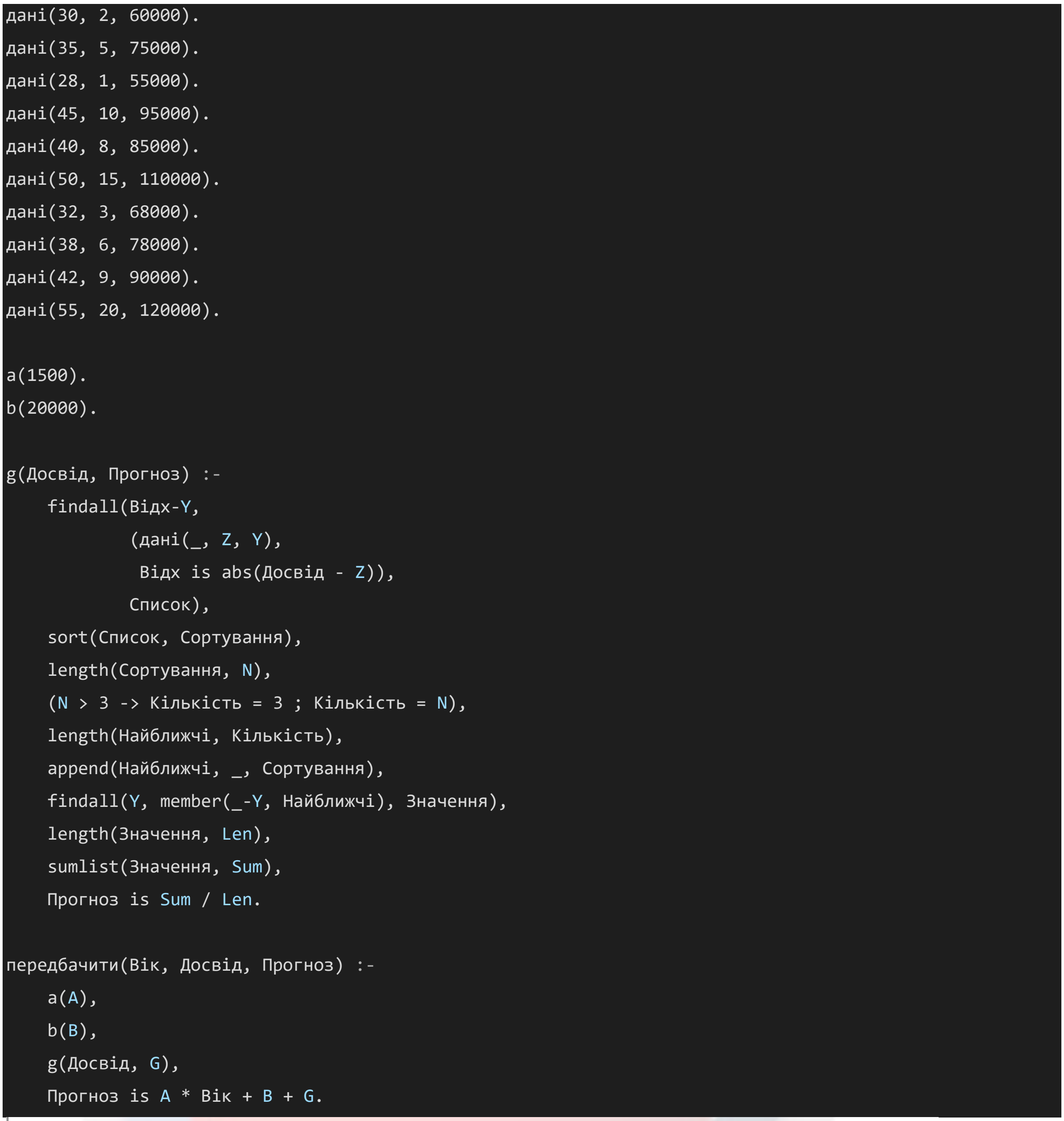
Костя Ткаченко ІП-з31

Варіант 16

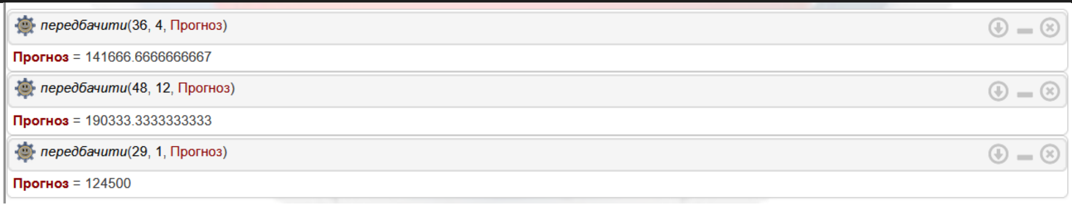
Розділ аналізу даних - Нестандартні передбачувачі

Алгоритм для реалізації - Напівпараметрична

Prolog

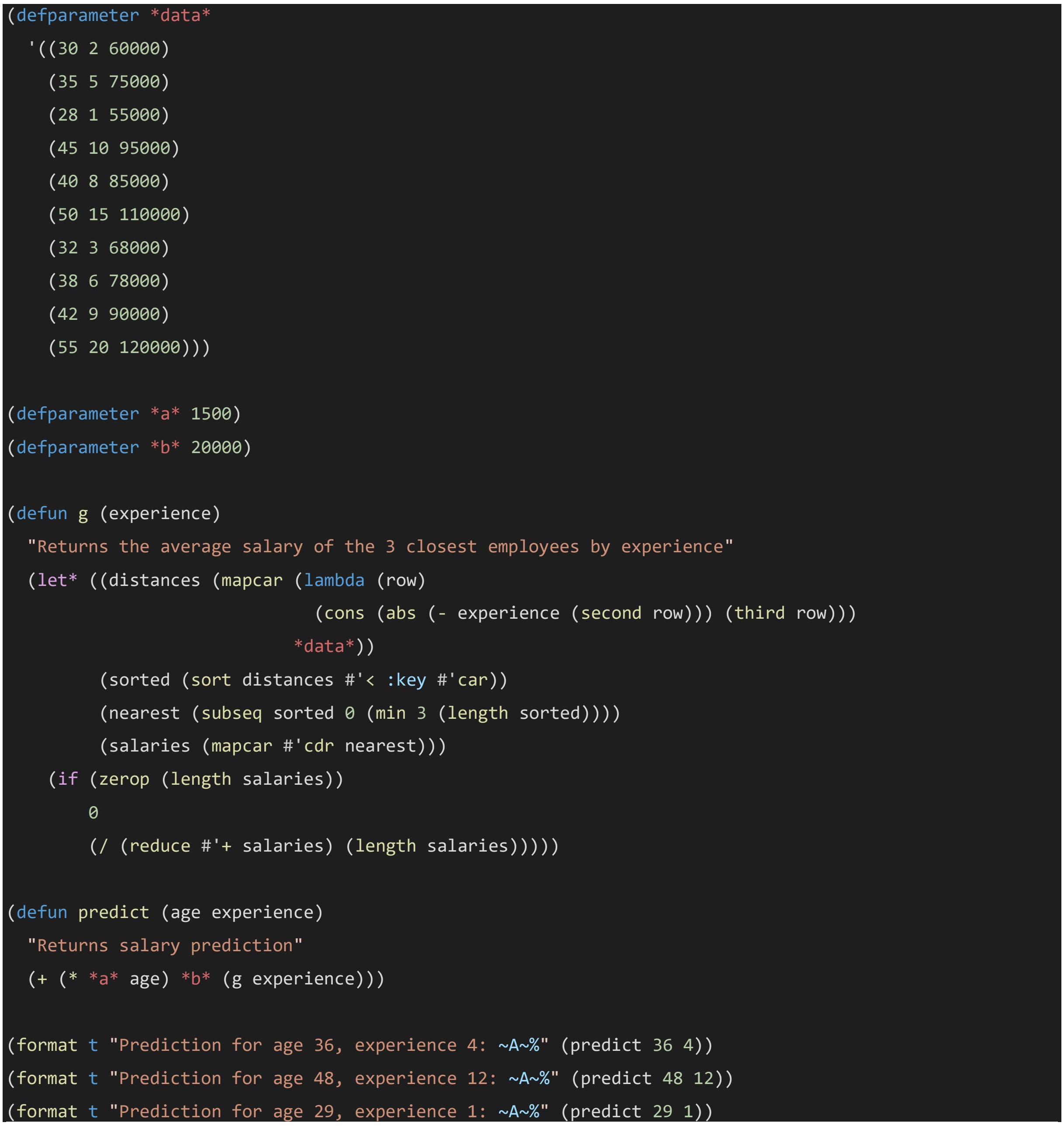


```
дані(30, 2, 60000).
дані(35, 5, 75000).
дані(28, 1, 55000).
дані(45, 10, 95000).
дані(40, 8, 85000).
дані(50, 15, 110000).
дані(32, 3, 68000).
дані(38, 6, 78000).
дані(42, 9, 90000).
дані(55, 20, 120000).

a(1500).
b(20000).

g(Досвід, Прогноз) :-
    findall(Відх-Y,
            (дані(_, Z, Y),
             Відх is abs(Досвід - Z)),
            Список),
    sort(Список, Сортування),
    length(Сортування, N),
    (N > 3 -> Кількість = 3 ; Кількість = N),
    length(Найближчі, Кількість),
    append(Найближчі, _, Сортування),
    findall(Y, member(_-Y, Найближчі), Значення),
    length(Значення, Len),
    sumlist(Значення, Sum),
    Прогноз is Sum / Len.

передбачити(Вік, Досвід, Прогноз) :-
    a(A),
    b(B),
    g(Досвід, G),
    Прогноз is A * Вік + B + G.
```

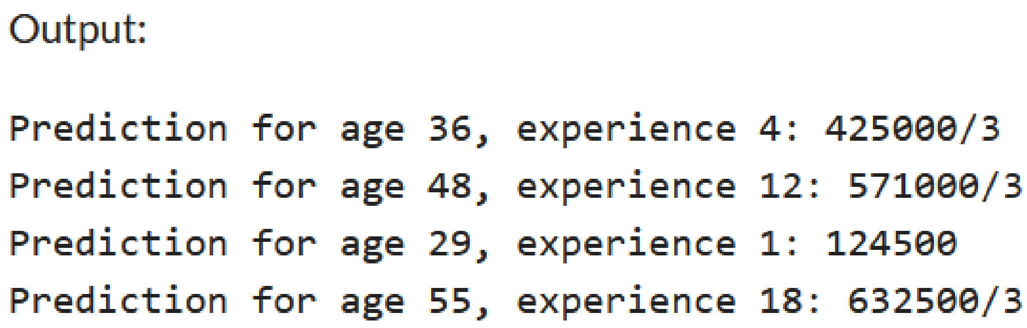


```
передбачити(36, 4, Прогноз)
Прогноз = 141666.6666666667
передбачити(48, 12, Прогноз)
Прогноз = 190333.3333333333
передбачити(29, 1, Прогноз)
Прогноз = 124500
```

Lisp



```lisp
(defparameter *data*
  '((30 2 60000)
    (35 5 75000)
    (28 1 55000)
    (45 10 95000)
    (40 8 85000)
    (50 15 110000)
    (32 3 68000)
    (38 6 78000)
    (42 9 90000)
    (55 20 120000)))

(defparameter *a* 1500)
(defparameter *b* 20000)

(defun g (experience)
  "Returns the average salary of the 3 closest employees by experience"
  (let* ((distances (mapcar (lambda (row)
                              (cons (abs (- experience (second row))) (third row)))
                            *data*))
         (sorted (sort distances #'< :key #'car))
         (nearest (subseq sorted 0 (min 3 (length sorted))))
         (salaries (mapcar #'cdr nearest)))
    (if (zerop (length salaries))
        0
        (/ (reduce #'+ salaries) (length salaries)))))

(defun predict (age experience)
  "Returns salary prediction"
  (+ (* *a* age) *b* (g experience)))

(format t "Prediction for age 36, experience 4: ~A~%" (predict 36 4))
(format t "Prediction for age 48, experience 12: ~A~%" (predict 48 12))
(format t "Prediction for age 29, experience 1: ~A~%" (predict 29 1))
```

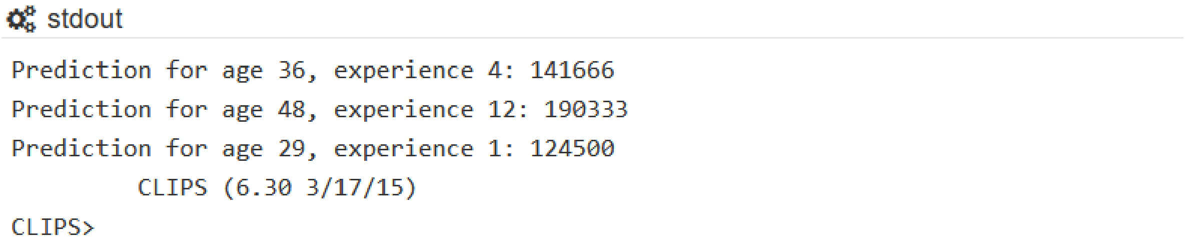


Output:

```
Prediction for age 36, experience 4: 425000/3
Prediction for age 48, experience 12: 571000/3
Prediction for age 29, experience 1: 124500
Prediction for age 55, experience 18: 632500/3
```

Clips

```
(deftemplate data
  (slot age (type INTEGER))
  (slot exp (type INTEGER))
  (slot salary (type INTEGER))
)

(defglobal ?*a* = 1500
        ?*b* = 20000)

(deffacts initial-data
  (data (age 30) (exp 2) (salary 60000))
  (data (age 35) (exp 5) (salary 75000))
  (data (age 28) (exp 1) (salary 55000))
  (data (age 45) (exp 10) (salary 95000))
  (data (age 40) (exp 8) (salary 85000))
  (data (age 50) (exp 15) (salary 110000))
  (data (age 32) (exp 3) (salary 68000))
  (data (age 38) (exp 6) (salary 78000))
  (data (age 42) (exp 9) (salary 90000))
  (data (age 55) (exp 20) (salary 120000))
)

(deffunction g (?exp_input)
  (bind ?deviations (create$))
  (bind ?salaries (create$))

  (do-for-all-facts ((?d data)) TRUE
    (bind ?dev (abs (- ?exp_input ?d:exp)))
    (bind ?deviations (create$ ?deviations ?dev))
    (bind ?salaries (create$ ?salaries ?d:salary))
  )

  (bind ?n (length$ ?deviations))
  (loop-for-count (?i 1 (- ?n 1)) do
    (loop-for-count (?j 1 (- ?n ?i)) do
      (bind ?dev1 (nth$ ?j ?deviations))
      (bind ?dev2 (nth$ (+ ?j 1) ?deviations))
      (if (> ?dev1 ?dev2) then

        (bind ?temp ?dev1)
        (bind ?deviations (replace$ ?deviations ?j ?j ?dev2))
        (bind ?deviations (replace$ ?deviations (+ ?j 1) (+ ?j 1) ?temp))

        (bind ?sal1 (nth$ ?j ?salaries))
        (bind ?sal2 (nth$ (+ ?j 1) ?salaries))
        (bind ?salaries (replace$ ?salaries ?j ?j ?sal2))
        (bind ?salaries (replace$ ?salaries (+ ?j 1) (+ ?j 1) ?sal1))
      )
    )
  )

  (bind ?k (min 3 ?n))
  (bind ?sum 0)
  (loop-for-count (?i 1 ?k) do
    (bind ?sum (+ ?sum (nth$ ?i ?salaries)))
  )

  (if (= ?k 0) then 0 else (/ ?sum ?k))
)

(deffunction predict (?age ?exp)
  (+ (* ?*a* ?age) ?*b* (g ?exp))
)

(reset)
(printout t "Prediction for age 36, experience 4: " (integer (predict 36 4)) crlf)
(printout t "Prediction for age 48, experience 12: " (integer (predict 48 12)) crlf)
(printout t "Prediction for age 29, experience 1: " (integer (predict 29 1)) crlf)
(printout t "Prediction for age 55, experience 18: " (integer (predict 55 18)) crlf)
```



```
Prediction for age 36, experience 4: 141666
Prediction for age 48, experience 12: 190333
Prediction for age 29, experience 1: 124500
         CLIPS (6.30 3/17/15)
CLIPS>
```