Програмування інтелектуальних інформаційних систем

3 курс, осінь 2021

- Доц. Баклан І.В.
- Email: iaa@ukr.net
- Web: baklaniv.at.ua

Лекція 3

Мова штучного інтелекту LISP

СТУДЕНТ -> УЧАЩИЙСЯ
|
|-> ПІБ -> ТЕКСТ -> ПЕТРЕНКО ІВАН МИКОЛАЙОВИЧ
V
|-> ГРУППА -> ТЕКСТ -> ІП-91
V
|-> ФАКУЛЬТЕТ -> ТЕКСТ -> ІОТ
V
|-> ДНЮХА -> ДАТА -> 25 2 2000

Треба створити програмну структуру, за допомогою якої зберігаються подібні дані.

Для цього використовуємо LISP. Структуру подаємо у вигляді складного списку.

(СТУДЕНТ (УЧАЩИЙСЯ)

((ПІБ (ТЕКСТ (ПЕТРЕНКО ІВАН МИКОЛАЙОВИЧ)))

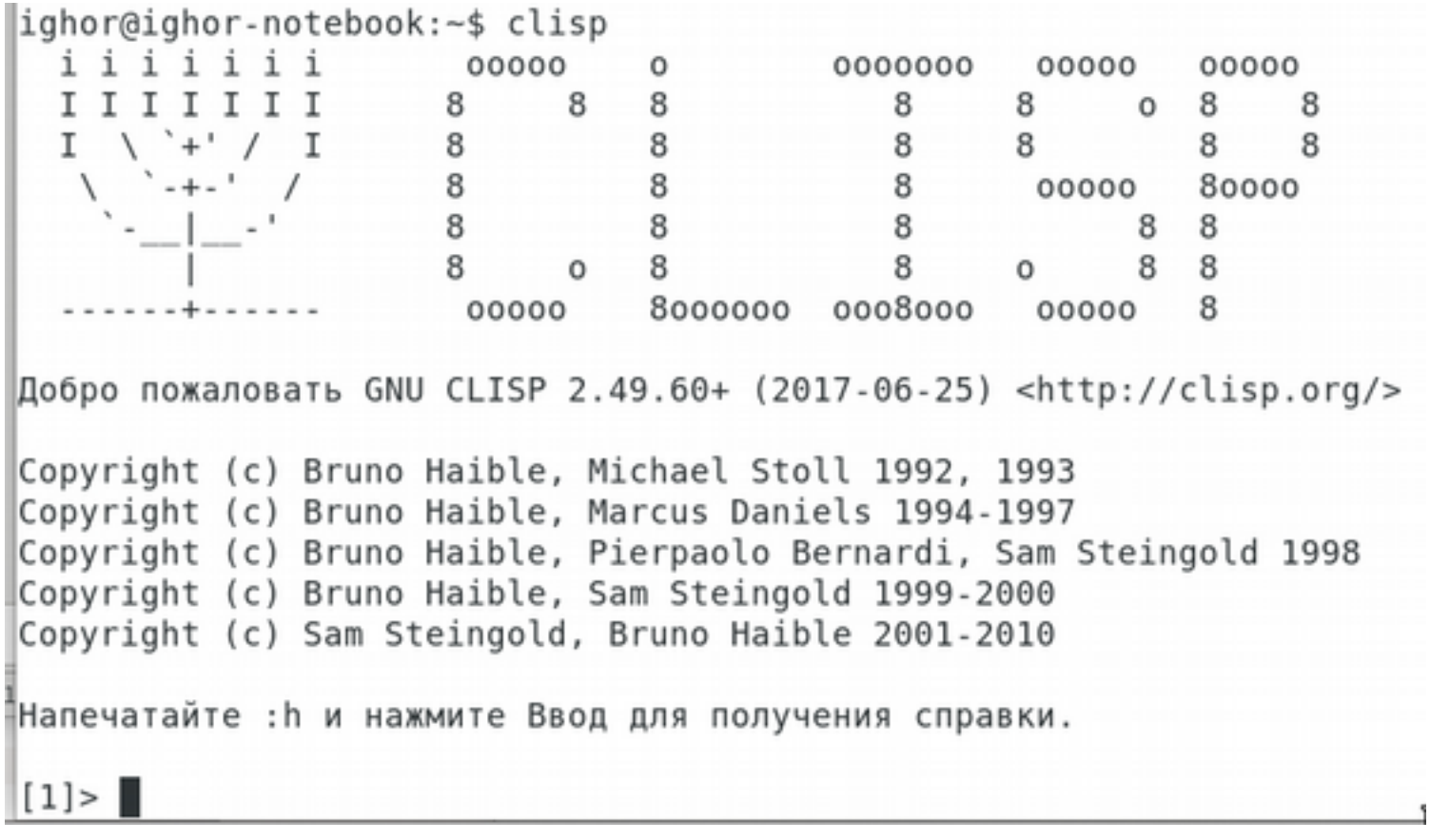
(ГРУППА (ТЕКСТ (ІП-91)))

(ФАКУЛЬТЕТ (ТЕКСТ (ІОТ)))

(ДНЮХА (ДАТА (25 2 2000))))))

Для зберігання списку в середовищі LISP використовуємо виклик наступної функції:

(**setq** cpisok `(СТУДЕНТ (УЧАЩИЙСЯ) ((ПІБ (ТЕКСТ (ПЕТРЕНКО ІВАН МИКОЛАЙОВИЧ))) (ГРУППА (ТЕКСТ (ІП-91))) (ФАКУЛЬТЕТ (ТЕКСТ (ІОТ))) (ДНЮХА (ДАТА (25 2 2000))))))



Спочатку запустимо середовище Common Lisp в терміналі ОС Ubuntu.

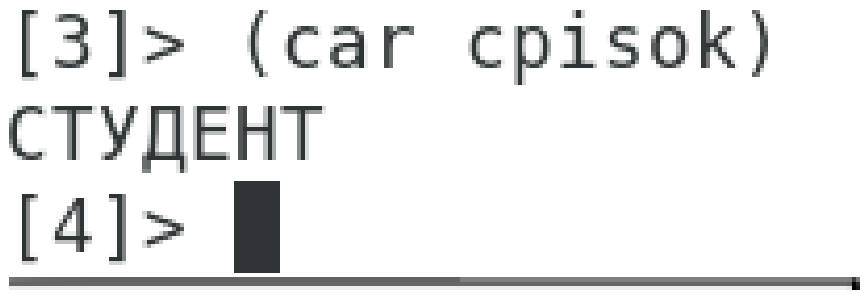
Після отримання запрошення > можемо ввести нашу команду.

```
[1]> (setq cpisok `(СТУДЕНТ (УЧАЩИЙСЯ)
 ((ПІБ (ТЕКСТ (ПЕТРЕНКО ІВАН МИКОЛАЙОВИЧ))) (ГРУППА (ТЕКСТ (ІП-91)))
  (ФАКУЛЬТЕТ (ТЕКСТ (ІОТ))) (ДНЮХА (ДАТА (25 2 2000))))))
(СТУДЕНТ (УЧАЩИЙСЯ)
 ((ПІБ (ТЕКСТ (ПЕТРЕНКО ІВАН МИКОЛАЙОВИЧ))) (ГРУППА (ТЕКСТ (ІП-91)))
  (ФАКУЛЬТЕТ (ТЕКСТ (ІОТ))) (ДНЮХА (ДАТА (25 2 2000)))))
[2]>
```

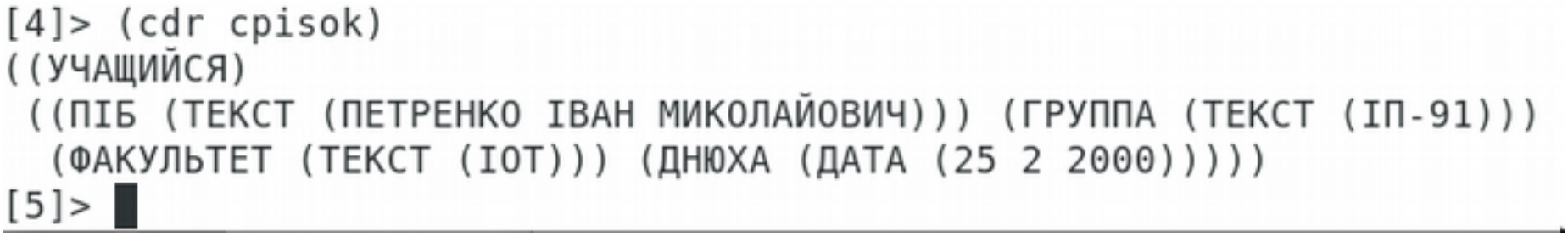
Після введення функції інтерпретатор виводить на екран значення функції. В нашому випадку список.

Функція встановлює зв’язок між іменем **`cpisok`** та нашим списком.

```
[2]> cpisok
(СТУДЕНТ (УЧАЩИЙСЯ)
 ((ПІБ (ТЕКСТ (ПЕТРЕНКО ІВАН МИКОЛАЙОВИЧ))) (ГРУППА (ТЕКСТ (ІП-91)))
  (ФАКУЛЬТЕТ (ТЕКСТ (ІОТ))) (ДНЮХА (ДАТА (25 2 2000)))))
[3]>
```

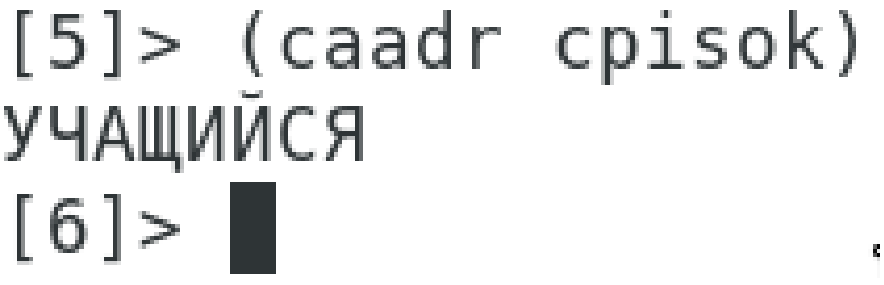


За допомогою функції car отримуємо “голову” списка.



Отримання хвоста списку.

Складна функція



Визначення користувацької функції.

```
[6]> (defun сінонім (x) (caadr x))
СІНОНІМ
[7]> (сінонім cpisok)
УЧАЩИЙСЯ
[8]>
```

Продовжимо розбір нашого списку на окремі частини.

```
[8]> (caaddr cpisok)
(ПІБ (ТЕКСТ (ПЕТРЕНКО ІВАН МИКОЛАЙОВИЧ)))
[9]> (cdaddr cpisok)
((ГРУППА (ТЕКСТ (ІП-91))) (ФАКУЛЬТЕТ (ТЕКСТ (ІОТ))) (ДНЮХА (ДАТА (25 2 2000))))
[10]> (car (cdaddr cpisok))
(ГРУППА (ТЕКСТ (ІП-91)))
[11]> (cadadr (car (cdaddr cpisok)))
(ІП-91)
[12]>
```

```
(caaddr cpisok)

(cdaddr cpisok)

(car (cdaddr cpisok))

(cadadr (car (cdaddr cpisok)))
```

Визначим функцію, значенням якої буде назва групи з нашого списку:

```
[12]> (defun какая-группа (x)(car (cadadr (car (cdaddr x)))))
КАКАЯ-ГРУППА
[13]> (какая-группа cpisok)
ІП-91
[14]>
```

```
(defun какая-группа (x)(car (cadadr (car
(cdaddr x)))))

(какая-группа cpisok)
```

Додамо в середовище ще один список аналогічної структури.

```
[14]> (setq cpisok2 `(СТУДЕНТ (УЧАЩИЙСЯ)
 ((ПІБ (ТЕКСТ (ДЖУРА ОКСАНА ПЕТРІВНА))) (ГРУППА (ТЕКСТ (ІП-91)))
  (ФАКУЛЬТЕТ (ТЕКСТ (ІОТ))) (ДНЮХА (ДАТА (31 6 2000))))))
(СТУДЕНТ (УЧАЩИЙСЯ)
 ((ПІБ (ТЕКСТ (ДЖУРА ОКСАНА ПЕТРІВНА))) (ГРУППА (ТЕКСТ (ІП-91)))
  (ФАКУЛЬТЕТ (ТЕКСТ (ІОТ))) (ДНЮХА (ДАТА (31 6 2000)))))
[15]>
```

```
(setq cpisok2 `(СТУДЕНТ (УЧАЩИЙСЯ)  ((ПІБ
(ТЕКСТ (ДЖУРА ОКСАНА ПЕТРІВНА))) (ГРУППА
(ТЕКСТ (ІП-91)))   (ФАКУЛЬТЕТ (ТЕКСТ (ІОТ)))
(ДНЮХА (ДАТА (31 6 2000))))))
```

Перевіримо як працюють наші користувацькі функції на новому списку.

```
[15]> (сінонім cpisok2)
УЧАЩИЙСЯ
[16]> (какая-группа cpisok2)
ІП-91
[17]>
```

(сінонім cpisok2)

(какая-группа cpisok2)

```
[3]> (defun ПІБ (x) (cadar (cdr (caaddr x))))
ПІБ
[4]> (піб cpisok)
(ПЕТРЕНКО ІВАН МИКОЛАЙОВИЧ)
[5]> (піб cpisok2)
(ДЖУРА ОКСАНА ПЕТРІВНА)
```

Визначимо функцію, яка видає список прізвища, ім’я та по-батькові з наших списків.

```
(defun ПІБ (x) (cadar (cdr (caaddr x))))

(піб cpisok)

(піб cpisok2)
```

Визначимо функцію яка видає лише прізвище з нашого списку.

```
[8]> (defun прізвище (x) (car (піб x)))
ПРІЗВИЩЕ
[9]> (прізвище cpisok)
ПЕТРЕНКО
[10]> (прізвище cpisok2)
ДЖУРА
[11]>
```

```
(defun прізвище (x) (car (піб x)))

(прізвище cpisok)

(прізвище cpisok2)
```

Визначимо функцію яка видає лише ім’я з нашого списку.

```
[11]> (defun імя (x)(cadr (піб x)))
ІМЯ
[12]> (імя cpisok)
IBAH
[13]> (імя cpisok2)
ОКСАНА
[14]>
```

```
(defun імя (x)(cadr (піб x)))

(імя cpisok)

(імя cpisok2)
```

Визначимо функція, яка видає список, що складається з прізвища та імені.

```
[14]> (defun хто (x) (list (прізвище x)(імя x)))
ХТО
[15]> (хто cpisok)
(ПЕТРЕНКО ІВАН)
[16]> (хто cpisok2)
(ДЖУРА ОКСАНА)
[17]>
```

Таким чином за короткий проміжок часу ми створили простеньку мову для роботи з базою знань.

Завдання: 1. Додати до спискової структури ще підструктури “Телефон” та “Адреса”.

2. Створити список зі своїми власними даними.

3. Написати функції “який-телефон” та “яка-адреса”, які видають відповідні значення з введених списків.

Наступна лекція буде присвячена мові представлення знань у вигляді фреймів — FRL.