

Федеральное агентство по образованию
Государственное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

В.В. Тимофеев

«Утверждаю»
Заведующий кафедрой ИУ6
_____ Сюзев В.В.
«___» _____ 2012г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ
ДЛЯ ДИСЦИПЛИНЫ ПО ВЫБОРУ.**

Москва
МГТУ им. Н.Э. Баумана
2012

Содержание

Цель работ. Основные функции языка Лисп. Структура команды пользователя. Формирование интерфейса команды Графические возможности Автолиспа. Обработка ошибок пользователя. Формирование меню команд пользователя. Практическая часть. Задание. Пример. Задание для домашней подготовки.

Цель работ: Приобретение навыков применения языка Автолисп для разработки команд пользователя и настройки среды, разработка пользовательских меню в рамках ACADa.

Краткие сведения о языке Автолисп

Автолисп представляет собой модификацию языка программирования Лисп, позволяющую пользователю Автокада писать функции и команды, преобразуя универсальную графическую систему в личный инструмент компьютерной графики.

Типы данных в Автолиспе

Автолисп поддерживает следующие основные типы данных:

1. Целые.
2. Действительные.
3. Строковые.
4. Встроенные функции.
5. Дескрипторы файлов.
6. Имена примитивов Автокада.
7. Наборы Автокада.
8. Списки.

Переменные целого типа могут принимать значения положительных или отрицательных целых чисел. Целые представлены двумя байтами, диапазон представления (-32768 32767).

Переменные действительного типа могут принимать значения положительные или отрицательные с десятичной точкой. В Автолиспе не разрешено начинать или заканчивать число десятичной точкой. Правильной записью действительных чисел являются числа 0.12, 5.0, -36.54, 8.2679E-11.

Переменные строкового типа это совокупности символов, заключенные в кавычки.

Встроенные функции-переменные, посредством которых осуществляется обращение к стандартной библиотеке функций Автолиспа, в них хранятся адреса стандартных функций Автолиспа.

Дескрипторы файлов- переменные, в которых содержится описание открытого файла.

Имена примитивов Автокада – переменные, в которых содержится уникальный индекс(восьмизначное шестнадцатиричное число), однозначно соответствующий примитивам текущего чертежа в графической базе данных.

Списки – один из важнейших типов данных в Лиспе, это последовательность элементов, заключенная в скобки. Элемент списка может быть любого типа.

Основные функции Лиспа

Структура функции Лиспа представлена списком, первый элемент которого имя функции, последующие – ее аргументы.

(<имя функции> <аргумент1>[<аргумент2> <...>])

или

(<имя функции>[<аргумент> <...>])

Присваивание в Лиспе выполняется в основном функцией

(SETQ <имя 1> <выражение1> <имя2> <выражение2> ...)

Определение функций пользователя

Структура функции пользователя следующая:

```
( DEFUN <имя функции>([<список аргументов>][ / < список локальных параметров>])  
< последовательность выражений Лиспа >  
)
```

Пример1.

; Функция преобразования углов в градусы

```
(DEFUN GRAD_RAD(A)  
  (*PI (/ A 180))  
)
```

Для функций пользователя не следует выбирать имена встроенных функций, т.к. они станут недоступны.

Если с помощью текстового редактора, который не вносит свои символы в текст, или редактора VsialLisp ACADa 2000 сформировать файл *.LSP, сохранить его, а затем в ответ на подсказку Автокада COMMAND: загрузить этот файл функцией Лиспа (LOAD “*.LSP”) или с использованием меню ACAD раздела Tolls, Автокад загрузит эту функцию или группу функций из указанного файла, отобразив имя файла на экране и сделает доступными эти функции для пользователя.

Структура команды пользователя

Команда пользователя оформляется как функция Автолиспа.

Особенностью команды пользователя в описании является ее имя, а также отсутствие аргументов и локальных параметров.

Требования к команде пользователя:

- 1) Команда должна однозначно выполняться при любых настройках системы;
- 2) При необходимых изменениях настроек, команда должна их восстанавливать;
- 3) Диалог в пользовательской команде должен соответствовать диалогу системы.

Имя команды начинается с обязательных символов C:, эти символы позволяют вызывать такую функцию как стандартную команду Автокада (вызов осуществляется по имени без символов C:).

Команда пользователя по протоколу работы должна соответствовать стандартным командам системы, быть структурированной и содержать следующее:

1. Заголовок команды.
2. Ввод исходных данных (если он необходим).
3. Запоминание состояния среды системы, для последующего изменения и восстановления (если это не специальная команда изменения состояния среды).
4. Настройку среды системы для выполнения данной команды пользователя.
5. Выполнение необходимых вычислений.
6. Вычерчивание фрагмента чертежа (если это необходимо в данной команде).
7. Восстановление состояния среды системы (если данная команда не предназначена для изменения состояния среды).
8. Окончание команды пользователя.

Заголовок команды оформляется как заголовок функции Лиспа с именем включающем предшествующие символы C: , например:

```
(DEFUN C:DETAL1 ()
```

```
<выражения Лиспа>
```

```
)
```

Вызов такой команды с командной строки на COMMAND: будет выполняться по имени DETAL1.

Ввод исходных данных в диалоге оформляется функциями (GET...)

1. Ввод угла (GETANGLE [<точка >][<текст запроса команды>])

Здесь точка – базовая точка в текущей ПСК (точка привязки к полю чертежа). Возвращает угол в радианах , который задается

пользователем путем набора числа на клавиатуре в текущих для ACAD единицах измерения или путем указания двух точек, при этом от первой (или базовой) точки рисуется «резиновая нить» до курсора, позволяя установить требуемый угол.

Например (Setq ANG1(GetAngle))

Или (Setq ANG2 (GetAngle “\n угол:”))

2. Ввод расстояния

(GETDIST[< точка>][<текст запроса команды>])

Здесь точка – точка привязки к полю чертежа 2х или 3хмерная.

Например:

(Setq DIST1 (GetDist))

(Setq DIST2 (GetDist P1 “\n расстояние:”))

В последнем случае рисуется «резиновая нить» от точки P1 до курсора, позволяя установить требуемое расстояние.

2. Ввод точки

(GetPoint[< точка>][<текст запроса команды>])

Например:

(Setq Point1(GetPoint “\n Базовая точка”))

3. Ввод действительного числа

(GETREAL >[<текст запроса команды>])

Например:

(setg M1 (GetReal “\n Масштабный коэффициент”))

Ввод данных в команде пользователя должен быть лаконичным, удобным для пользователя (использовать угловые или линейные шкалы, диалоговые окна), непротиворечивым, с применением значений по умолчанию и проверкой ошибок пользователя.

- Пример ввода по умолчанию:

п-р:

Масштаб <10>:

```
; ввод масштаба
(setq M(GetReal "\n Масштаб <10>:"))
(if (= M nil) (setq M 10))
```

Запоминание состояния среды системы

Команда пользователя, если это не специальная команда изменения настроек среды, не должна изменять настройки и должна адекватно выполняться при любых настройках.

Для реализации этого необходимо выполнить ряд действий с системными переменными. К числу таких действий следует отнести:

1. Запоминание состояния среды (запоминание значений необходимых системных переменных).
2. Установка новых значений системных переменных.

После выполнения требуемых действий по команде, восстановление старых значений системных переменных.

Некоторые системные переменные:

CMDECHO

Режим эхо-вывода (отображение подсказок AutoCAD) при выполнении функции command языка AutoLISP.

VLIPMODE

Режим видимости маркеров.

PLINEWID

Ширина полилинии по умолчанию

CELTYPE

Текущий тип линии

3.

Запоминание значений системных переменных можно выполнить прочитав их с помощью функции

(GETVAR <имя системной переменной>).

Например:

```
(Setq ECHO (GetVar "CMDECHO")) ; считывание системной переменной
;CMDECHO и запоминание этого значения в переменной пользователя
;ECHO.
```

Запоминать и восстанавливать необходимо значения тех системных переменных, которые будут изменяться в команде пользователя.

Например: системная переменная CMDECHO имеет значение 0 – эхо (запросы) команд системы отключено, 1 – эхо включено.

Системная переменная VLIPMODE - 0 – отключен вывод маркеров в поле чертежа, 1 - включены маркеры.

Если Ваша команда изменяет тип линии – работайте с переменной CELTYPE, если изменяет ширину сегмента полилинии – PLINEWID и так далее.

Установка новых значений системных переменных

Установить необходимые значения системных переменных можно либо при помощи функции Лиспа (SETVAR <имя переменной> <значение>), либо при помощи соответствующей команды Автокада.

Например:

(SETVAR “VLIPMODE” 0) ;отключение маркеров.

(SETVAR “CMDECHO” 0) ;отключение запросов команд системы.

(SETVAR “PLINEWID” 1) ; установка ширины сегмента полилинии.

(COMMAND “PLINE” P1 P2); рисует сегмент полилинии от P1 до ;P2 шириной =1.

Выполнение вычислений

Для вычисления координат точек можно использовать функцию

(POLAR <точка><угол><расстояние>), вычисляющую координату заданной точки в плоскости построения на основании <точки>, <угла> и <расстояния>, либо функции работы со списками

(LIST A B) - объединяет A и B в список (может быть координатой точки).

(CAR P) – выделяет координату X точки P (первый элемент списка).

(CADR P) - выделяет координату Y точки P (второй элемент списка).

(CADDR P)- выделяет координату Z точки P (третий элемент списка).

Вычисление переменных следует выполнять с помощью функций Лиспа, выдающих требуемый тип результата.

Вычисление угла образуемого вектором, заданным координатами P1 и P2 и осью X в плоскости построения можно выполнить функцией

(ANGLE P1 P2) - возвращает угол в радианах (измеряется против часовой стрелки).

Вычисление расстояния между точками в поле чертежа можно выполнить функцией

(DISTANSE P1 P2), возвращающей расстояние между точками P1 и P2.

Вычерчивание фрагмента чертежа

Фрагмент чертежа можно выполнить программой на Лиспе, используя функцию COMMAND, позволяющую включать команды Автокада в программу на Автолиспе.

(COMMAND <аргумент1> <аргумент2> ...) аргумент это команда Автокада в кавычках и ее опции (ответы на запросы команды).

Например:

(COMMAND "LINE" P1 P2 P3 P4 "") Вычерчивает отрезки от P1 к P2 к P3 и к P4.

(COMMAND "") – равносильно нажатию пробела на клавиатуре.

(COMMAND) – равносильно прерыванию большинства команд системы.

Для того, чтобы использовать команду Автокада в функции (COMMAND...) необходимо знать протокол ее выполнения (знать последовательность запросов команды и иметь ответы на эти запросы).

- функция загрузки файла команды (функции пользователя) :

(load <файл> [<сообщение>])

Здесь <файл> – это строка с именем загружаемого файла, а <сообщение> – это строка с текстом сообщения, которое нужно вывести, если загружаемый файл не будет обнаружен. Возвращаемое значение в случае удачной загрузки – результат последнего выражения или имя последней функции, если в файле находятся определения функций. В случае если загружаемый файл не найден, возвращается текст сообщения об ошибке.

Если расширение в имени файла опущено, то система AutoCAD пытается выполнить загрузку, поочередно добавляя следующие расширения:

- vlx
- fas
- lsp

Расширение vlx соответствует файлам приложения, fas — компилированным LISP-файлам, lsp — исходным LISP-файлам. Для непосредственного редактирования пользователю доступны только файлы с расширением lsp, т. к. файлы с расширениями vlx и fas не являются текстовыми.

- функция разветвления:

(if <логическое выражение> <выражение тогда> [<выражение иначе>])

Возвращаемое значение: вычисленное значение <выражение тогда>, если значение <логическое выражение> отлично от *nil* (*nil* в логических операциях используется как "ложь"); или значение <выражение иначе>, если значение <логическое выражение> равно *nil* (когда <выражение тогда> опущено, тогда вместо <выражение тогда> возвращается *nil*).

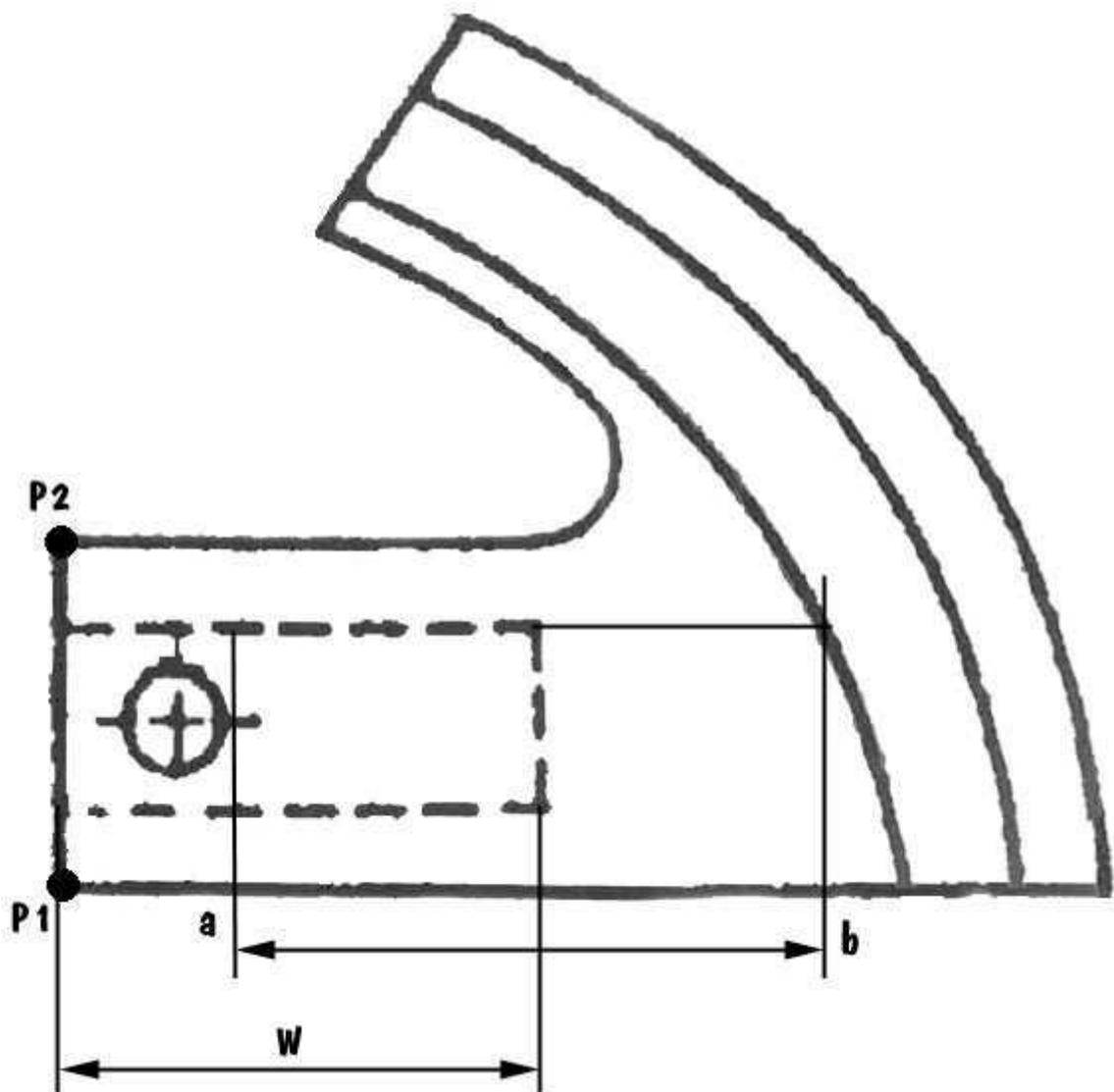
Если функция if используется для выполнения по условию не одного выражения, а нескольких, то эти выражения следует объединить с помощью функции progn (иначе будет синтаксическая ошибка, вызванная неправильным количеством аргументов при обращении к функции if):

(progn <выражение1> <выражение2> ... [<выражениеN>])

Функция объединяет несколько (не менее двух!) выражений в одно, когда по синтаксису языка AutoLISP может использоваться только одно (как в функции if).

Пример оформления отчета

Рисунок задания:



Задание:

Разработать команду, которая по заданным двум точкам P1 и P2 выполняет построение чертежа. Также команда должна запросить ввод ширины (W) стороны прямоугольника, причем если не выполняется неравенство ($a < W < b$), то прямоугольник выводится красным цветом и в командной строке выводится сообщение об ошибке. Команда должна поддерживать возможность задания ширины по умолчанию.

Текст команды:

```
(Defun C:Pavel2()  
; Загрузка файла (LOAD "D:\\AutoCAD\\VBAkyp\\tsilenko2.lsp")  
  
; Вводим точки P1 и P2  
  (Setq P1 (GetPoint "\n Базовая точка №1: ")  
    P2 (GetPoint "\n Базовая точка №2: ")  
  )  
  
; Вводим ширину невидимого объекта  
  (initget 6) (setq W (getreal "Введите ширину (<4.2>): "))  
  (if (not W) (setq W 4.2))  
  (setq WTH (/ W 3))  
  
; Сохраняем параметры среды  
  (Setq OldWidth (GetVar "PLINEWID")  
    OldEcho (GetVar "CMDECHO")  
    OldBlip (GetVar "BLIPMODE")  
    OldCelType (GetVar "CELTYPE")  
  )  
  
; Устанавливаем параметры среды  
  (SetVar "CMDECHO" 0)  
  (SetVar "BLIPMODE" 0)  
  (setVar "CELTYPE" "Continuous")  
  
; Вычисляем коэффициент пропорциональности , угла наклона и  
ширины линии  
  (Setq K (Distance P1 P2)  
    A (Angle P1 P2)  
    Width (/ K 60)  
    Width2 (/ (* Width 2) 3)
```

```

)

; Устанавливаем ширину полилинии
(SetVar "PLINEWID" Width)

; Вычисление центра всех дуг
(Setq C1 (Polar P1 (- A PI) (/ K 3)))

; Вычисление центра маленькой окружности
(Setq C2 (Polar (Polar P1 A (/ K 2)) (- A (/ PI 2)) (/ K
3)) )

; Вычисление точек контура
(Setq P3 (Polar P1 (- A (/ PI 2)) (* 2.33 K))
P4 (Polar P3 (- A (/ PI 2)) (/ K 7.5))
P5 (Polar P4 (- A (/ PI 2)) (/ K 3))
P6 (Polar P5 (- A (/ PI 2)) (/ K 4.29))
P7 (Polar P1 (+ A (/ PI 2)) (/ K 2))
P8 (Polar P7 (- A (/ (* PI 34) 180)) (* 2.29 K))
P9 (Polar P8 (- A (/ (* PI 34) 180)) (/ K 7.5))
P10 (Polar P9 (- A (/ (* PI 34) 180)) (/ K 3))
P11 (Polar P10 (- A (/ (* PI 34) 180)) (/ K 4.29))
P12 (Polar P2 (- A (/ PI 2)) (* 1.33 K))
P13 (Polar C1 (- A (/ (* PI 39) 180)) (* 2.355 K))
P14 (Polar (Polar P1 A (/ K 2)) (- A (/ PI 2)) (/ K
6))
P15 (Polar (Polar P1 A (/ K 2)) (- A (/ PI 2)) (/ K
2))
)

; Вычисление точек невидимого контура
(Setq
P16 (Polar P2 (- A PI) (* K 0.25))
P17 (Polar P1 A (* K 0.25))
P18 (Polar P16 (- A (/ PI 2)) (* K WTH))
P19 (Polar P17 (- A (/ PI 2)) (* K WTH))
)

; Вычисление точек для осевых линий
(Setq F1 (Polar C2 A (* K 0.25))
F2 (Polar C2 (- A (/ PI 2)) (* K 0.25))
F3 (Polar C2 (- A PI) (* K 0.25))
F4 (Polar C2 (+ A (/ PI 2)) (* K 0.25))
)

; Вычисление радиуса маленькой окружности
(Setq C2R1 (/ K 6)
)

; Вычисление данных для надписи

```

```

(Setq
  TextHeight (/ K 12)
  Ptext (Polar P2 (+ A (/ PI 2)) (/ K 6))
)

; Прорисовка видимого контура фигуры
(Command
  "_PLINE" P4 "_A" "_R" C1 P4 P9 ""
  "_PLINE" P5 "_A" "_R" C1 P5 P10 ""
  "_PLINE" P6 "_A" "_R" C1 P6 P11 ""
  "_PLINE" P8 P11 ""
  "_PLINE" P6 P5 P4 P3 P1 P2 P12
  "_ARC" P13
  "_ARC" "_R" C1 P3 P8 ""
  "_PLINE" P14 "_ARC" "_R" C2R1 P15
  "_ARC" "_R" C2R1 P14 ""
)

; Прорисовка невидимых контуров
(if (and (>= W 1.5) (<= W 6.5)) (Command "_COLOR"
"BYLAYER") (progn (Command "_COLOR" 1) (princ "ОШИБКА!\n")))
(setVar "CELTYPE" "Hidden")
(SetVar "PLINEWID" Width2)
(Command "_PLINE" P16 P18 P19 P17 "")
(Command "_COLOR" "BYLAYER")

; Прорисовка осевой линии
(SetVar "CELTYPE" "Center")
(Command "_LINE" F1 F3 ""
  "_LINE" F2 F4 ""
)

; Вывод надписи
(COMMAND "_TEXT" "Justify" "Center" Ptext TextHeight (* (/
A Pi) 180) "Циленко П.Н. группа ИУ6-32")

; Возврат параметров среды
(SetVar "PLINEWID" OldWidth)
(SetVar "BLIPMODE" OldBlip)
(SetVar "CMDECHO" OldEcho)
(SetVar "CELTYPE" OldCelType)

; Завершение команды
(Print )
)

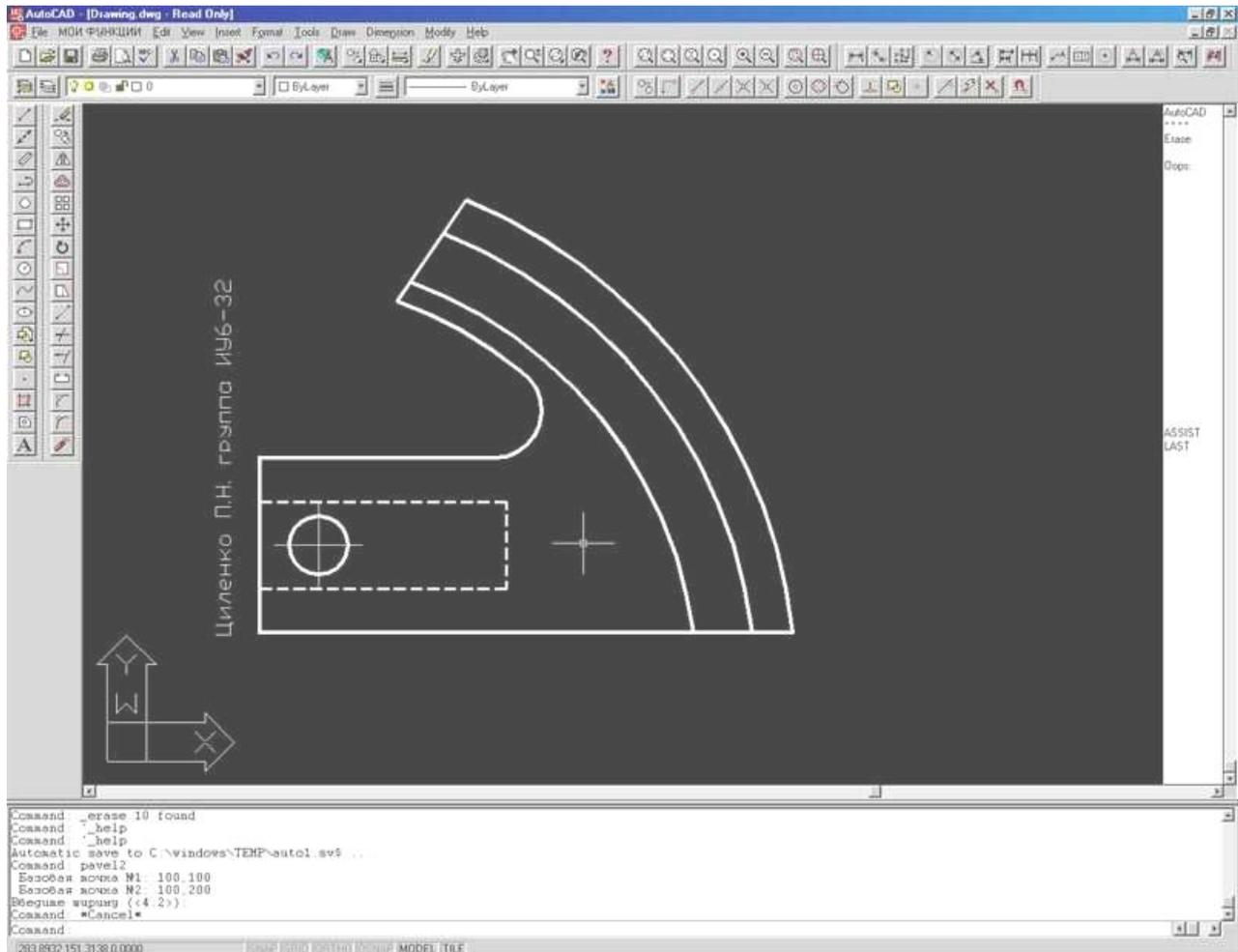
```

Тесты команды:

1.

Command: pavel2 **ENTER**
Базовая точка #1: 100,100 **ENTER**
Базовая точка #1: 100,200 **ENTER**
Введите ширину (<4.2>): **ENTER**
Command:

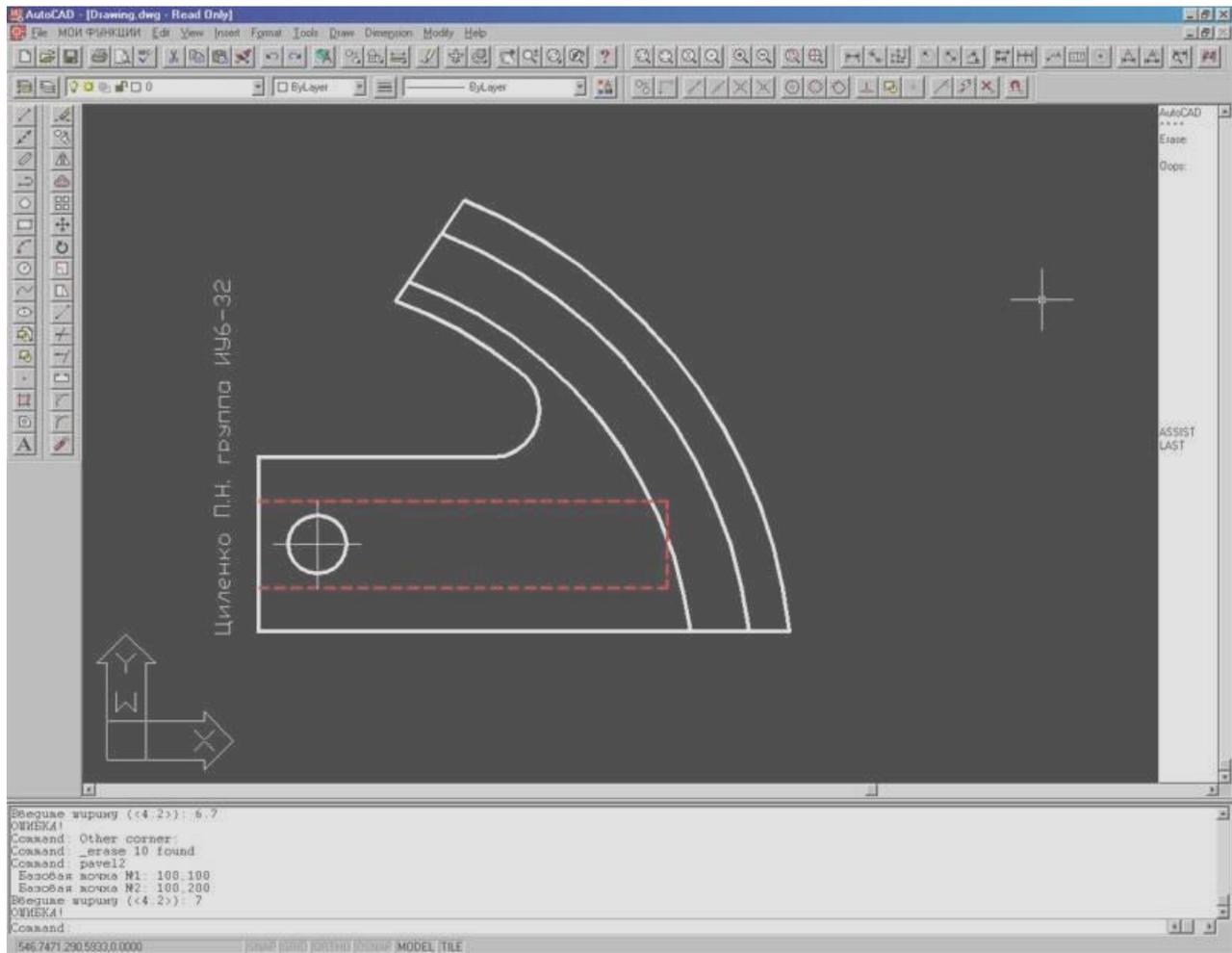
Ширина задается по умолчанию = 4,2 и строится изображение.



2.

Command: pavel2 **ENTER**
Базовая точка #1: 100,100 **ENTER**
Базовая точка #1: 100,200 **ENTER**
Введите ширину (<4.2>): 7 **ENTER**
ОШИБКА!
Command:

Ширина = 7 и НЕ удовлетворяет неравенству $a < 7 < b$ - ошибка
ВЫВОДИТСЯ
и невидимый элемент прорисовывается красным цветом.



3.

Command: pavel2 **ENTER**
 Базовая точка #1: 10,40 **ENTER**
 Базовая точка #1: 100,70 **ENTER**
 Введите ширину (<4.2>): -1 **ENTER**
 Value must be positive and nonzero.
 Введите ширину (<4.2>): 0 **ENTER**
 Value must be positive and nonzero.
 Введите ширину (<4.2>): 1,1 **ENTER**
 ОШИБКА!
 Command:

Ширина = 1,1 и НЕ удовлетворяет неравенству $a < 1,1 < b$ -
 ошибка ВЫВОДИТСЯ
 и невидимый элемент прорисовывается красным цветом.

Работа №1

1. Для заданного преподавателем варианта , подготовьте программу на AutoLISP, реализующую команду пользователя.
2. Проведите отладку программы.
3. Подготовьте отчет о проделанной работе.

Работа №2

1. Напишите команду пользователя для Autocada по заданному варианту, которая включает:
 - ввод данных по протоколу системы (включая ввод по умолчанию);
 - проверку правильности работы пользователя с командой;
 - работу с системными переменными;
 - выполнение необходимых геометрических расчетов;
 - выполнение геометрических построений;
 - восстановление системных переменных.
2. Проведите отладку команды.
3. Подготовьте отчет о проделанной работе.

Работа № 3

1. Познакомьтесь с файлом ACAD.MNU, определите разделы или подразделы в которые Вам задано установить разработанные ранее команды.
2. Сформируйте соответствующие пункты меню.
3. Проведите отладку меню.
4. Подготовьте отчет о проделанной работе.

Содержание отчетов

1. Задание, выданное преподавателем.

2. Программа на LISPе, реализующая команду пользователя или фрагмент файла ACAD.mnu , реализующий меню пользователя в ACADe/
3. Результаты работы команды или компиляции файла меню.

Задание для домашней подготовки

1. Изучите основные функции языка
2. Изучите программу примера.
3. Ознакомьтесь с результатами решения примера.
4. Составьте программу, реализующую команду пользователя в соответствии с вашим вариантом.
5. Продумайте меры по улучшению структуры и интерфейса команды.

Библиографический список

1. Тимофеев В.В. Конспект лекций по дисциплине «Моделирование» МГТУ 2013

Электронные ресурсы.

Сайт кафедры

