні біпери там де я стою?
beepersInBag()	noBeepersInBag()	Чи присутні біпери в мішку?
facingNorth()	notFacingNorth()	Чи стою я обличчям на північ?
facingEast()	notFacingEast()	Чи стою я обличчям на схід?
facingSouth()	notFacingSouth()	Чи стою я обличчям на південь?
facingWest()	notFacingWest()	Чи стою я обличчям на захід?

Умова

- Інколи нам не потрібен цикл.
- Інколи нам потрібно перевірити одну умову і виконати якусь дію.
- Що нам робити?
  - if (умова) {
    - …
  - }
- Давайте напишемо якийсь приклад.

If , else

- Інколи нам в залежності від умови потрібно виконати або одну або іншу дію.
- Що робити тоді?
  - if(умова){
  - …
  - }
  - else {
  - …
  - }
- Давайте спробуємо написати приклад.
- Питання?

Програма

- Давайте напишемо велику програму де будемо використовувати всі наші знання.
- Напишемо програму в якій Керол перестрибує будинки.
- Умови:
  - світ має розмірність 9xN
  - будинки можуть бути будь-якої висоти
  - будинки розташовуються на будь якій відстані один від одного але містять обов’язково порожню клітинку

- Питання?