МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКО ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.И. ЛОБАЧЕВСКОГО»**

Дзержинский филиал

**Кафедра** Прикладная информатика

**ОТЧЕТ**

**по дисциплине «Проектный практикум»**

**к практической работе № 3**

**Тема «РАЗРАБОТКА ДЕМОНСТРАЦИОННОЙ ЭКСПЕРТНОЙ СИСТЕМЫ»**

Выполнил:

Студент гр. № 2721Б1ПИ

Погодин Дмитрий Евгеньевич

Проверил:

к.т.н. доцент

\_\_\_\_\_\_\_\_/Нажимова Н.А./ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дзержинск 2024

Оглавление

[**Практическая работа №3** 3](#_Toc147859884)

[**Вывод** 9](#_Toc147859885)

# **Практическая работа №3**

**«РАЗРАБОТКА ДЕМОНСТРАЦИОННОЙ ЭКСПЕРТНОЙ СИСТЕМЫ»**

1. Сформировать базу знаний демонстрационной ЭС и сохранить ее в файле rulebase.clp. ЭС должна вырабатывать рекомендации студенту накануне зачета и иметь следующие входные, промежуточные и выходные переменные:

*четыре входные переменные:*

– количество несданных работ (в %);

– число дней до зачета (целое число от 0 до 100);

– температура на улице (1-тепло, 0-холодно);

– наличие осадков (0-нет, 1-есть).

*две промежуточные:*

– свободное время (0-отсутствует, 1-мало, 2-много);

– погода (0-плохая, 1-хорошая).

*выходная переменная:*

– рекомендуемое действие (0-надо учить, 1-отдыхать).

Диаграмма зависимости переменных показана на рис. 1, в скобках указаны возможные имена переменных.

Перед началом составления правил на языке CLIPS необходимо составить таблицы правил, отражающие все возможные комбинации переменных с их словесными пояснениями.

Например, для переменной «Свободное время» таблица может выглядеть как табл. 2.

Таблица 1

|  |  |  |
| --- | --- | --- |
| Temper | Rain | Weather |
| -30<Холодно(0)<=+10  -30<Холодно(0)<=+10  +11<Тепло (1)<=+30  +11<Тепло (1)<=+30 | 0<Есть (1)=1  -1<Нет (0)=0  -1<Нет (0)=0  0<Есть (1)=1 | Плохая (0)  Плохая (0)  Хорошая (1)  Хорошая (1) |

Таблица 2

|  |  |  |
| --- | --- | --- |
| Days | Works | Free time |
| любое  0<days<=7  0<days<=7  0<days<=7  7<days<∞  7<days<∞  7<days<∞ | 0%  0%<works<=30%  30%<works<=60%  60%<works<=100%  0%<works<=30%  30%<works<=60%  60%<works<=100% | Много (2)  Мало (1)  Мало (1)  Отсутствует (0)  Много (2)  Много (2)  Мало (1) |

Таблица 3

|  |  |  |
| --- | --- | --- |
| Free time | Weather | Act |
| 1<Много (2)<=2  0<Мало (1)<=1  -1<Отсутствует (0)<1  -1<Отсутствует (0)<1  1<Много (2)<=2  0<Мало (1)<=1 | -1<Плохая (0)<1  -1<Плохая (0)<1  0<Хорошая (1)<=2  -1<Плохая (0)<1  0<Хорошая (1)<=2  -1<Плохая (0)<1 | Надо учить (0)  Отдыхать (1)  Отдыхать (1)  Отдыхать (1)  Надо учить(0)  Отдыхать (1) |

Алгоритм практической работы

2. Для ввода исходных значений переменных необходимо в начале программы добавить правило, использующее команду связывания bind для считывания значений, например:

(defrule data-input

(initial-fact)

=>

(printout t crlf "Сhislo dnei do zacheta: ")

(bind ?days (read))

(assert (days ?days))

(printout t crlf "Сhislo nesdelannyh rabot (%) ")

(bind ?works (read))

(assert (works ?works))

(printout t crlf "Na ulitse teplo? (da - 1/net - 0): ")

(bind ?temper (read))

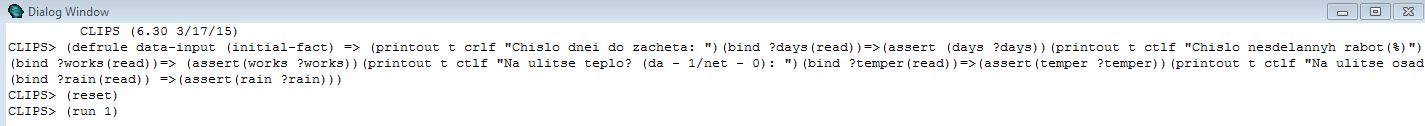
(assert (temper ?temper))

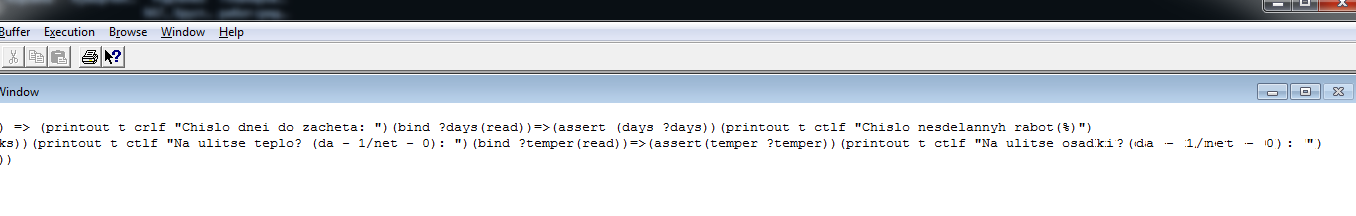
(printout t crlf "Na ulitse osadki? (da - 1/net - 0): ")

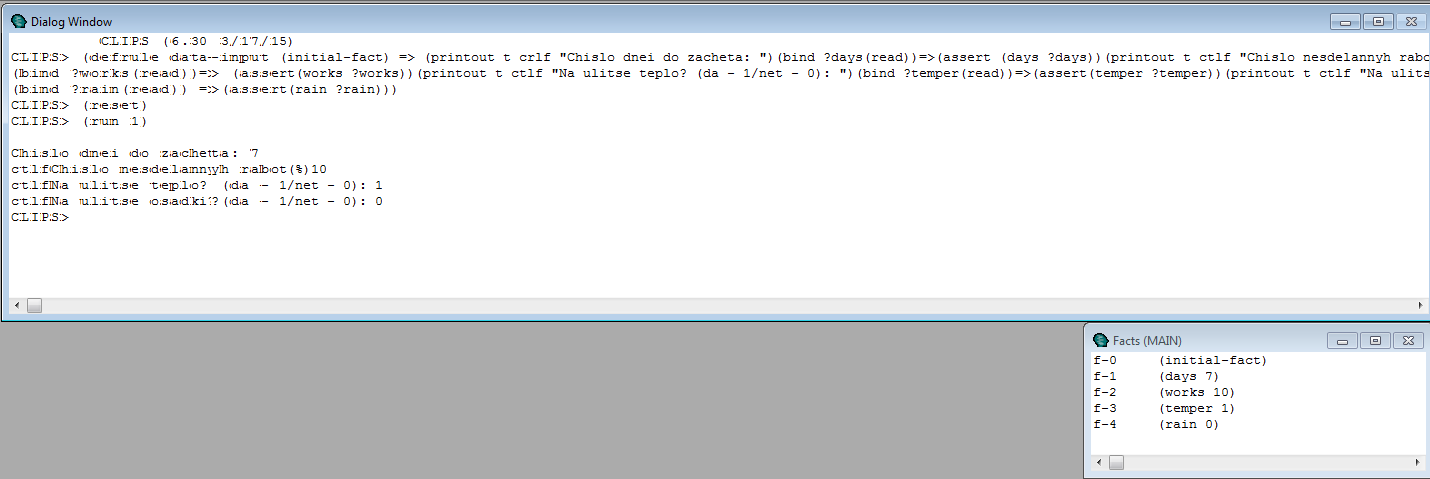
(bind ?rain (read))

(assert (rain ?rain))

)







3. В соответствии с составленными таблицами правил записываем правила на языке CLIPS. Пример правила для переменной «Свободное время» (табл.1, строка 2) записан ниже:

(defrule pravilo2

(works ?works)

(days ?days)

(test (and (and (>= ?days 0) (< ?days 7)) (and (> ?works 0) (< ?works 30))))

=>

(printout t crlf crlf "Vremeni malo" crlf)

(assert (freetime 1))

Здесь:

defrule pravilo2 *–* имя правила;

works ?works – считываем значения из факта works в переменную ?works;

test (and (and (>= ?days 0) (< ?days 7)) (and (> ?works 0)  
(< ?works 30))) –задаем основной антецедент правила командой (test) (проверить);

команда (and) вычисляет логическое «И» для двух значений,

команды (>,<,<=,>=,=,<>) проверяют выполнение соответствующего неравенства для двух значений (например: запись на языке CLIPS   
(>= ?days 0) тождественна математической записи ?days>=0);

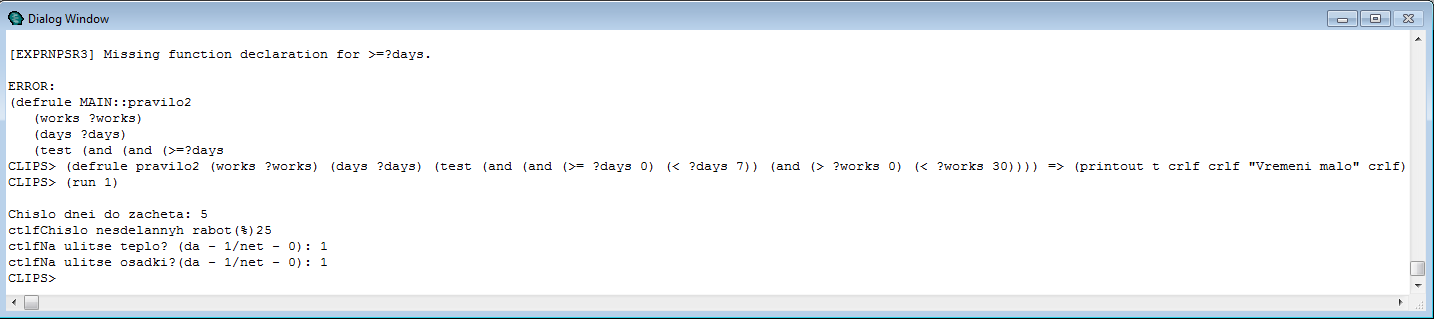
printout t crlf crlf "Vremeni malo" crlf – вывод на экран;

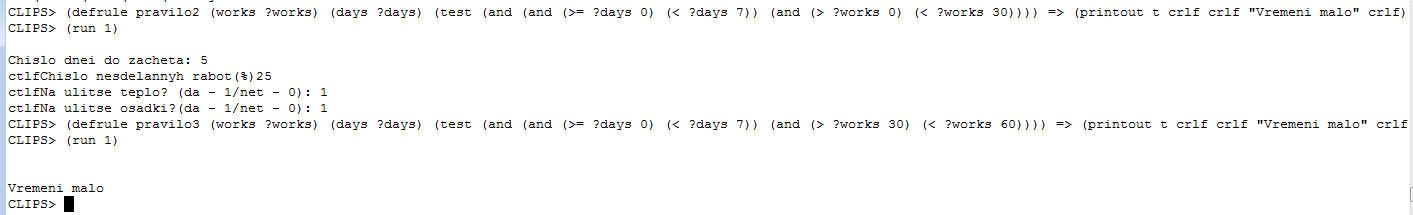
assert (freetime 1) – добавление нового факта.

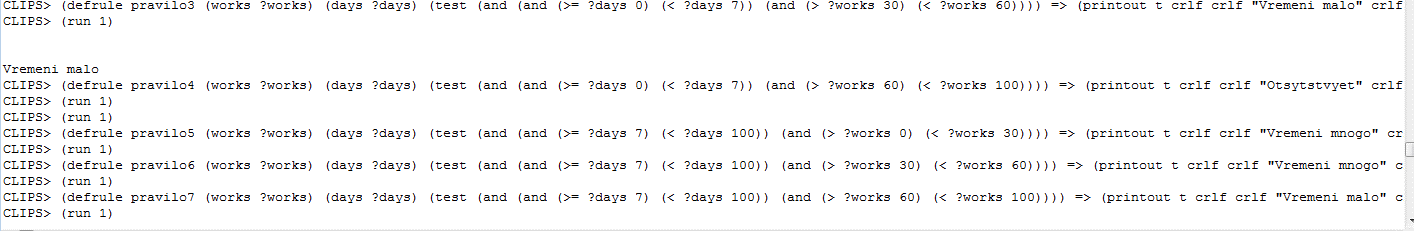
Также база знаний должна содержать в себе набор команд добавления новых правил.

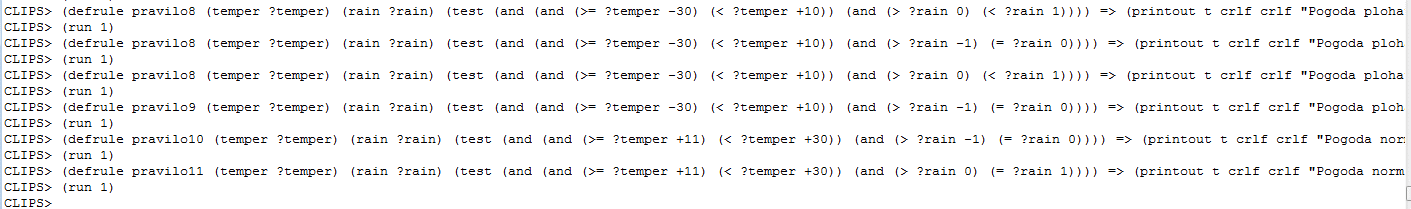
* (defrule data-input (initial-fact) => (printout t crlf "Chislo dnei do zacheta: ")(bind ?days(read))=>(assert (days ?days))(printout t ctlf "Chislo nesdelannyh rabot(%)")
* (bind ?works(read))=> (assert(works ?works))(printout t ctlf "Na ulitse teplo? (da - 1/net - 0): ")(bind ?temper(read))=>(assert(temper ?temper))(printout t ctlf "Na ulitse osadki?(da - 1/net - 0): ")
* (bind ?rain(read)) =>(assert(rain ?rain)))
* (defrule pravilo2 (works ?works) (days ?days) (test (and (and (>= ?days 0) (< ?days 7)) (and (> ?works 0) (<= ?works 30)))) => (printout t crlf crlf "Vremeni malo" crlf) (assert (freetime 1))
* (defrule pravilo3 (works ?works) (days ?days) (test (and (and (>= ?days 0) (< ?days 7)) (and (> ?works 30) (<= ?works 60)))) => (printout t crlf crlf "Vremeni malo" crlf) (assert (freetime 1)))
* (defrule pravilo4 (works ?works) (days ?days) (test (and (and (>= ?days 0) (< ?days 7)) (and (> ?works 60) (<= ?works 100)))) => (printout t crlf crlf "Otsytstvyet" crlf) (assert (freetime 0)))
* (defrule pravilo5 (works ?works) (days ?days) (test (and (and (>= ?days 7) (< ?days 100)) (and (> ?works 0) (<= ?works 30)))) => (printout t crlf crlf "Vremeni mnogo" crlf) (assert (freetime 2)))
* (defrule pravilo6 (works ?works) (days ?days) (test (and (and (>= ?days 7) (< ?days 100)) (and (> ?works 30) (<= ?works 60)))) => (printout t crlf crlf "Vremeni mnogo" crlf) (assert (freetime 2)))
* (defrule pravilo7 (works ?works) (days ?days) (test (and (and (>= ?days 7) (< ?days 100)) (and (> ?works 60) (<= ?works 100)))) => (printout t crlf crlf "Vremeni malo" crlf) (assert (freetime 1)))
* (defrule pravilo8 (temper ?temper) (rain ?rain) (test (and (and (>= ?temper -30) (< ?temper +10)) (and (> ?rain 0) (= ?rain 1)))) => (printout t crlf crlf "Pogoda plohaya" crlf) (assert (weather 0)))
* (defrule pravilo9 (temper ?temper) (rain ?rain) (test (and (and (>= ?temper -30) (< ?temper +10)) (and (> ?rain -1) (= ?rain 0)))) => (printout t crlf crlf "Pogoda plohaya" crlf) (assert (weather 0)))
* (defrule pravilo10 (temper ?temper) (rain ?rain) (test (and (and (>= ?temper +11) (< ?temper +30)) (and (> ?rain -1) (= ?rain 0)))) => (printout t crlf crlf "Pogoda norm" crlf) (assert (weather 1)))
* (defrule pravilo11 (temper ?temper) (rain ?rain) (test (and (and (>= ?temper +11) (< ?temper +30)) (and (> ?rain 0) (= ?rain 1)))) => (printout t crlf crlf "Pogoda norm" crlf) (assert (weather 1)))
* (defrule pravilo12 (freetime ?freetime) (weather ?weather) (test (and (and (> ?freetime 1) (= ?freetime 2)) (and (> ?weather -1) (= ?weather 0)))) => (printout t crlf crlf "Nado ychit" crlf) (assert (act 0)))
* (defrule pravilo13 (freetime ?freetime) (weather ?weather) (test (and (and (> ?freetime 0) (= ?freetime 1)) (and (> ?weather -1) (= ?weather 0)))) => (printout t crlf crlf "Nado relax" crlf) (assert (act 1)))
* (defrule pravilo14 (freetime ?freetime) (weather ?weather) (test (and (and (> ?freetime -1) (= ?freetime 0)) (and (> ?weather 0) (= ?weather 1)))) => (printout t crlf crlf "Nado relax" crlf) (assert (act 1)))
* (defrule pravilo15 (freetime ?freetime) (weather ?weather) (test (and (and (> ?freetime -1) (= ?freetime 0)) (and (> ?weather -1) (= ?weather 0)))) => (printout t crlf crlf "Nado relax" crlf) (assert (act 1)))
* (defrule pravilo16 (freetime ?freetime) (weather ?weather) (test (and (and (> ?freetime 1) (= ?freetime 2)) (and (> ?weather 0) (= ?weather 1)))) => (printout t crlf crlf "Nado ychit" crlf) (assert (act 0)))

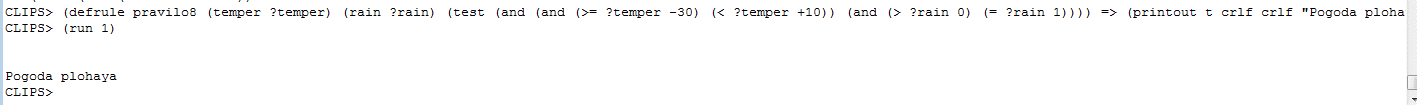
(defrule pravilo17 (freetime ?freetime) (weather ?weather) (test (and (and (> ?freetime 0) (= ?freetime 1)) (and (> ?weather -1) (= ?weather 0)))) => (printout t crlf crlf "Nado relax" crlf) (assert (act 1)))

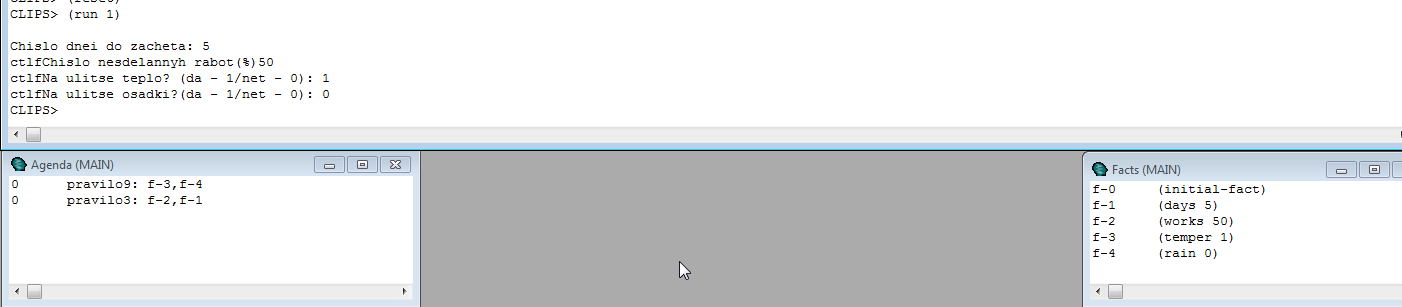


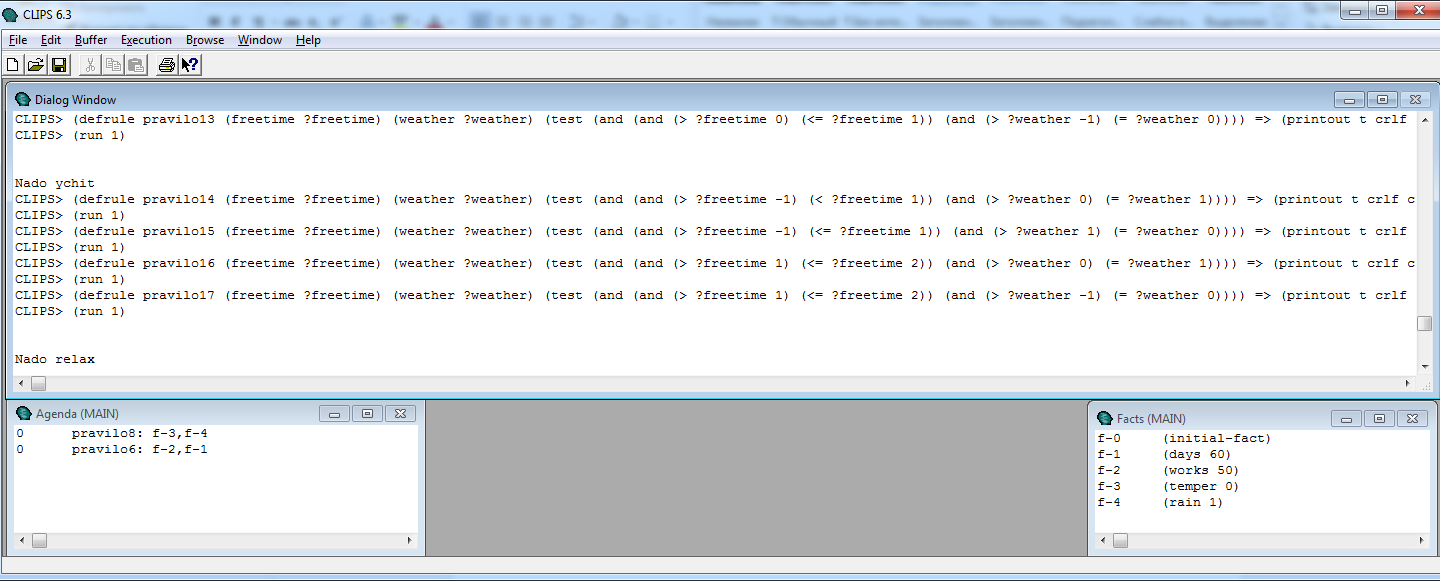


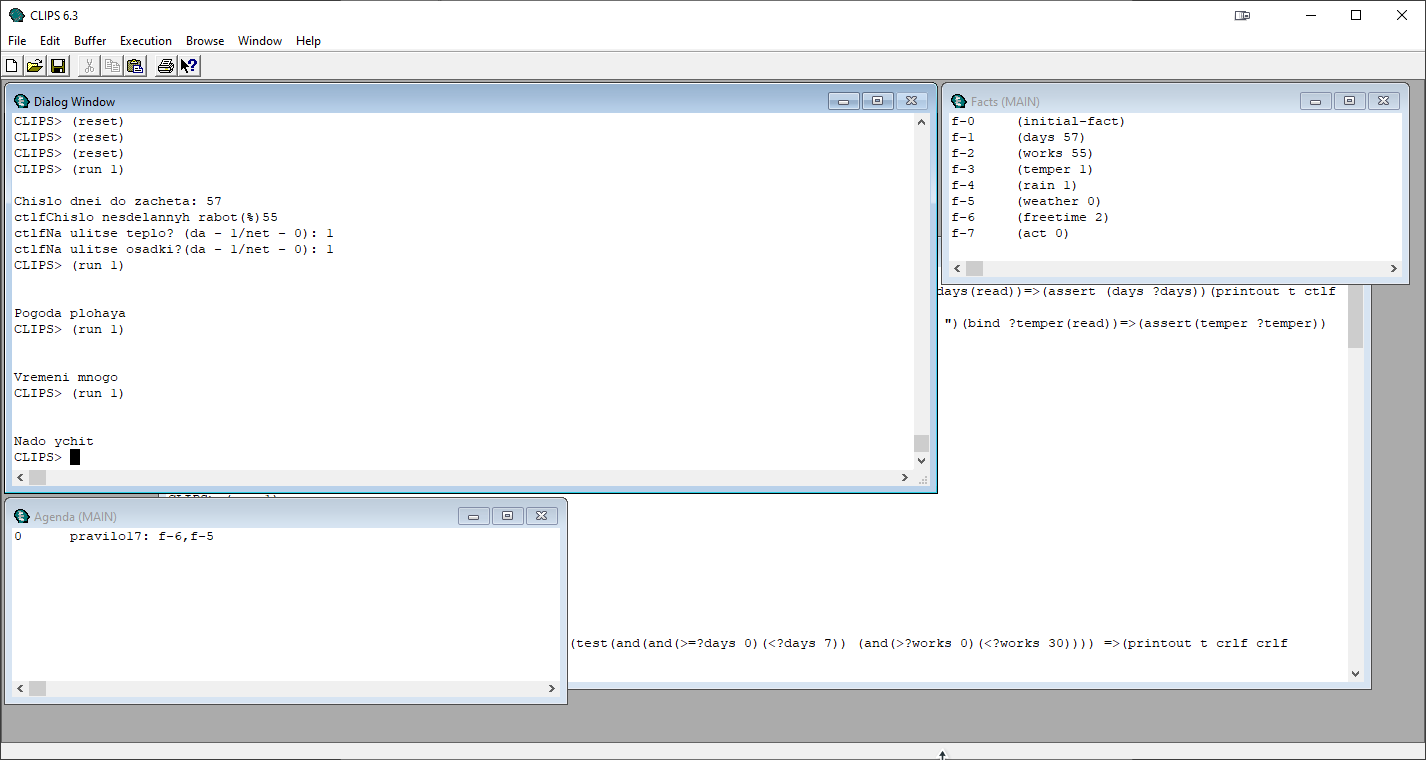


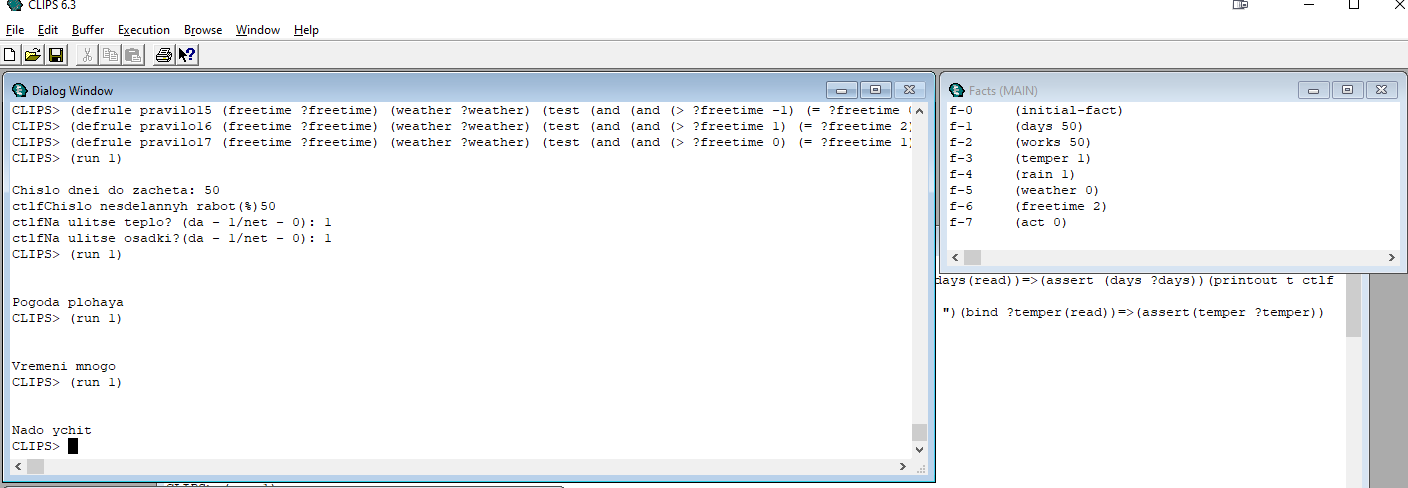
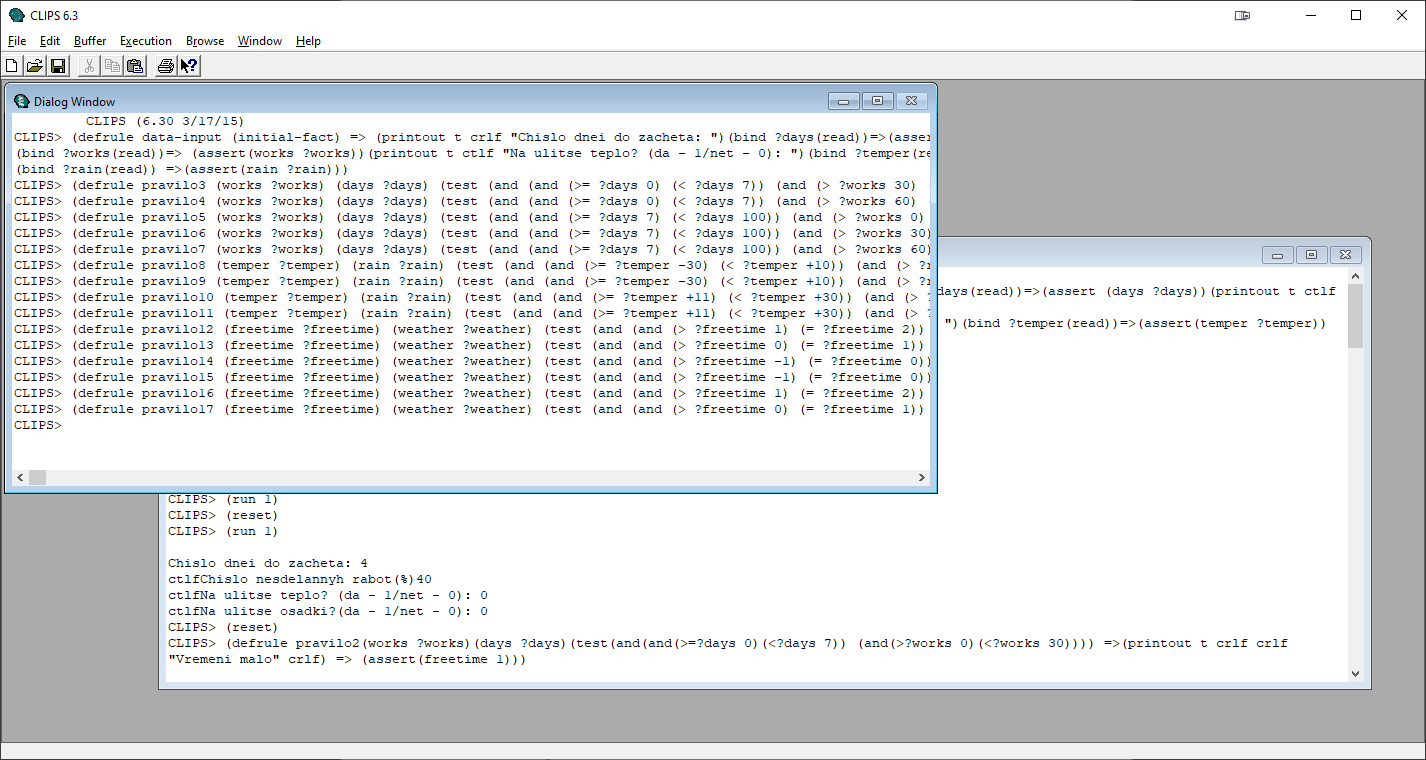


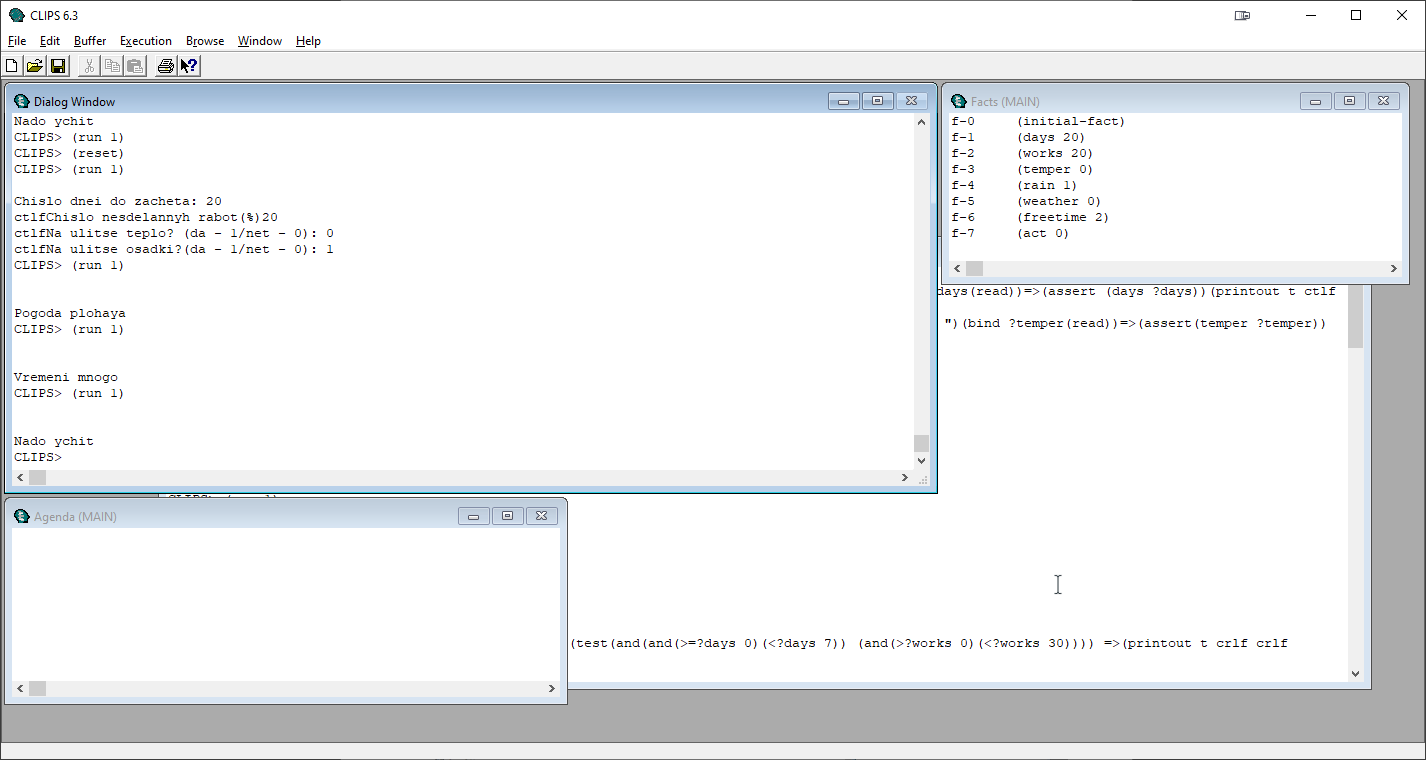












# **Вывод**

В ходе выполнения практической работы №3 я выполнил поставленные задачи в CLIPS, а именно научился разрабатывать демонстрационную экспертную систему.