МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

Кафедра ИИ

Дисциплина: Инженерия знаний

Отчет по лабораторной работе №5

Тема: «Создание прототипа экспертной системы»

Выполнили: Проверил:

Ст. гр.КН-11-3 Золотухин О.В.

Бурлай А.

Миролюбова О.

Неблиенко М.

Харьков 2014

**Цель работы**:

Создать прототип экспертной системы при помощи языка CLIPS.

**Программа**:

(deffunction ask-question (?question $?allowed-values)

(printout t ?question)

(bind ?answer (read))

(while (not (member ?answer ?allowed-values)) do

(printout t ?question)

(bind ?answer (read))

)

?answer)

(deffunction yes-or-no (?question)

(bind ?response (ask-question ?question yes no))

(if (eq ?response yes)

then TRUE

else FALSE))

(defrule is-pc-working

(not (pc-working ?))

(not (advice ?))

=>

(if (yes-or-no "Is your PC working? (yes/no) ")

then

(assert (pc-working yes))

(assert (advice "You do not need any advices."))

else

(assert (pc-working no))))

(defrule is-pc-powered

(pc-working no)

(not (pc-powered ?))

(not (advice ?))

=>

(if (yes-or-no "Is your PC powered on? (yes/no) ")

then (assert (pc-powered yes))

else (assert (pc-powered no))))

(defrule is-light

(pc-powered no)

(not (light ?))

(not (advice ?))

=>

(if (yes-or-no "Do you have the light? (yes/no) ")

then

(assert (light yes))

(assert (advice "The power supply is broken."))

else

(assert (light no))

(assert (advice "Turn the light on."))

)

)

(defrule picture-on-monitor

(pc-powered yes)

(not (picture-on-monitor ?))

(not (advice ?))

=>

(if (yes-or-no "Is there any picture on the monitor? (yes/no) ")

then

(assert (picture-on-monitor yes))

else

(assert (picture-on-monitor no))))

(defrule monitor-connected-to-network

(picture-on-monitor no)

(not (monitor-connected ?))

(not (advice ?))

=>

(if (yes-or-no "Is your monitor connected to network? (yes/no) ")

then

(assert (monitor-connected yes))

else

(assert (monitor-connected no))

(assert (advice "Connect monitor to the network."))

))

(defrule lamp-lights-up

(monitor-connected yes)

(not (lamp-lights-up ?))

(not (advice ?))

=>

(if (yes-or-no "Does lamp on monitor light up? (yes/no) ")

then

(assert (lamp-lights-up yes))

(assert (advice "Graphics card is out of service."))

else

(assert (lamp-lights-up no))

(assert (advice "Monitor is out of service."))

))

(defrule is-squeak-on-load

(picture-on-monitor yes)

(not (squeak-on-load ?))

(not (advice ?))

=>

(if (yes-or-no "Is there any squeak while loading? (yes/no) ")

then

(assert (squeak-on-load yes))

(assert (advice "Check BIOS list of signals."))

else

(assert (squeak-on-load no))))

(defrule is-os-loading

(squeak-on-load no)

(not (os-loading ?))

(not (advice ?))

=>

(if (yes-or-no "Is OS loading? (yes/no) ")

then

(assert (os-loading yes))

else

(assert (os-loading no))))

(defrule is-error-message

(os-loading no)

(not (error-message ?))

(not (advice ?))

=>

(if (yes-or-no "Is there any error message? (yes/no) ")

then

(assert (error-message yes))

(assert (advice "Follow OS advices."))

else

(assert (error-message no))

(assert (advice "Sorry, we can't help you."))

))

(defrule is-sound-entering-system

(os-loading yes)

(not (sound-entering-system ?))

(not (advice ?))

=>

(if (yes-or-no "Is there any sound while entering the system? (yes/no) ")

then

(assert (sound-entering-system yes))

else

(assert (sound-entering-system no))))

(defrule is-sound-card

(sound-entering-system yes)

(not (sound-card ?))

(not (advice ?))

=>

(if (yes-or-no "Do you have sound card? (yes/no) ")

then

(assert (sound-card yes))

(assert (advice "Sound card is out of service."))

else

(assert (sound-card no))))

(defrule sytem-loaded-no-errors

(or (sound-entering-system no) (sound-card no))

(not (loaded-wo-errors ?))

(not (advice ?))

=>

(if (yes-or-no "Did system load without errors? (yes/no) ")

then

(assert (loaded-wo-errors yes))

(assert (advice "Have nice work!"))

else

(assert (loaded-wo-errorsno))

(assert (advice "Follow OS reccomendations."))

))

(defrule print-advice

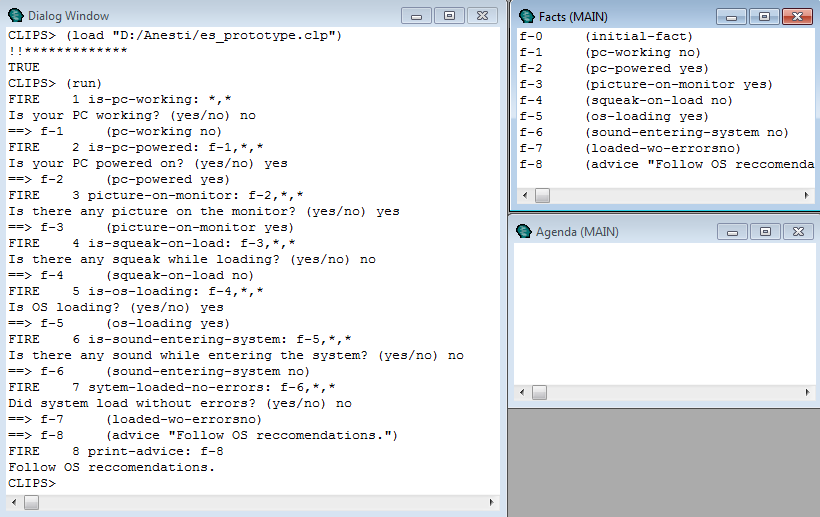
(advice ?advice)

=>

(printout t ?advice crlf)

)

**Результат выполнения**:

****

**Выводы**:

В данной лабораторной работе спроектировали при помощи языка CLIPS прототип экспертной системы, диагностирующей причины поломки компьютера.