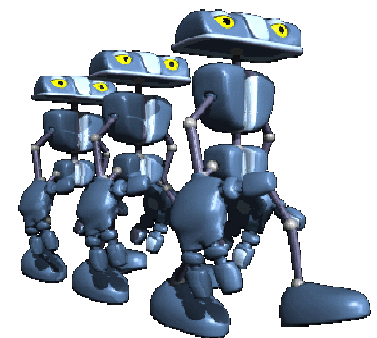


ЛАБОРАТОРИСКА ВЕЖБА 7:

**ПРЕТСТАВУВАЊЕ НА ЈАЗИЦИТЕ КОИ
ЌЕ ТРЕБА ДА СЕ КОРИСТАТ ВО
ПРОЕКТИТЕ**



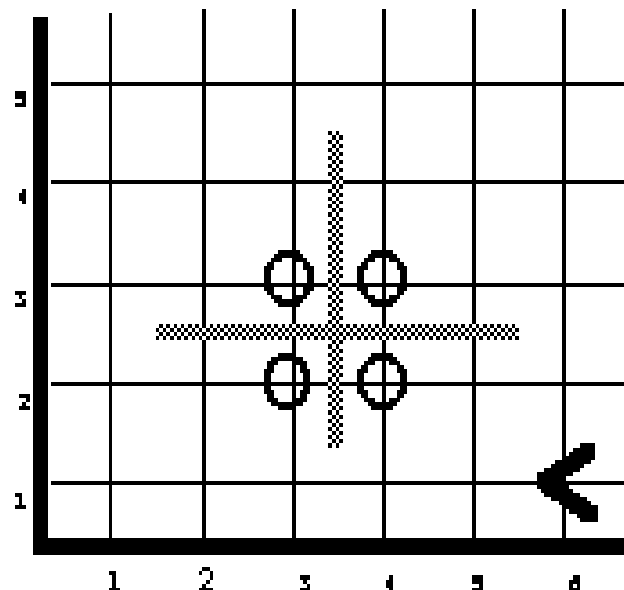
СОДРЖИНА

- Повторување: Дефинирање на околина
- Повторување: Дефинирање на јазикот RoboL
- Дефинирање на јазикот RoboL+
 - Нова, подобрена верзија на RoboL



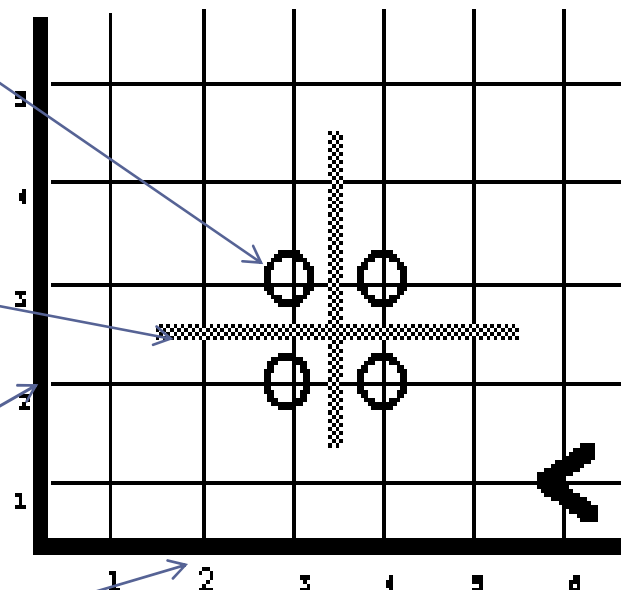
ДЕФИНИРАЊЕ НА ОКОЛИНА

- Околина е $m \times n$ поле
 - m и n мора да се зададени и мора да бидат позитивни природни броеви
 - n паралелни улици (исток - запад)
 - m паралелни булевари (север - југ)
- Пресекот на некоја улица со некој булевар го нарекуваме **агол** (раскрсница).



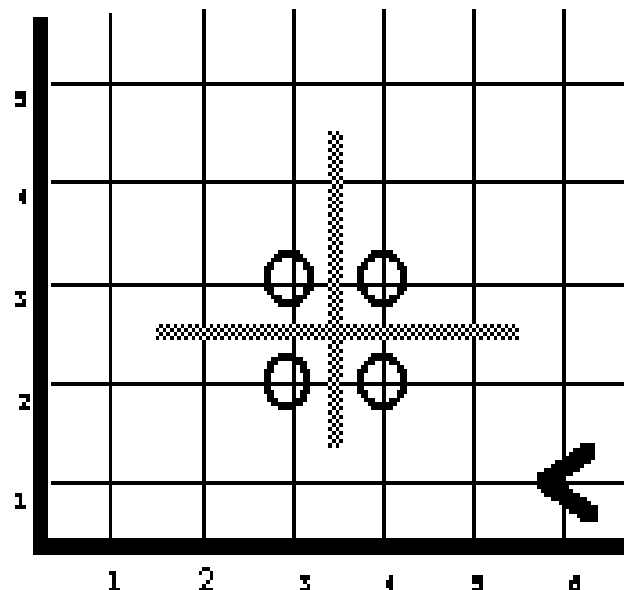
ДЕФИНИРАЊЕ НА ОКОЛИНА

- На аглите може да се поставуваат **ознаки** (раскрсница со
 - Може да има, но и да нема ознаки
 - Може да се постават на секој од нивните агли
 - Ознака : **кружница**
- На дел од булеварот меѓу две соседни улици и на дел од улицата меѓу два соседни булевари може да се стави **сид**(прекин на патот)
 - Може да има, но и да нема сидови
 - Сид може да има и на една улица пред првиот и по последниот булевар
 - Сид може да има и на еден булевар пред првата и по последната улица
 - Ознака : **правоаголник**



ДЕФИНИРАЊЕ НА ОКОЛИНА

- Во секоја околина има еден робот за кој мора да се дефинираат
 - Неговата позиција (на кој агол стои роботот, т.е. На пресекот на која улица и кој булевар)
 - Неговата насока (Исток, Запад, Север или Југ)
- Ознака : >



ДЕФИНИРАЊЕ НА ОКОЛИНА

- Една програма со која се дефинира околината треба да ги изгледа како прикажаната скица која е десно прикажана
 - Треба да се запази редоследот
- Програмите во овој јазик ќе ги зачувуваме со наставка `.env`
- Во програмата задолжително мора да има дел во кој ќе се дефинира околината и во кој ќе се одреди каде е робот
- Во програмата може да нема дел за дефинирање на сидови и ознаки (ако такви не постојат во околината)

*Дефинирање на околина
(задолжително)*

Дефинирање на сидови (ако има)

Дефинирање на ознаки (ако има)

*Дефинирање на роботот
(задолжително)*

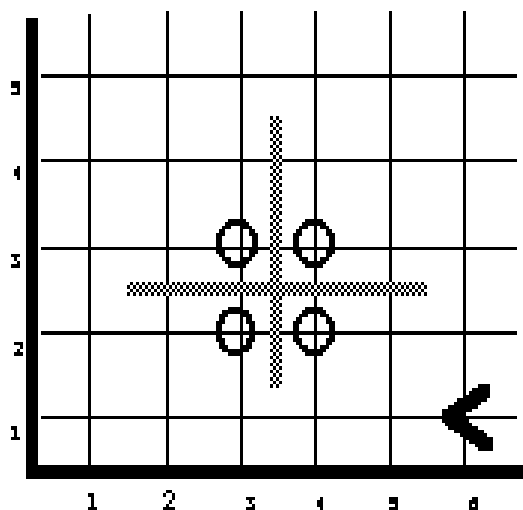


ДЕФИНИРАЊЕ НА ОКОЛИНА

○ Дефинирање на околина

- m, n се цели позитивни броеви
 - m -булевари и n -улицы

Околина(m, n)



Околина(6,5)

Забелешка:

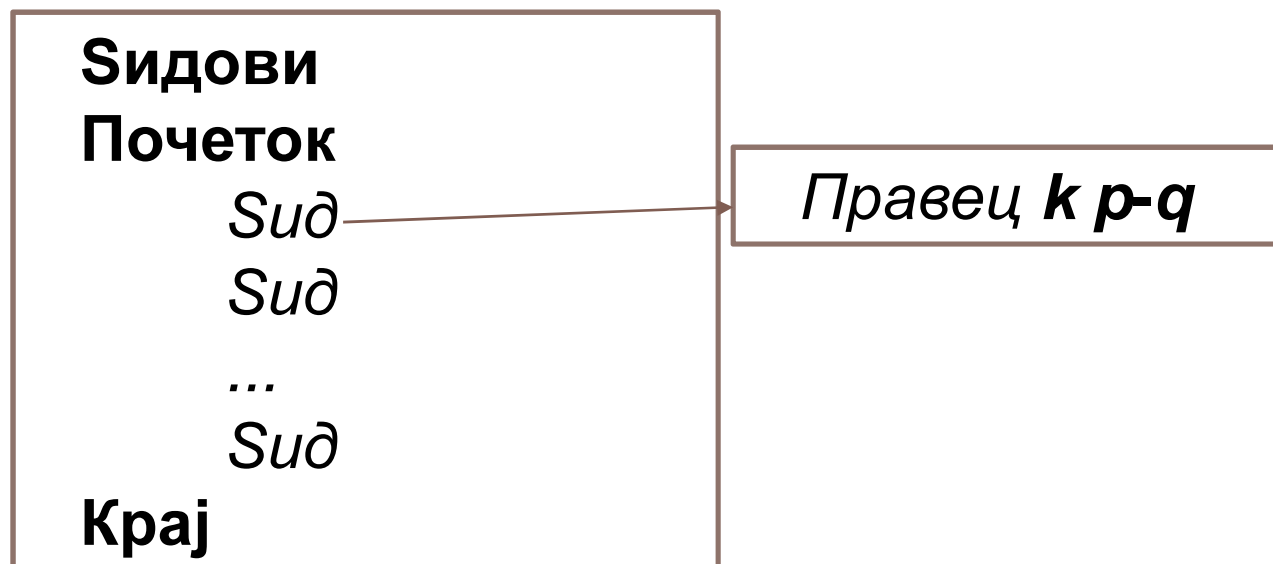
- Задебелени букви – терминали
 - Накосени букви – нетерминали
 - Накосени и задебелени букви – терминали кои во јазикот може да имаат различна вредност.
- Пример: броеви, идентификатор.

ДЕФИНИРАЊЕ НА ОКОЛИНА

○ Дефинирање на сидови

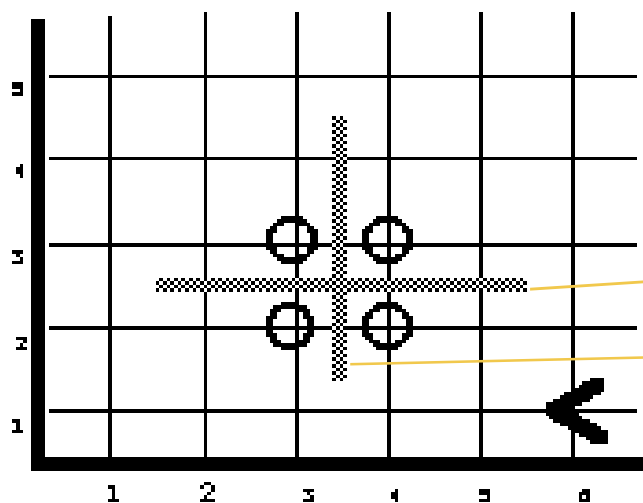
- Сид во правец исток – запад се наоѓа помеѓу две соседни улици (k), а сече еден или повеќе булевари (од p до q)
- Сид во правец север – југ се наоѓа помеѓу два соседни булевари (k), а сече една или повеќе улици (од p до q)

○ $\text{Правец} \in \{\text{ИЗ}, \text{СЈ}\}$ и $k, p, q \in N_0$



ДЕФИНИРАЊЕ НА ОКОЛИНА

- Дефинирање на сидови



Сидови

Почеток

ИЗ 2 2-5

СЈ 3 2-4

Крај

ДЕФИНИРАЊЕ НА ОКОЛИНА

○ Дефиниција на ознаки

- r -тата ознака се наоѓа на пресек на i_r -тиот булевар и j_r -тата улица

Ознаки

Почеток

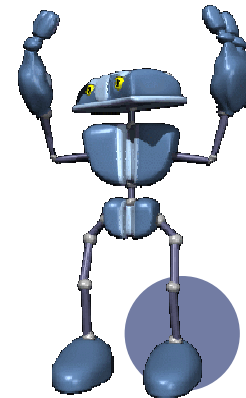
(i_1, j_1)

(i_2, j_2)

...

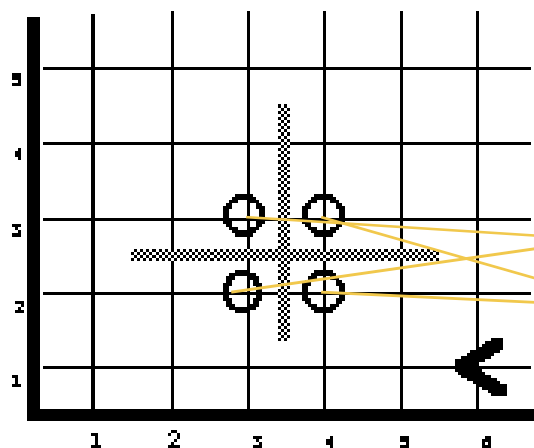
(i_s, j_s)

Крај



ДЕФИНИРАЊЕ НА ОКОЛИНА

○ Дефиниција на ознаки



Ознаки
Почеток

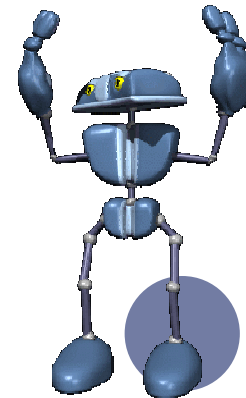
(3,2)

(3,3)

(4,2)

(4,3)

Крај

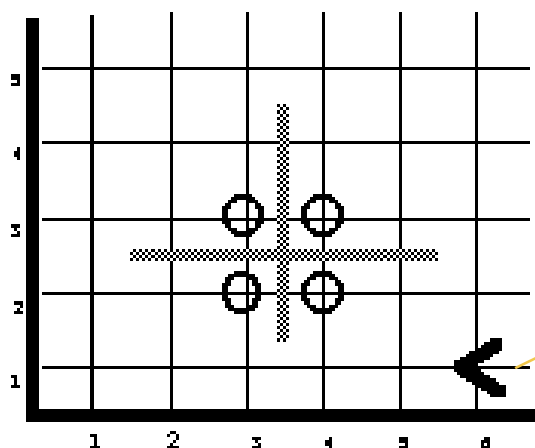


ДЕФИНИРАЊЕ НА ОКОЛИНА

- Дефиниција на поставеноста на роботот

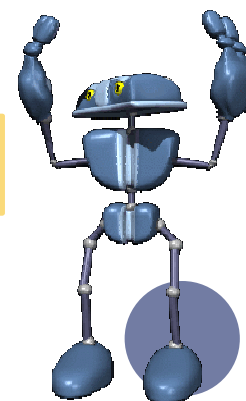
- Позиција: пресек на j -та улица и i -ти булевар
- Свртен во некоја насока (исток, запад, север или југ)

- $Насока \in \{И,З,С,Ј\}$



Робот $Насока(i,j)$

Робот 3(6,1)



ДЕФИНИРАЊЕ НА ОКОЛИНА

○ Ограничувања на броевите

- Сидови

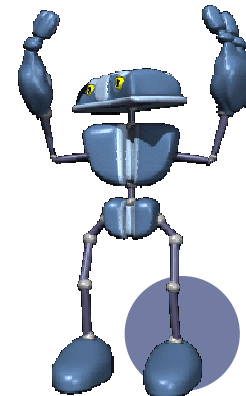
- Ако $Насока=ИЗ$, тогаш $0 \leq k \leq n$ и $1 \leq p \leq q \leq m$
- Ако $Насока=СЈ$, тогаш $0 \leq k \leq m$ и $1 \leq p \leq q \leq n$

- Ознаки

- $1 \leq i \leq m$ и $1 \leq j \leq n \quad \forall r \in \{1, \dots, s\}$

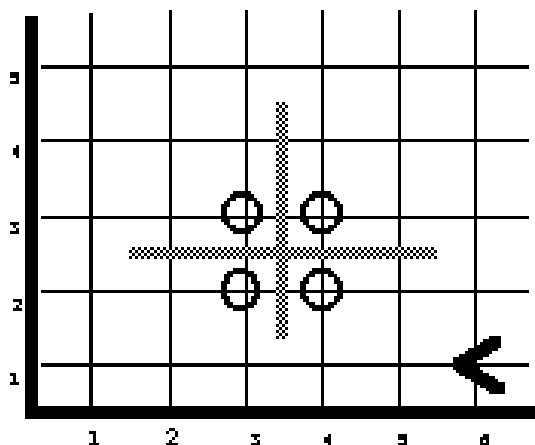
- Робот

- $1 \leq i \leq m$ и $1 \leq j \leq n$,



ДЕФИНИРАЊЕ НА ОКОЛИНА

○ Primer.env



Околина (6,5)

Сидови

Почеток

ИЗ 2 2-5

СЈ 3 2-4

Крај

Ознаки

Почеток

(3,2)

(3,3)

(4,2)

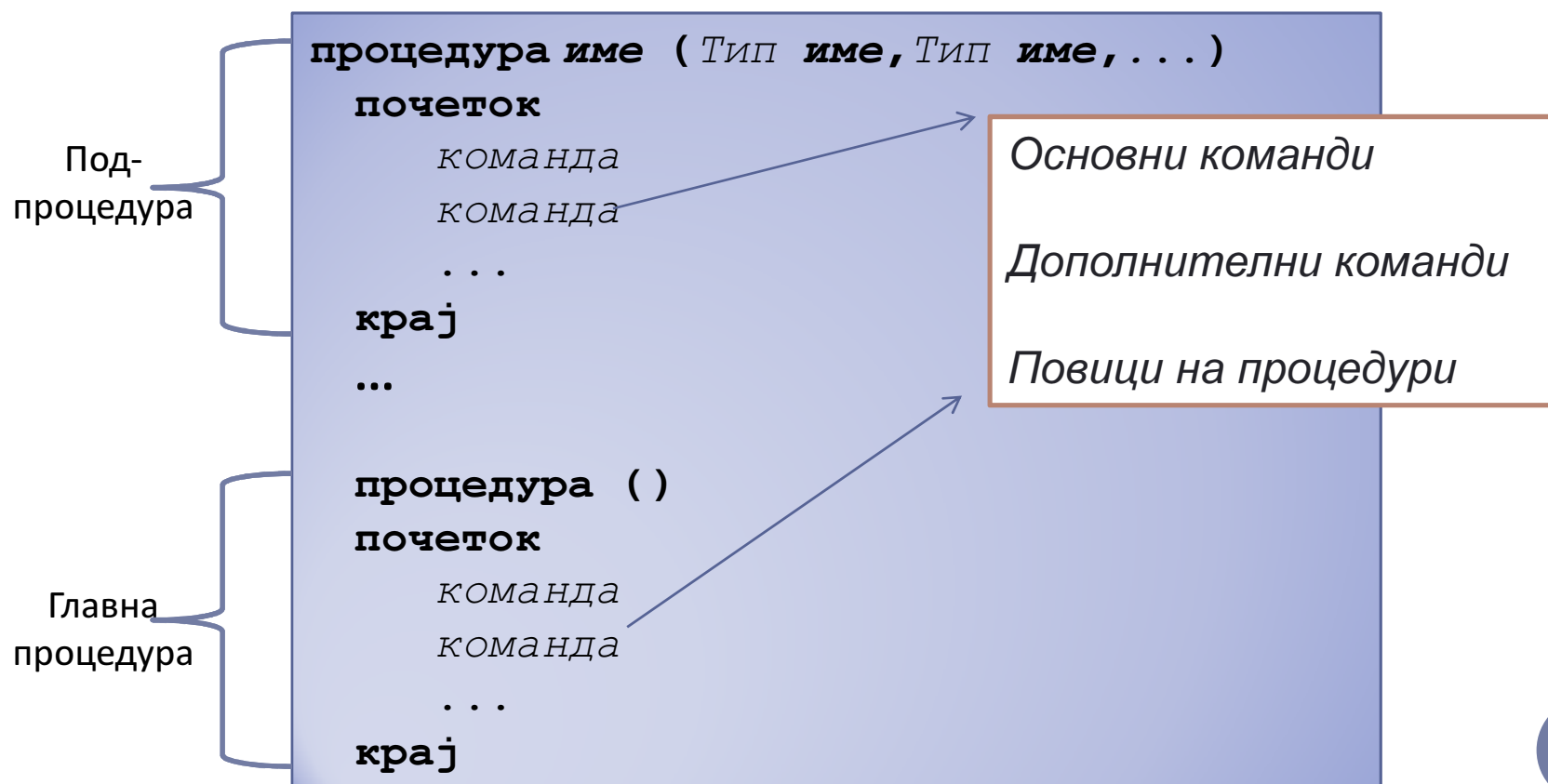
(4,3)

Крај

Робот 3(6,1)

ДЕФИНИРАЊЕ НА ЈАЗИКОТ RoboL

○ Програма и процедури во RoboL



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROVOL

○ Во една програма

- Подпроцедури може да има 0 или повеќе
- Подпроцедурите се еден по друг пред главната процедура
- Главната процедура нема аргументи и е на крајот
- Подпроцедурите може да имаат или на немаат параметри
- Главната процедура нема име
- Подпроцедурите имаат име



ДЕФИНИРАЊЕ НА ЈАЗИКОТ RoboL

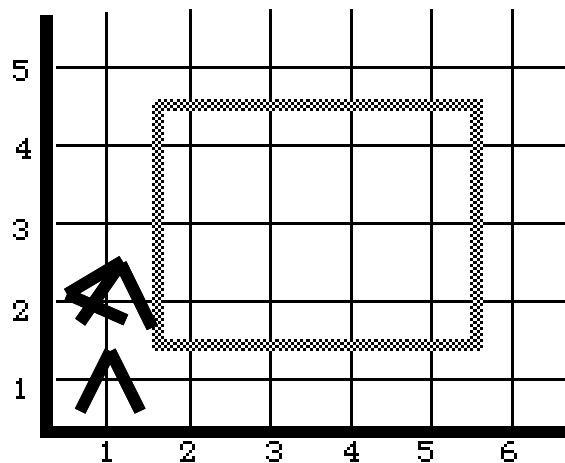
- **име:** (терминал во граматиката на RoboL)
 - Име на променливите и аргументите е составено од
 - Букви
 - Цифри
 - Специјални знаци : „ _ “ , и „ . “
- Аргументите можат да бидат од тип
 - Број
 - Насока
- Нови ознаки
 - БРОЈ- ги вклучува имињата на променливите (**име**) и вредности од тип **број**
 - НАСОКА ги вклучува имињата на променливите (**име**) и вредностите од тип **Насока**=**{И,З,С,Ј}**



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROBOC

○ Основни команди

- Оди
- СвртиЛево
- СвртиДесно



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROVOL

- Дополнителни команди
- Контролни команди за избор

Ако Ознака *основна_команда*

Ако Сид *основна_команда*

Ако НАСОКА *основна_команда*



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROBOC

○ Дополнителни команди

➤ Команди за повторување (со мали измени)

Повторувај БРОЈ пати:

!

команда

команда

...

команда

!

Повторувај до Ознака:

!

команда

команда

...

команда

!

Повторувај до Сид:

!

команда

команда

...

Команда

!

Повторувај до НАСОКА:

!

команда

команда

...

команда

!



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROVOL

- Повик на процедура

***име** (параметар1, параметар2,)*

- Ако се повика процедура а која не постои или со грешни параметри – семантичка грешка
- Параметрите можат да бидат :
 - податок од тип **број**
 - податок од тип **Насока** т.е. една од константите: **И,З,С** и **Ј**
 - имиња на променливи: **име**



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROVOL

○ Пример

```
процедура оди_х_чекори  
(Број х)  
почеток  
повторувај х пати:  
    !  
    оди  
    !  
крај
```

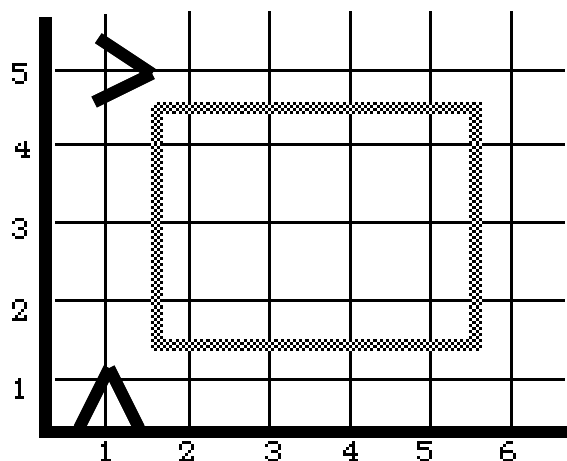
```
процедура ()  
почеток  
оди_х_чекори (4)  
свртиДесно  
оди_х_чекори(2)  
крај
```

```
процедура ()  
почеток  
повторувај 4 пати:  
    !  
    оди  
    !  
свртиДесно  
оди  
оди  
крај
```



ДЕФИНИРАЊЕ НА ЈАЗИКOT ROVOL

- Графички приказ на примерот



ПРОШИРУВАЊЕ НА ЈАЗИКОТ

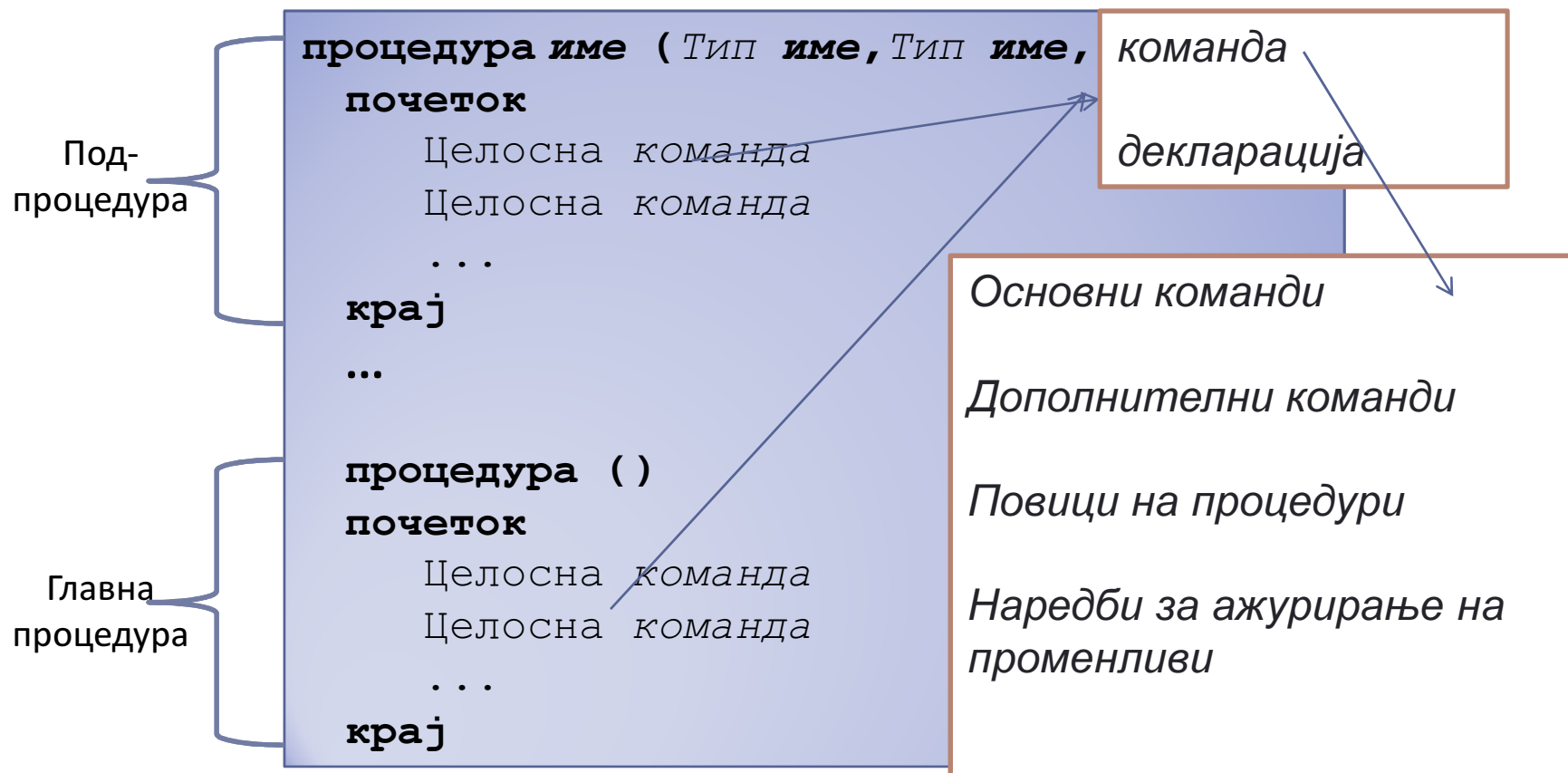
○ Јазикот RoboL+

- Мало проширување на RoboL
- Поекспресивни условни наредби
- Декларирање на променливи
- Оператори
- Услов
- Аритметички операции
- Не е осетлив н мали и големи букви (Нема разлика меѓу **Ако** и **ако**)



ДЕФИНИРАЊЕ НА ЈАЗИКОТ RoboL+

○ Програма и процедури во RoboL+



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROVOL+

○ Во една програма

- Подпроцедури може да има 0 или повеќе
- Подпроцедурите се еден по друг пред главната процедура
- Главната процедура нема аргументи и е на крајот
- Подпроцедурите може да имаат или на немаат параметри
- Главната процедура нема име
- Подпроцедурите имаат име



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROBO L+

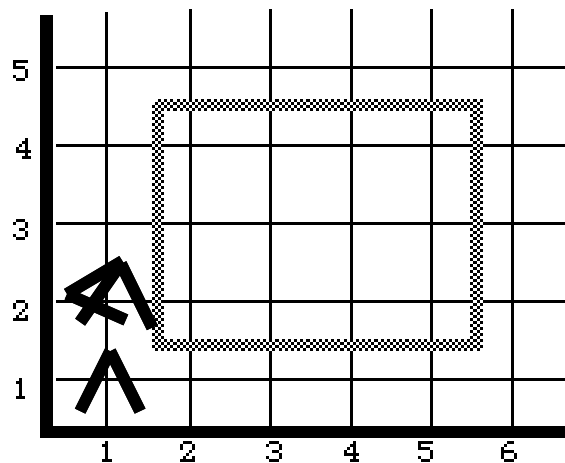
- **име:** (терминал во граматиката на RoboL)
 - Име на променливите и аргументите е составено од
 - Букви
 - Цифри
 - Специјални знаци : „ _ “ , и „ . “
- Аргументите можат да бидат од тип
 - Број
 - Насока
- Ознаките
 - БРОЈ- ги вклучува имињата на променливите (**име**) и вредности од тип **број**
 - НАСОКА ги вклучува имињата на променливите (**име**) и вредностите од тип **Насока**=**{И,З,С,Ј}**



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROVOL+

○ Основни команди

- Оди
- СвртиЛево
- СвртиДесно



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROVOL+

○ Контролни команди за избор

Ако Ознака

!

команда

команда

...

команда

!

Ако Сид

!

команда

команда

...

команда

!

Ќе го дефинираме
понатака

Ако НАСОКА

!

команда

команда

...

команда

!

Ако Услов

!

команда

команда

...

команда

!

ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROBOC+

о Команди за повторување

Повторувај *БРОЈ* пати:

!
команда
команда
...
команда
!

Повторувај до Ознака:

!
команда
команда
...
команда
!

Повторувај до *Услов*:

!
команда
команда
...
команда
!

Повторувај до Сид:

!
команда
команда
...
Команда
!

Повторувај до НАСОКА:

!
команда
команда
...
команда
!

ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROVOL+

○ Повик на процедура

***име** (параметар1, параметар2,)*

- Ако се повика процедура а која не постои или со грешни параметри – семантичка грешка
- Параметрите можат да бидат :
 - податок од тип **број**
 - податок од тип **Насока** т.е. една од константите: **И,З,С** и **Ј**
 - имиња на променливи: **име**



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROVOL+

- Декларација на променлива

Променлива *име, име, ..., име*

- Декларирањето на променлива мора да биде во главната процедура или во подпроцедурите, но не во некоја команда за повторување или услов
- Со ова се декларираат повеќе променливи
- После клучниот збор **Променлива** мора да има барем едно име



ДЕФИНИРАЊЕ НА ЈАЗИКOT ROBO L+

- Наредби за ажурирање на променливи

име = БРОЈ

име = БРОЈ + БРОЈ

име = БРОЈ – БРОЈ

*име = БРОЈ * БРОЈ*

- Ако се користи променлива, а не е иницијализирана – семантичка грешка
- Ако име не е променлива од тип **Број** – семантичка грешка



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROBOC+

- Што е Услов?

(БРОЈ = БРОЈ)

(БРОЈ <> БРОЈ)

(БРОЈ > БРОЈ)

(БРОЈ < БРОЈ)

(БРОЈ >= БРОЈ)

(БРОЈ <= БРОЈ)

- Условни наредби кои се или вистинити или неистинити



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROVOL+

- Пример: Дали можете да објасните што треба да прави оваа програма?

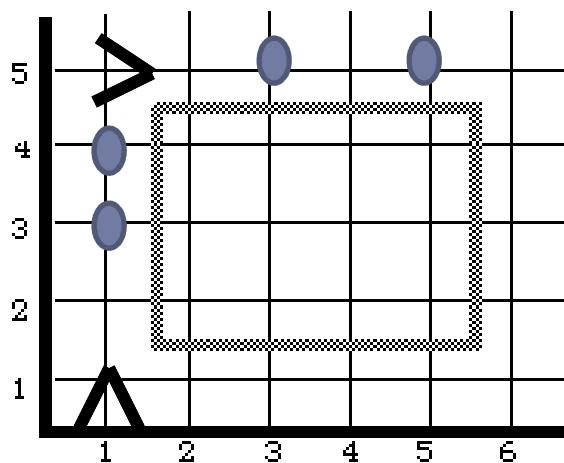
```
процедура ()  
  почеток  
  променлива к, бр  
  к=4  
  бр=0  
  ако ознака  
  !  
    бр=1  
  !  
  повторувај к пати:  
  !  
    оди  
    ако ознака  
    !  
      бр=бр+1  
    !  
  !
```

```
свртиДесно  
повторувај до (бр==0):  
!  
  оди  
  ако ознака  
  !  
    бр=бр-1  
  !  
!  
крај
```



ДЕФИНИРАЊЕ НА ЈАЗИКОТ ROVOL+

- Графички приказ на примерот



СЛЕДУВА

- Поделба по групи
- Ќе има 6 групи со најмногу 8 студенти
- Проектите се рангирени по тежина
 - Согласно со тоа за успешно сработен проект има различен број на максимум поени кои можете да ги освоите
- Пријавување
 - На анкетата поставена на курсот
 - Рок: 23.04.2010, петок 23:55

