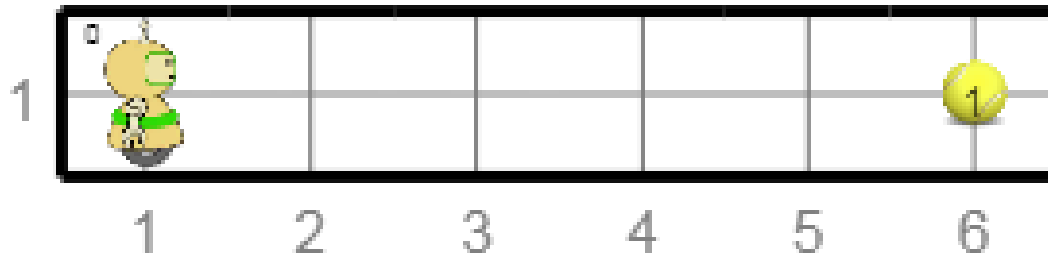


# Програмирање

## Понављање

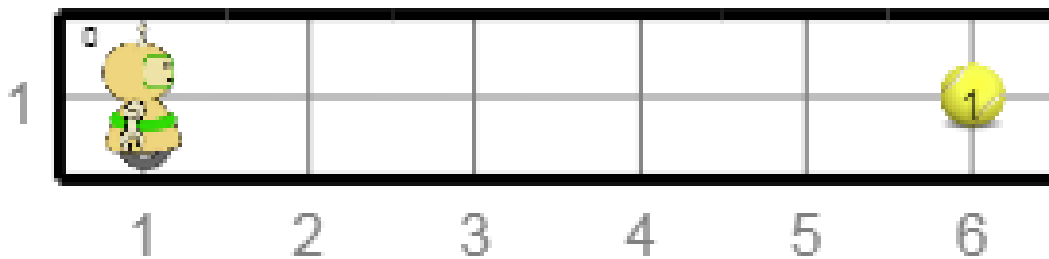
# Задатак

- У наредном лавиринту постоји 5 поља испред робота. Потребно је да робот дође до лоптице и да је затим покупи.



- `napred()` - помери се једно поље напред,
- `levo()` - окрени се 90 степени налево (у смеру супротном казаљки на сату),
- `desno()` - окрени се 90 степени надесно (у смеру казаљке на сату),
- `uzmi()` - покупи лоптицу са поља на којем се налазиш,
- `ostavi()` - спусти лоптицу на поље на којем се налазиш,

# Решење



```
from karel import *  
napred()  
napred()  
napred()  
napred()  
napred()  
uzmi()
```

# Врсте циклуса

- Постоји више врста циклуса (понављања, итерација) `for` и `while`:
  - када знамо тачно колико пута желимо да се нешто понови (*for*)
  - обезбеђује да се наредбе понављају све док је неки услов испуњен (*while*).

# Задачић - решење

```
from karel import *
```

```
napred()
```

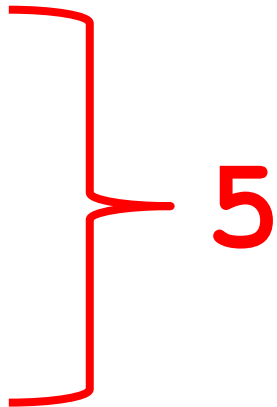
```
napred()
```

```
napred()
```

```
napred()
```

```
napred()
```

```
uzmi()
```



```
from karel import *
```

```
for i in range(5):
```

```
    napred()
```

```
uzmi()
```

# for

Бројач *i* добија почетну вредност 0

Кључа реч, дефинише понављање са бројачем

**for** **i** **in** **range**(**n**) :  
**naredbe()**

Бројач

Уграђена функција

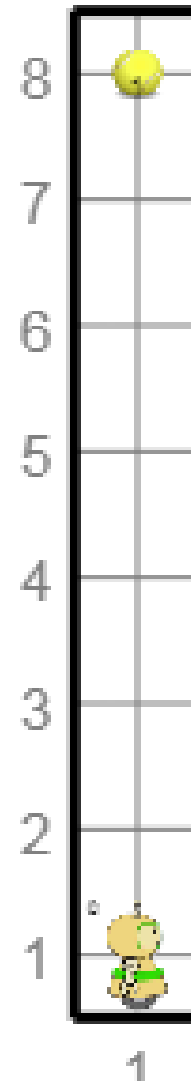
одређује колико ће пута бити понављена петља

Наредбе које се понављају (за њих се каже да чине *тело* петље) се обавезно морају увући у односу на наредбу **for** (обично помоћу неколико размака)

| Наредба       | Генерисане вредности |
|---------------|----------------------|
| range(10)     | 0,1,2,3,4,5,6,7,8,9  |
| range(1,10)   | 1,2,3,4,5,6,7,8,9    |
| range(3,7)    | 3,4,5,6              |
| range(2,15,3) | 2,5,8,11,14          |
| range(9,2,-1) | 9,8,7,6,5,4,3        |

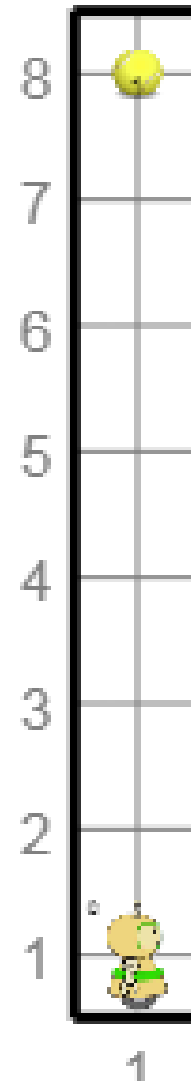
## Задатак 2

- У лавиринту постоји 7 поља испред робота. Коришћењем петље учини да робот дође до лоптице и да је затим покупи.



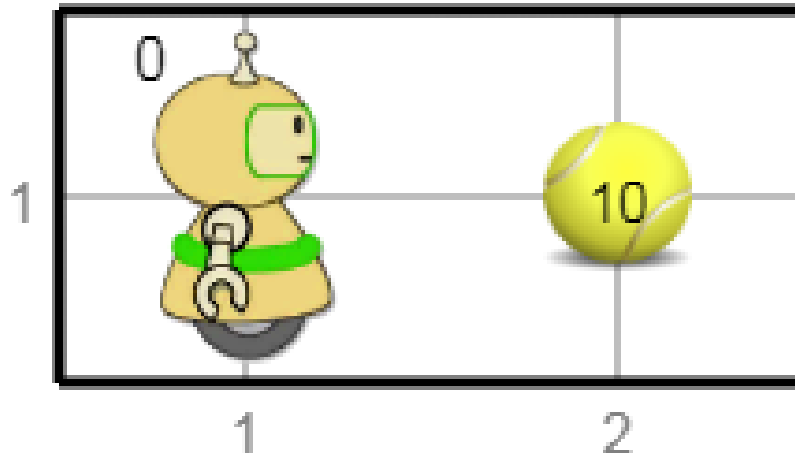
## Задатак 2

```
from karel import *  
levo()  
for i in range(7):  
    napred()  
uzmi()
```



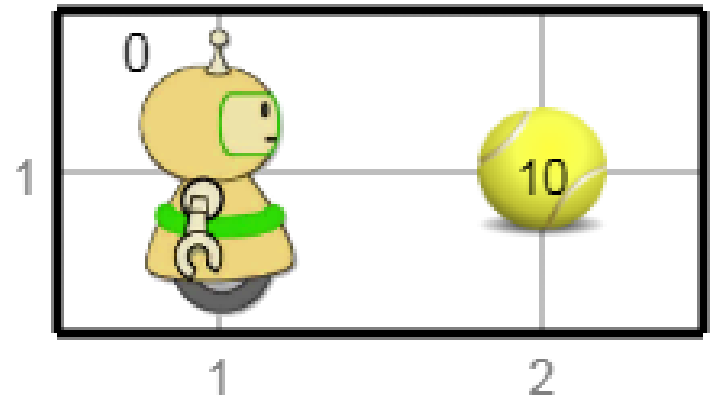
# Задатак 3

- Испред робота се налази 10 лоптица.  
Напиши програм којим робот купи све те лоптице.



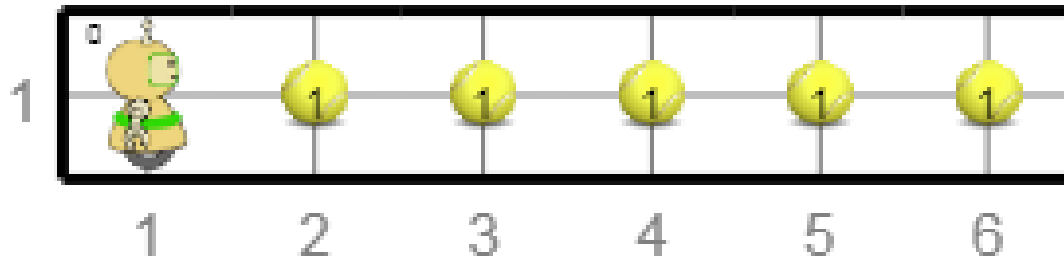
# Задатак 3

```
from karel import *  
napred()  
for i in range(10):  
    uzmi()
```



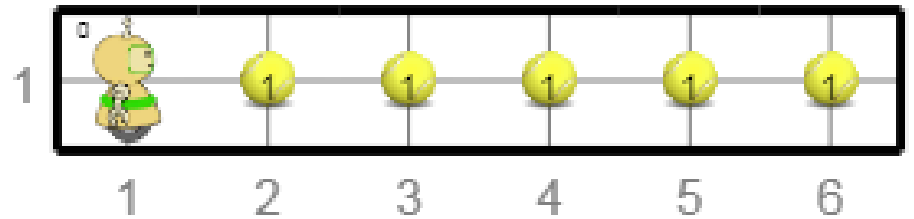
## Задатак 4

- Напиши програм у којем робот купи лоптице на пет поља испред себе.



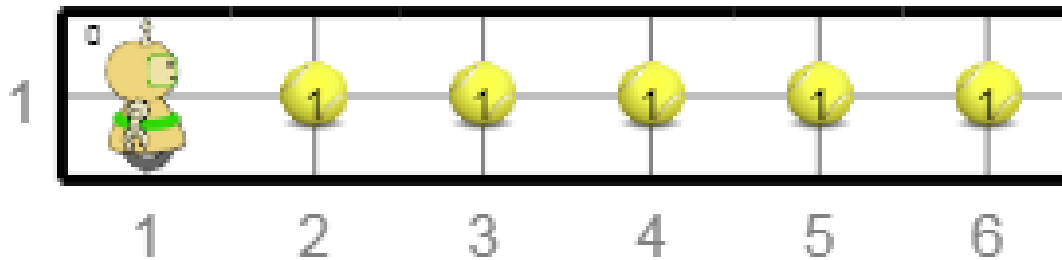
# Задатак 4

```
from karel import *  
for i in range(5):  
    napred()  
    uzmi()
```



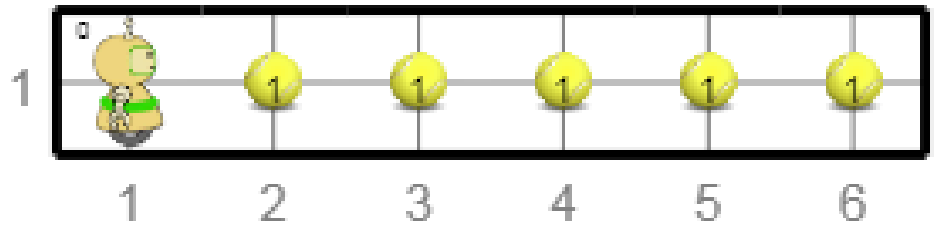
# Задатак 5

- Напиши програм у којем робот купи лоптице на пет поља испред себе, а затим се враћа на почетно поље када покупи лоптице



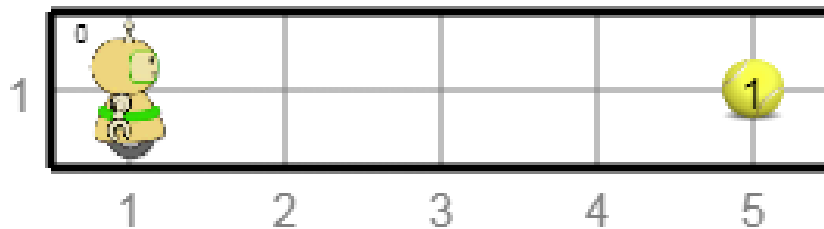
# Задатак 5

```
from karel import *  
for i in range(5):  
    napred()  
    uzmi()  
levo()  
levo()  
for j in range(5):  
    napred()
```



# Задатак 6

- \*Лавиринт је зачаран и не зна се колико тачно поља постоји између робота и лоптице. Напиши програм тако да робот и у таквом лавиринту увек стиже до лоптице и узима је.



# Задатак 6

- желимо роботу да наредимо да иде напред док год је то могуће и након тога да покупи лоптицу.
- `mozeNapred()` проверава да ли се робот може померити напред, тј. да ли се испред њега налази зид.

- `moze_napred()` - проверава да робот може да се помери напред (да ли испред њега постоји зид),
- `broj_loptica_na_polju()` - враћа број лоптица на пољу на ком се робот налази,
- `ima_loptica_na_polju()` - проверава да ли на пољу на ком се робот налази има лоптица,
- `broj_loptica_kod_sebe()` - враћа број лоптица које робот тренутно има код себе,
- `ima_loptica_kod_sebe()` - проверава да ли робот тренутно има лоптица код себе.

# while

- Контролише се понављање, тј. неке наредбе се понављају све док је испуњен неки услов.

Кључа реч, дефинише понављање са условом



while

Логички услов



uslov:

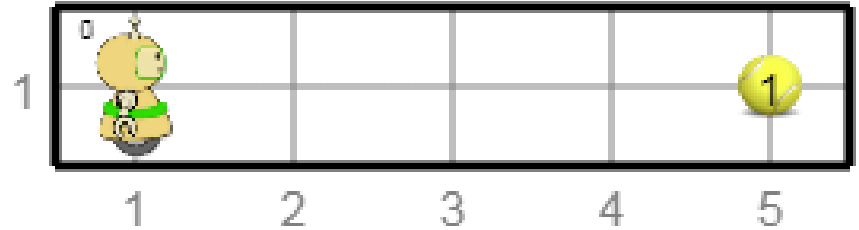
naredbe ()



Наредбе које се понављају (за њих се каже да чине *тело* петље) се обавезно морају увући у односу на наредбу while (обично помоћу неколико размака)

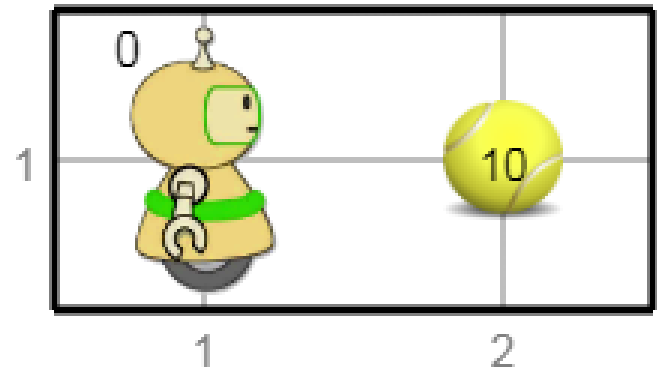
# Задатак 6

```
from karel import *  
while mozeNapred() :  
    napred()  
uzmi()
```



# Задатак 7

- Наредни лавиринт је зачаран и не зна се колико тачно има лоптица испред робота. Напиши програм којим робот купи све те лоптице.
  - Коришћењем `for`
  - Коришћењем `while`

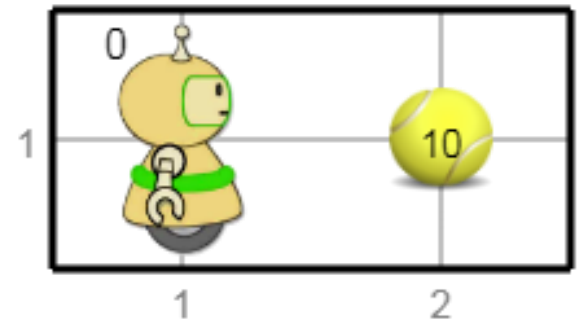


# Задатак 7

– Коришћењем `for` (одреди колико је лоптица на пољу на ком се робот налази. Број лоптица на пољу се може одредити позивом `broj_loptica_na_polju()`.)

```
from karel import *  
napred()  
for i in range(broj_loptica_na_polju()):  
    uzmi()
```

```
from karel import *  
napred()  
for i in range(10):  
    uzmi()
```

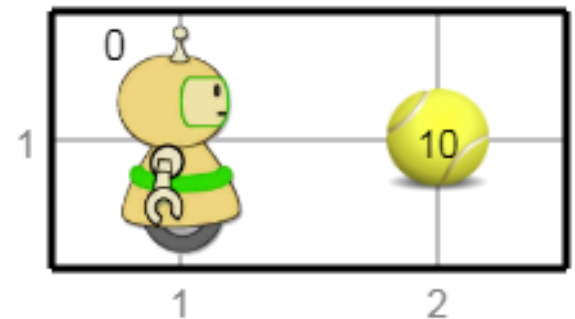


- `broj_loptica_na_polju()` - враћа број лоптица на пољу на ком се робот налази,

# Задатак 7

– Коришћењем `while` (лоптице се купе све док их има на пољу)

```
from karel import *  
napred()  
while ima_loptica_na_polju():  
    uzmi()
```



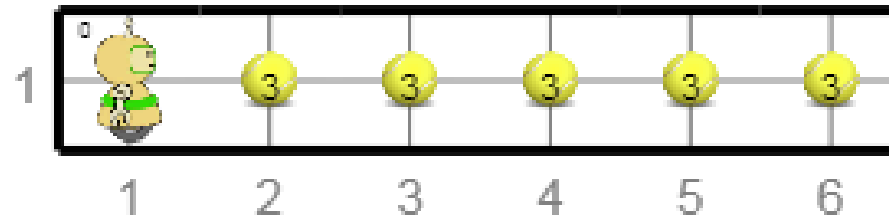
- `ima_loptica_na_polju()` - проверава да ли на пољу на ком се робот налази има лоптица

# Угњеждене петље

- Наредбе понављања се могу комбиновати тако да се једна петља позива унутар друге, може бити неограничен број овако угњеждених петљи, а могу се и комбиновати различите врсте петљи `for` и `while`.

# Пример угњеждене петље

- На сваком од пет поља испред робота налазе се по три лоптице. Напиши програм на основу којег робот купи све те лоптице.



# Пример угњеждене петље

```
from karel import *  
for i in range(5):  
    napred()  
    for j in range(3):  
        uzmi()
```

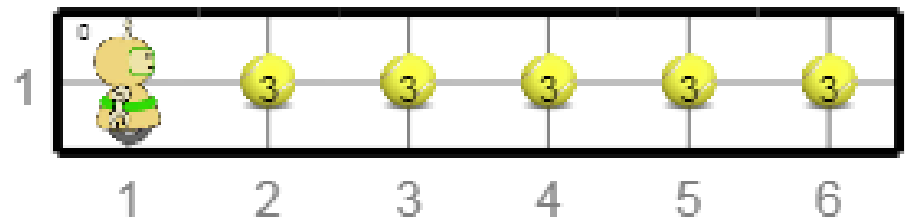


Бројач унутрашње  
петље одговара броју  
лоптица



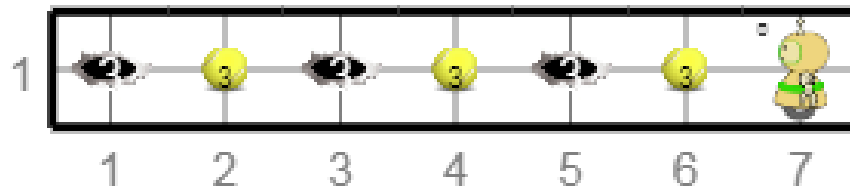
Бројач спољашње петље  
одговара броју корака које  
робот направи

Цела унутрашња петља  
понавља се 3 пута, по  
једном за сваку вредност  
бројача спољашње петље.



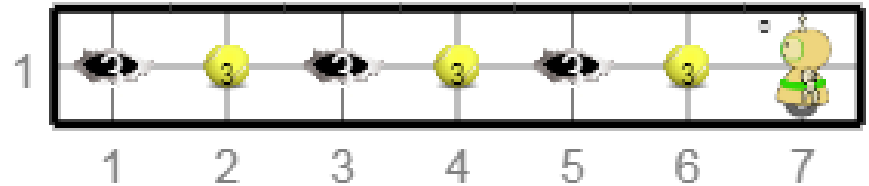
# Пример

- Помози роботу да покупи све лоптице премести у рупе испред њих. Лавиринт је увек исти (на три поља испред робота се налази по три лоптице).



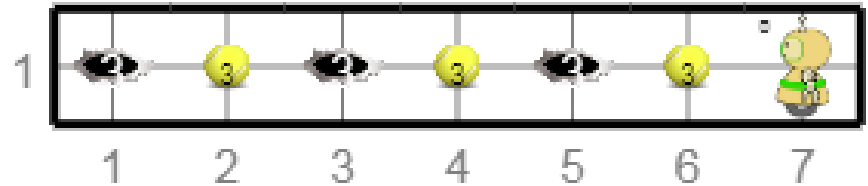
# Пример

```
from karel import *  
for i in range(3):  
    napred()  
    for j in range(3):  
        uzmi()  
    napred()  
    for j in range(3):  
        ostavi()
```



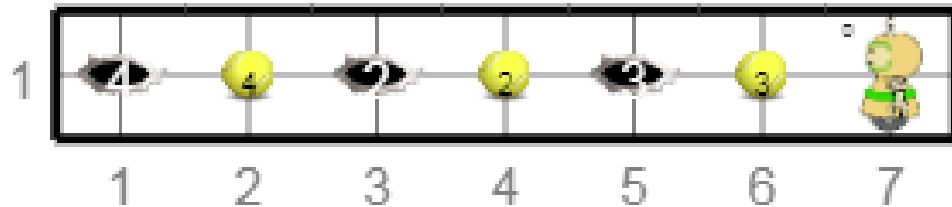
# Пример

```
from karel import *  
for i in range(3):  
    napred()  
    for i in  
range(broj_loptica_na_polju()):  
        uzmi()  
    napred()  
    for i in  
range(broj_loptica_kod_sebe()):  
        ostavi()
```



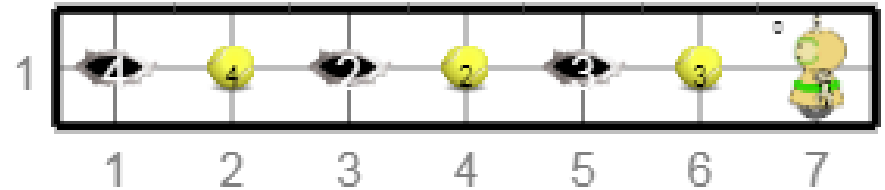
# Задатак

- Лавиринт је зачаран, робот не зна унапред колико ће лоптица бити испред њега  
Помози роботу да покупи све лоптице премести у рупе испред њих.



# Задатак

```
from karel import *  
while moze_napred():  
    napred()  
    while ima_loptica_na_polju():  
        uzmi()  
    napred()  
    while ima_loptica_kod_sebe():  
        ostavi()
```

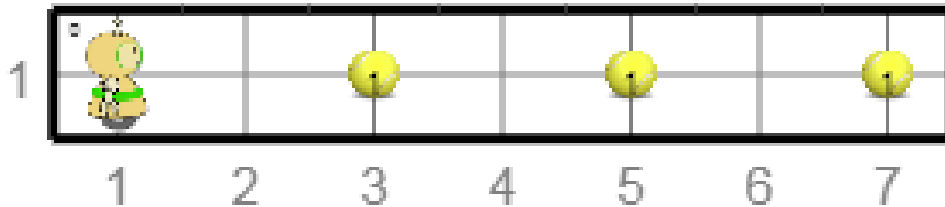


- `napred()` - помери се једно поље напред,
- `levo()` - окрени се 90 степени налево (у смеру супротном казаљки на сату),
- `desno()` - окрени се 90 степени надесно (у смеру казаљке на сату),
- `uzmi()` - покупи лоптицу са поља на којем се налазиш,
- `ostavi()` - спусти лоптицу на поље на којем се налазиш,
- `moze_napred()` - проверава да робот може да се помери напред (да ли испред њега постоји зид),
- `broj_loptica_na_polju()` - враћа број лоптица на пољу на ком се робот налази,
- `ima_loptica_na_polju()` - проверава да ли на пољу на ком се робот налази има лоптица,
- `broj_loptica_kod_sebe()` - враћа број лоптица које робот тренутно има код себе,
- `ima_loptica_kod_sebe()` - проверава да ли робот тренутно има лоптица код себе.

## Додатни задаци

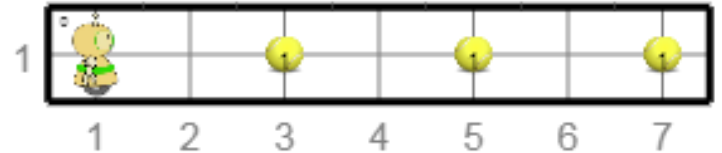
# Додатни задаци

- Задатак 1: Помози роботу да покупи три лоптице испред себе. Напиши програм са петљом.



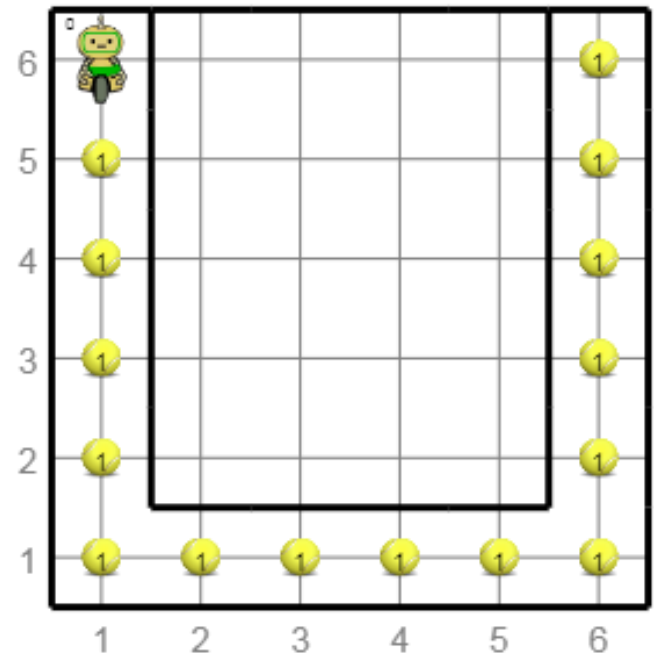
# Додатни задаци

```
from karel import *  
for i in range(3):  
    napred()  
    napred()  
    uzmi()
```



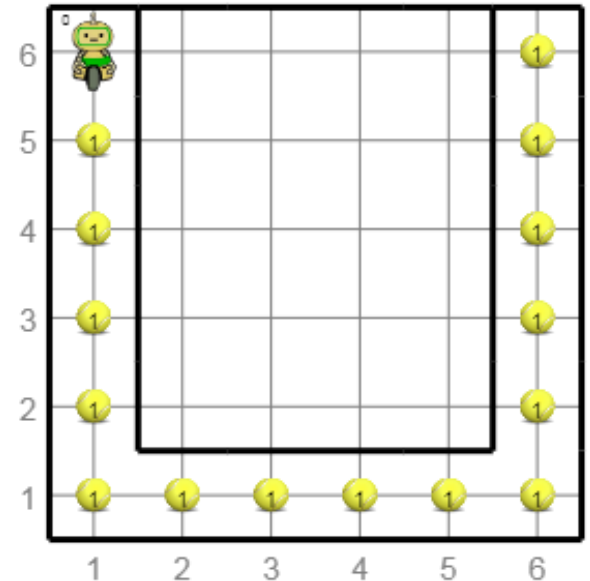
# Додатни задаци

- Задатак 2: Помози роботу да покупи све лоптице. Лавиринт је зачаран и број степеница се мења приликом сваког покретања програма.



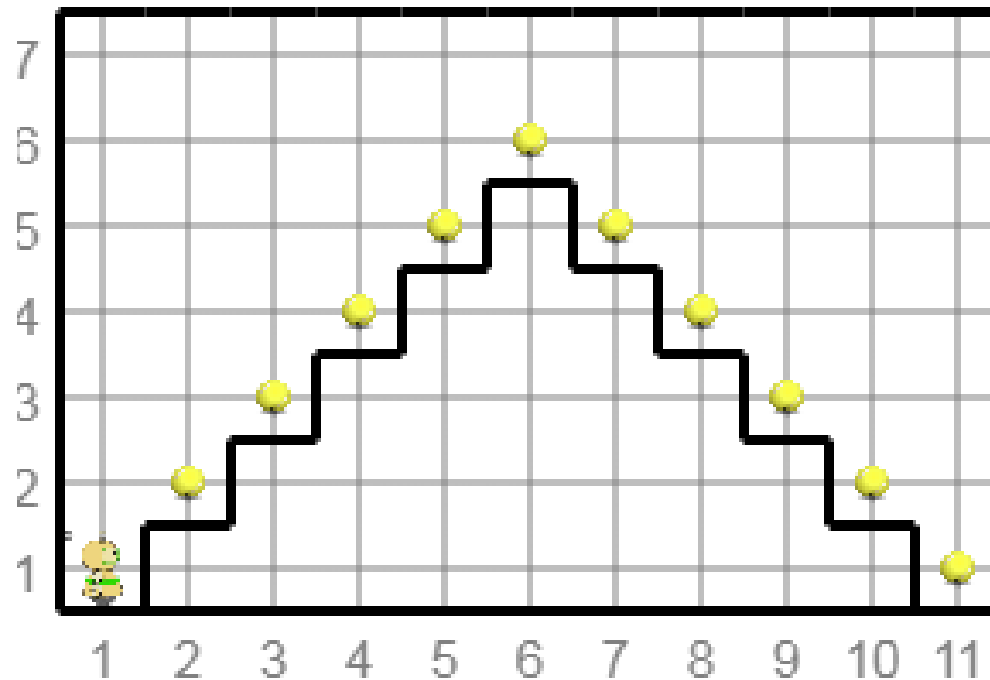
# Додатни задаци

```
from karel import *  
for i in range(3):  
    while moze_napred():  
        napred()  
        uzmi()  
levo()
```



# Додатни задаци

- Задатак 3: Помози роботу да покупи све лоптице.



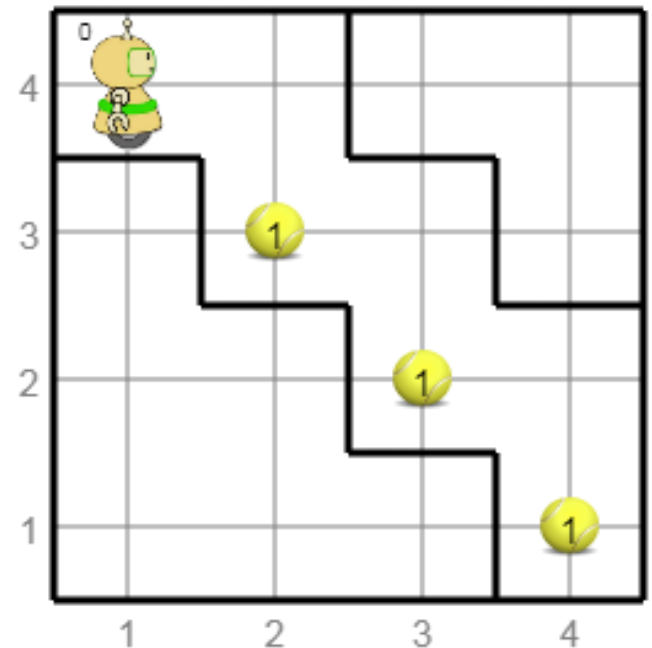
# Додатни задаци

```
from karel import *  
for i in range(5):  
    levo()  
    napred()  
    desno()  
    napred()  
    uzmi()  
for i in range(5):  
    napred()  
    desno()  
    napred()  
    uzmi()  
    levo()
```



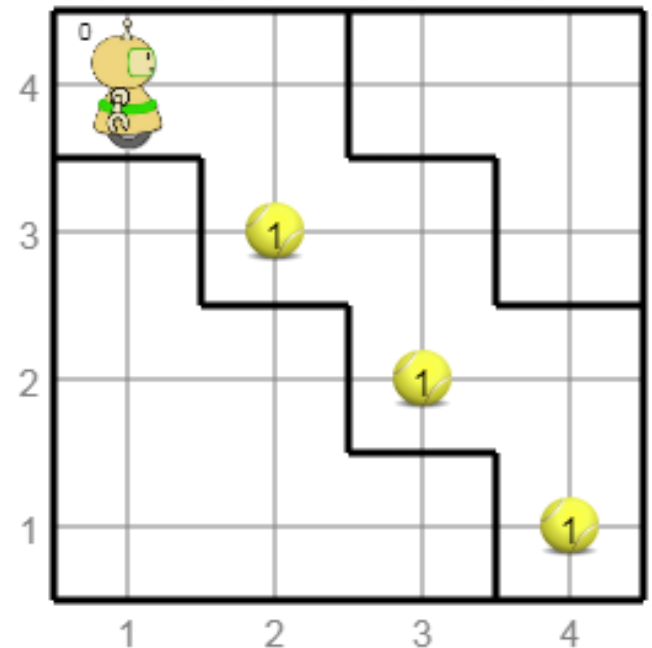
# Додатни задаци

- Задатак 4: Помози роботу да покупи све лоптице. Лавиринт је зачаран и број степеница се мења приликом сваког покретања програма.



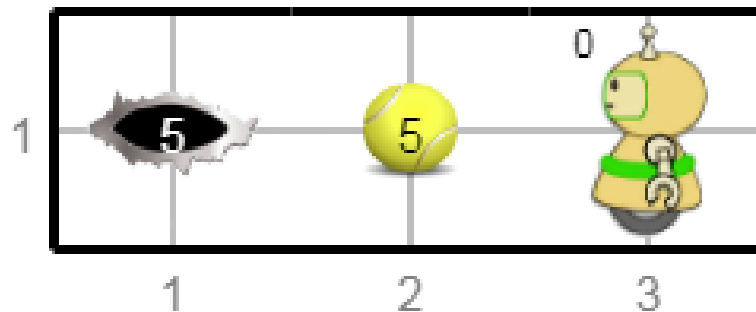
# Додатни задаци

```
from karel import *  
while  
moze_napred() :  
    napred()  
    desno()  
    napred()  
    uzmi()  
    levo()
```



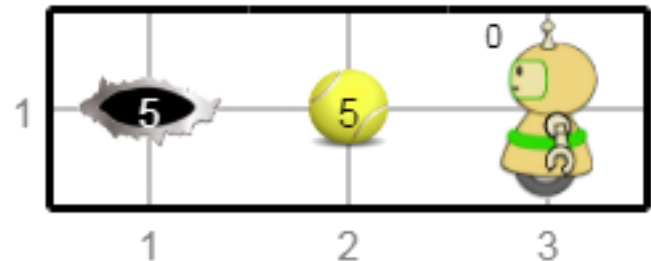
# Додатни задаци

- Задатак 5: Помози роботу да покупи свих пет лоптица и премести у рупу испред њих.



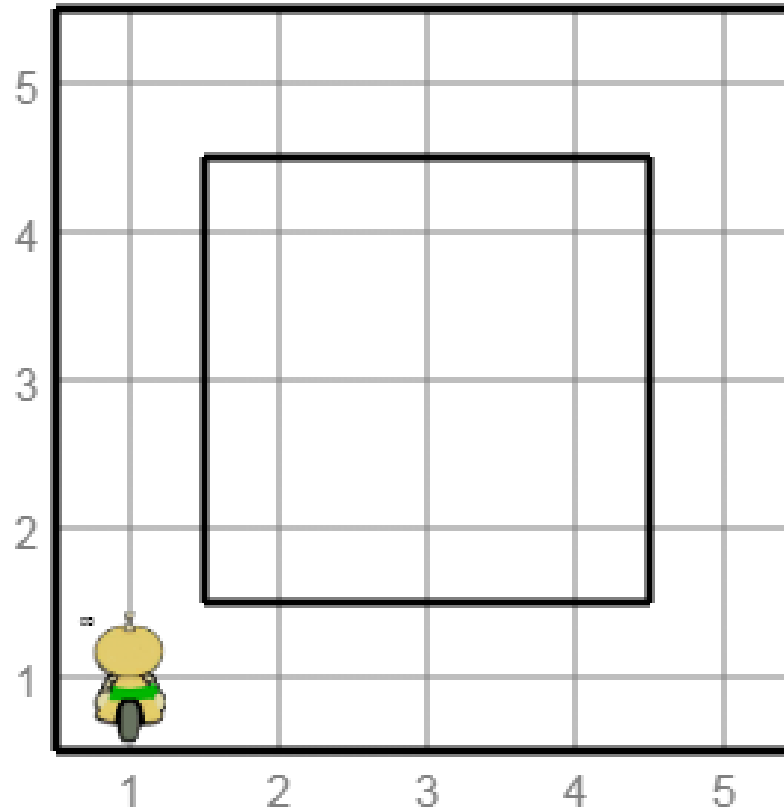
# Додатни задаци

```
from karel import *  
napred()  
for j in range(5):  
    uzmi()  
napred()  
for j in range(5):  
    ostavi()
```



# Додатни задаци

- Задатак 6: Помози роботу да постави лоптице дуж целог лавиринта.



# Додатни задаци

```
from karel import *  
for i in range(4):  
    while moze_napred():  
        napred()  
        ostavi()  
desno()
```

