

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
MULTIMEDIJOS INŽINERIJOS KATEDRA

Lina Koskutė

**Automatizuotojo projektavimo sistemos AutoCAD galimybių  
papildymo programinės įrangos projektavimas ir tyrimas**

Magistro darbas

Darbo vadovas

doc. A.Lenkevičius

Kaunas,2007

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
MULTIMEDIJOS INŽINERIJOS KATEDRA

Lina Koskutė

**Automatizuotojo projektavimo sistemos AutoCAD galimybių  
papildymo programinės įrangos projektavimas ir tyrimas**

Magistro darbas

Vadovas

doc. A.Lenkevičius  
2007-05

Atliko

Recenzentas

doc. R.Butkienė

2007-05

IFM 1-3 gr. Stud.

Lina Koskutė

2007-05-21

Kaunas,2007

## **Santrauka**

AutoCAD sistemos papildymas sukurtas tam, kad būtų lengviau ir paprasčiau dirbti su AutoCAD grafine sistema.

Funkcijos sukurtos naudojant AutoLisp ir VisualLISP programavimo kalbas.

Sistemos papildymą galima įdiegti į bet kurią AutoCAD versiją. Sukurtas papildymas lankstus naujų funkcijų prijungimui, lengvai eksploatuojamas. Funkcijos suskirstytos į keletą grupių pagal jų formatą.

## **Summary**

AutoCAD system complement is created to make more easy working with AutoCAD graphic system.

Functions are created using AutoLisp and VisualLISP programming languages.

System complement can be implement to whatever release of AutoCAD. Developed complement is flexible for adding new function, maintenance is simple. Functions are divided into some parts by format.

## Turinys

1.	Įvadas .....	6
2.	Analitinė dalis .....	7
2.1.	AutoCAD .....	7
2.2.	Kitos grafinės sistemos .....	12
2.3.	AutoCAD vartotojų klasifikacija .....	13
2.4.	AutoCAD komandų klasifikacija .....	15
	Išvados .....	15
3.	Projektinė dalis .....	16
3.1	Reikalavimų specifikuojimas .....	16
3.1.1.	Projekto varovai .....	16
3.1.2.	Projekto Apribojimai .....	16
3.1.3.	Funkciniai reikalavimai .....	18
3.1.4.	Nefunkciniai reikalavimai .....	21
3.1.5.	Projekto išeiša .....	25
3.2.	Architektūros specifikacija .....	27
3.2.1.	Įvadas .....	27
3.2.2.	Architektūros pateikimas .....	27
3.2.3.	Architektūros tikslai ir apribojimai .....	28
3.2.4.	Panaudojimo atvejų vaizdas .....	28
3.2.5.	Sistemos statinis vaizdas .....	29
3.2.6.	Sistemos dinaminis vaizdas .....	31
3.2.7.	Išdėstymo vaizdas .....	34
3.2.8.	Duomenų vaizdas .....	34
3.2.9.	Kokybė .....	34
3.3	Detali sistemos architektūra .....	35
3.3.1.	Vartotojai .....	35
3.3.2.	AutoCAD sistemos papildymas .....	41
	Sukurtų funkcijų aprašas .....	45
	Naujų funkcijų atitikmenys AutoCAD'e .....	58
	Išvados .....	60
4.	Vartotojo dokumentacija .....	60
4.1.	Sistemos funkcinis aprašymas .....	60
4.2.	Sistemos vadovas .....	61
4.3.	Sistemos instaliavimas .....	61
5.	Ekspirimentinis tyrimas .....	63
6.	Išvados .....	63
7.	Literatūros sąrašas .....	65
8.	Terminų ir santrupų žodynas .....	66
9.	Priedai .....	67

## 1. Įvadas

Šiuo metu sunku rasti tokią sritį, kur grafinius duomenis apdoroti ir vaizduoti kompiuteriu būtų nenaudinga. Tai lemia ypač sparčią kompiuterinės grafikos techninės, programinės ir matematinės įrangos plėtrą. Šiandien jau įprasta, kad kompiuterinė grafika vartojama labai skirtingose srityse, pavyzdžiui, moksle, inžinerijoje, versle, pramonėje, mene, medicinoje, reklamoje, žaidimuose, mokymo, valdymo, treniravimo ir kitose srityse.

Viena populiariausių grafinių automatizuotojo projektavimo sistemų yra AutoCAD, kurioje išplėstas trimatis modeliavimas ir vizualizavimas, ryšys su duomenų bazėmis, yra galimybė dirbti su keliais brėžiniais ir t.t., dirbanti Windows NT 3.51 arba 4.0 ir Windows 9x, 2000 terpėje. AutoCAD sistema turi vienas iš geriausių modifikavimo priemonių, tai viena iš svarbiausių jos triuškinančio pasisekimo priežasčių. Galima sakyti, kad AutoCAD sistema skirta ne tiek vartotojams, o kiek kūrėjams. Galimybė papildyti priedais išauga kiekvienoje naujoje versijoje. Darbo režimas – autonominis arba tinklinis. Pagrindinis sistemos komponentas yra grafinis redaktorius. Anksčiau, ruošiant tekstinius dokumentus, naudojomės tekstiniais redaktoriais: Word, PageMaker, Paint. Grafinis redaktorius – tai toks pat patogus instrumentas, tik skirtas braižymui. Braižymas AutoCAD sistemoje ne tik patogus, bet ir greitas. Sluoksniuoto brėžinio projektavimo sistema leidžia įvesti jau paruoštus detalių variantus, projektuoti statybinius pastatus ir t.t. AutoCAD sistema pasižymi aukšta brėžinių kokybe, greitos korekcijos galimybe, tipinių sprendimų ir prototipų panaudojimu, plačios kūrybos galimybe.

AutoCad yra atvira sistema, t.y. joje leidžiama keisti darbo langų, meniu struktūrą, kurti ir įdiegti savo sukurtas komandas, pašalinti senąsias ar suteikti joms kitokią prasmę. Tą padaryti galima keliais būdais:

1. pakeisti ankstesnius meniu failus naujais;
2. sukurti makrokomandų rinkinius-scenarijus(script failus);
3. parašyti programas, automatiškai vykdančias vartotojui reikalingas funkcijas.

Modifikuoti AutoCAD sistemą galima tokiomis programavimo priemonėmis:

- funkcinė programavimo kalba AutoLisp;
- integruota programavimo kalba Visual Lisp;
- programavimo kalba Visual Basic for Application;
- programavimo kalba C;
- simbolinė kalba Diesel;
- dialogų programavimo kalba DCL;
- periferinių įrenginių valdymo kalbomis HPGL, DMPL, DXY ir kitomis.

**Darbo tikslas:** modifikuoti AutoCAD standartinį meniu naujai sukurtomis funkcijomis. Funkcijos kuriamos AutoLisp ir Visual LISP kalbomis.

## 2. Analitinė dalis

### 2.1. AutoCAD

#### Meniu redagavimas ir kūrimas

Meniu failai apibrėžia meniu sričių funkcionalumą ir išvaizdą. Kiekvieno meniu skyriaus (srities) nariai sudaryti iš AutoCAD komandų eilučių, makrokomandų, kurios nustato vykdomus veiksmus, kai atitinkamas meniu narys parenkamas. Standartinio AutoCAD meniu išėties failas yra ACAD.MNU. Meniu failai apibrėžia šias sritis:

- viršutinio meniu (pull-down);
- šoninio meniu;
- įrankių juostų (toolbars);
- grafinio meniu (image);
- nurodymo įrenginio klavišų meniu (pointing- device);
- planšečių meniu (digitizing-tablet);
- susiejančios būsenos eilutės teikiamus pagalbinius pranešimus su viršutiniojo ir įrankių juostų meniu nariais (help strings and tooltips);
- greitinančių klavišų (keyboard accelerators).

Kiekvieną meniu narį bendruoju atveju sudaro vardo identifikatorius, žymė ir meniu makrokomanda (išskyrus grafinio meniu skyrių Image, kuriame vardo identifikatoriai nevertojami). Meniu narys rašomas vienoje eilutėje tokiu formatu:

Vardo\_identifikatorius Žymė Meniu\_makrokomanda

Vardo identifikatorius, sudarytas iš raidžių, skaitmenų ir pabraukimo simbolio susieja viršutiniojo meniu narius ir įrankių juostos piktogramas su atitinkamais būsenos eilutėje teikiamais pranešimais, klaviatūros klavišų seką su atitinkamais viršutiniojo meniu makrokomanda, leidžia perkelti meniu narį iš vienos būsenos į kitą.

Meniu nario žymės formatas ir paskirtis skirtinga kiekviename meniu skyriuje. Žymė rašoma tarp laužtinių skliaustų ([ ]) ir apibrėžia, ką rodyti ekrane. Jeigu žymė baigiasi daugtaškiu – tai šis meniu punktas atveria dialogo langą.

Meniu makrokomanda – tai paprasta komanda, komandų grupė arba išraiška AutoLisp kalba. Jeigu meniu narys yra komanda su parametrais, reikia žinoti parametrų sekos pateikimo eiliškumą. Kiekvienas simbolis, net tarpas, meniu makrokomandoje yra reikšminis. (5, 12)

## AutoLisp

AutoLisp kalbos palaikymas AutoCAD'e atsirado tarp versijų Release 6 ir Release 7 1986 metų birželio mėnesį. Tai buvo pirmoji galinga, bet labai paprastai įsisavinama priemonė. Paprastas inžinierius įsisavinęs AutoCAD, galėjo be ypatingo vargo programuoti AutoLisp'u, tam didelę reikšmę turėjo funkcija `command`. Kol jaunesni imperatoriškų kalbų: C, Pascal ar Basic šalininkai bergždžiai ginčijasi, kuri iš jų yra geresnė, profesionalai su pasisekimu naudoja vieną ar kitą kalbą tinkančią konkrečioms uždaviniams spręsti. Lisp kalba yra atsisikyrusi nuo visų kitų. Funkcinė dirbtinio intelekto valdymo kalba Lisp gali vartoti tokias duomenų struktūras kaip lentelės, sąrašai, objektų aprašymai, failai. Naujų primitivų kūrimui naudojamas funkcinis modelis. Lisp yra funkcijų rinkinys, todėl darbas su sąrašais įkunjamas bazinių funkcijų `car`, `cdr`, `cons`, `atom`, `eq`, `equal` pagalba. Rekursyvos informacijos apdorojimo dėka, Lisp'u galima labai kompaktiškai aprašyti realizacijos funkcijas, kurioms kitose programavimo kalbose prireikia šimto ar tūkstančio kodo eilučių. Lisp kalba sukurta 1956 metais John McCarthy. Nuo to laiko atsirado daug šios kalbos dialektų: BBNLisp, Interlisp, MacLisp, NIL, Franz Lisp, Scheme, Flavors, LOOPS, Common LISP, kuris ir tapo kalbos standartu. Vėliau, kiek pakeistas Common LISP dialektas buvo specializuotas grafiinei aplinkai AutoCAD ir pavadintas XLISP, neilgai trukus - AutoLISP. Kadangi jos interpretatorius yra pati AutoCAD terpė, todėl programuojant AutoLisp išraiškos gali būti tiesiogiai rašomos AutoCAD komandų eilutėje arba įrašomos į failą ir vykdomos vėliau. Daugelis AutoCAD komandų parašytos AutoLisp kalba, pvz., 3DARRAY, BMAKE, PROPERTIES ir kiti. Iš tiesų profesionalių programuotojų AutoLisp'u yra labai nedaug. O ir iš kur jiems atsirasti, jei Amerikoje 90% visų programų parašyta Visual Basic'u. Neįprasta skliaustelinė sintaksė yra pagrindinė reto naudojimo priežastis. Didžiausia efektą AutoLisp'as suteikia kaip "dirbtinio intelekto" kalba.

AutoLisp'e sutinkami tokie duomenys:

- Sąrašai ("lists"), kuriuos sudaro tam tikra tvarka sudaryta narių seka iš atomų ar kitų sąrašų; čia atomas - tai vienas nedalomas sąrašo narys,
- taškinės poros, t.y. dviejų koordinačių atomai, pavyzdžiui:  $(v.u)$ ,
- simboliai, vardais iš raidžių, skaičių ir specialių ženklų, išskyrus `() . , ;`,
- eilutės ("string-constant"),
- realūs skaičiai,
- sveikieji skaičiai, nuo -32768 iki 32767 ribose,
- bylų aprašai ("file-descriptor"), įvertinantys vardą, maršrutą, režimą ir t.t.,
- AutoCAD primitivų vardai, LINE, CIRCLE, PLINE...,
- AutoCAD rinkiniai, pavyzdžiui, pažymėta *kertančiuoju langu* primitivų grupė,



- vidinės funkcijos (“subr”), esančios AutoCAD branduolyje,
- išorinės funkcijos, parašytos AutoCAD aptarnavimui ar modifikavimui.

AutoLisp funkcija į AutoCAD programą įkeliami pasirinkus Tools → AutoLisp→Load.  
(5, 7)

### Visual LISP

Pavadinimo Visual Lisp pirmasis žodis tėra tik reklaminis triukas, nes jokio „vizualaus“ konstravimo kaip kitose Visual sistemose, išskyrus iš anksto ranka parašytus dialogus, čia nėra. Tai tik nuostabus redaktorius su kompiliatoriumi. Daugelis laukė kol Visual Lisp taps tikrai „vizualiu“, bet Autodesk firma pasuko kitu keliu t.y. įdiegė į AutoCAD‘ą VBA. Atrodė, kad AutoCAD nebesidomėjo senu geru Lisp‘u, nors būtent jo dėka AutoCAD tapo populiarus. Visual Lisp palaiko visas AutoLisp funkcijas. Tai reiškia, kad visos senos programos gali dirbti naujose AutoCAD versijose. Visual Lisp‘e AutoLisp funkcijų sąrašas papildytas naujomis funkcijomis, patogios, paprasta sąsaja programoms rašyti, derinti ir testuoti: Visual Lisp= AutoLisp + integruota programavimo terpė+Visual Lisp funkcijos. Galima išskirti kelias naudingas Visual Lisp galimybes:

- išėties teksto sintaksės tikrinimas ir sintaksinių klaidų pažymėjimas;
- visas AutoLisp žinynas, tame tarpe ir kontekstinis;
- kompiliatorius File Compilation programoms sparčiau vykdyti;
- neribotas atveriamų failų skaičius atskiruose languose;
- derinimo režimas klaidoms lokalizuoti, kai programos vykdomos pažingsniui;
- kintamųjų reikšmių peržiūrėjimas specialiuose languose programos vykdymo metu;
- įvykdytų operacijų protokolo saugojimas su jų pakartotinio panaudojimo galimybe;
- programų kūrimas ir kompiliavimas naudojant programų kūrimo žymę;
- AutoLisp ekumulatorius, vaizdžiai iliustruojantis programos vykdymo procesą.

Dar viena svarbi savybė, kad Visual Lisp palaiko ActiveX technologiją sukurtą Microsoft firmos. Ši technologija leidžia vykdyti kliento – serverio architektūrą. Dėl ActiveX technologijos iš Visual Lisp galima naudotis duomenų baze, kurioje informacija saugoma realicinėse duomenų lentelėse.

Visual Lisp iškviečiama AutoCAD komanda Tools →AutoLisp → Visual LISP Editor. AutoLisp failas įkeliamas **File → Open File**, atvertame dialogo lange pasirenkamas reikiamas failas ir paspaudžiama Open. AutoLisp failas įkeliamas iš Visual Lisp į AutoCAD (įkeltas į Visual Lisp): **Tools → Load Text in Editor**. AutoLisp failas įkeliamas į AutoCAD (neįkeltas į Visual Lisp) : **File →Load File...** . (5,7)

## C++

1989 metais AutoCAD'e atsiranda ADS (AutoCAD Development System) sistema leidžianti modifikuoti C kalba. Tai buvo „C sąsaja“, kuri leido AutoLisp'ą papildyti reikalingomis funkcijomis. Iš pradžių ADS modifikavimams buvo galima naudoti skirtingus C kalbos kompiliatorius. Galiausiai ADS išaugo į ARX (AutoCAD Runtime Extension), suteikianti objektiškai orientuotą interfeisą paruoštą su C++. ARX dirba toje pačioje adresu erdvėje kaip ir AutoCAD. Ji dirba kaip dinaminė biblioteka (DDL) ir turi tiesioginį priėjimą į grafinę AutoCAD duomenų bazę. Svarbiausia ARX ypatybė buvo ta, kad atsirado galimybė papildyti AutoCAD ne tik naujomis funkcijomis, bet ir naujais primityvais. (7)

### Visual Basic for Applications (VBA)

VBA (Visual Basic for Application) yra VB kalbos variantas, kuris naudojamas Microsoft Office programose (Word, Excel, PowerPoint, Outlook, Access) ir „Autodesk“ grafinėje sistemoje AutoCAD. VBA įvedimas į AutoCAD'ą, turintį galingas modifikavimo priemones kaip Lisp ir ARX buvo tik noras daryti pinigus kartu su Microsoft'u. Visual Basic for Application populiarus tik dėl to, kad:

- Visual Lisp žino nedaug programuotojų;
- Basic kalba sukurta pradedantiesiems, o VBA tęsia šią tradiciją.

VBA šalininkai bijo Lisp'o dėl to, kad jame „paslaptingos komandos ir ištisi skliaustai“. Pagrindiniai VBA privalumai:

- Skirtingai nei Lisp, VBA iš tikrųjų yra „vizualus“ t.y., turi geras priemones dialogų kūrimui ir integruotą sritį programavimo palengvinimui;
- Stipriai integruotas su Windows, turi nuosavas funkcijas, dauguma jų dirba ActiveX'e;
- Iš VBA galima užkrauti DDL biblioteką ir naudotis jos funkcijomis, to iš Visual Lisp padaryti negalima.
- Visi AutoCAD sistemos kodai naudojantys ActiveX technologija parašyti VBA ir pradedantis programuotojas iš karto gali juos naudoti, Visual Lisp kodus visų pirma reiktų išversti į kitą kalbą.

VBA trūkumai tie, kad:

- VBA sąveikauja su AutoCAD per ActiveX, kuri yra ribota. ActiveX technologija gali dirbti tik su tais objektais ir metodais, kurie yra įdiegti AutoCAD'e.
- VBA negali sąveikauti su AutoLisp. ActiveX paprogramės negali iškviešti AutoLisp paprogramių.

- VBA negalima perduoti parametrų, duomenų pakeitimas įmanomas tik per failus, sisteminius AutoCAD kintamuosius ar DWG failus.

Visi šie trūkumai apriboja VBA naudojimo sritį. VBA (Visual Basic for Application) redaktorius grafinėje sistemoje AutoCAD paleidžiamas per meniu **Tools/Macro** suaktyvinus komandą **Visual Basic Editor**. Kviečiant anksčiau sukurtą projektą, komanda **Load Project** aktyvinamas meniu langas „Open VBA Project“ arba komanda **VBA Manager** aktyvinamas meniu langas „VBA Manager“ ir pasirinkus iškviečiamas atitinkamai su mygtuku [Open] arba [Load] . Iškviečiant projektą, visi projektą sudarantys elementai (dialogo langai, moduliai, klasės ir t. t.) bei redaktorius pakraunami automatiškai. VBA projektams kurti skiriami redagavimo laukai. Redaguojant dialogo langą, naudojamas grafinis redagavimo laukas, redaguojant projekto kodus – teksto redagavimo laukas. Kiekvienam dialogo langui, projektui, moduliui skiriamas tam tikras redagavimo laukas. (7, 8)

### DXF formatas

DXF formatas taip pat neatsiejama AutoCAD dalis. Bet kokie vartotojo duomenys pavaizduoti ekrane, gali būti išreikšti DXF formatu. Remiantis DXF formato charakteristika, galima manipuluoti objektų parametrais, juos tikrinti. DXF kodo nereikia kurti pačiam, nes jį braižant objektą, AutoCAD sistema sukuria automatiškai. DXF failo sandara atrodo taip:

1. Header skyrius – pagrindinė informacija apie piešinį. Kiekvienas parametras turi savo vardą ir ilgį.
2. Tables skyrius – šį skyrių sudaro apibrėžimai, tokie kaip:
  - linijos tipas
  - sluoksnis
  - teksto stilius
  - vartotojo koordinatinių sistema
3. Blocks section – informacija apie blokus.
4. Entities skyrius – primityvo informacija įskaitant blokų aprašus.
5. Failo pabaiga.

Taip pat nėra būtina kodo išsaugoti išoriniame faile. Išsaugojus brėžinį DXF formatu, AutoCAD sistema jį visada atkuria. Tai patogiu, jei kietajame diske nėra daug vietos, nes darbas išsaugotas \*.dxf formatu, užima mažiau vietos negu jis būtų išsaugotas \*.dwg formatu. (5, 12)

### XML formatas

XML (Extensible Markup Language) buvo sukurta informacijos paskirstymui pasauliniame voratinklyje (World Wide Web). Kalbos specifikacija pasirodė 1998 metais.

XML primena daugeliui žinomą hipertekstinę kalbą HTML. Tačiau, skirtingai nuo jos, XML galima sukurti savo elementus, nustatyti jiems vardus ir atributus. Todėl XML galima naudoti bet kurio dokumento rašymui, nuo paprastų žiniaraščių iki didelių AutoCAD brėžinių. Daugelis programų nustatymų yra saugoma XML formatu. Ši kalba naudojama:

- Iliustruoto meniu piešimui;
- Duomenų laikymui;
- Duomenų lentelių saugojimui ir redagavimui. (7)

## **2.2. Kitos grafinės sistemos**

### **Adobe Photoshop**

Photoshop – programa, kuri leidžia koreguoti ir keisti fotografijas kompiuterio ekrane. Ji yra ko gero labiausiai išvystyta taškinės grafikos programa rinkoje. Pagrindinė Photoshop paskirtis – fotorealistinių vaizdų kūrimas, darbas su skanuotais spalvotais vaizdais, retušavimas, spalvų koregavimas, koliažai, transformacijos, spalvų skaidymas ir kita. Dirbant photoshop‘u nereikia kurti vaizdų nuo nulio: jūs tiesiog redaguojate fotografijas. Photoshop‘as pasižymi ir funkcijų įvairove ir paprastumu. Palaikomos sluoksnių priemonės, vaizdo objektai, programinių priemonių nustatymų išsaugojimas, daugybiniai vaizdai ir pan. Photoshopas toks „gilus“, kad savarankiškas jo nagrinėjimas gali pareikalauti ištisų mėnesių rimtų pastangų. (1, 13, 14)

Savomis funkcijomis nepapildomas.

### **CorelDraw**

Tai galinga kompiuterinės vektorinės grafikos programa. Ji naudojama filmų, žaidimų, fotografijų, automobilių ir projektavimo šakose. Išskirtinė CorelDraw galimybė išsaugoti kelis puslapius, leidžia šią programą ribotai panaudoti ir leidybai.

CorelDraw papildyti galima VBA, kuris atsirado nuo 6-osios versijos . VBA arba tiksliau tariant CorelDraw objekto modelis, leidžia plėtotojams valdyti daugelį CorelDraw aspektų programuojamu kodu. Naudojant VBA galima parašyti mažas dešimties eilučių programas, kurios atliks paprastas užduotis. Taip pat galima sukurti nedideles programas iš šimtų ar tūkstančių eilučių, kurios atliktų sudėtingas užduotis, tačiau tai padaryti yra sunku, o kartais ir neįmanoma arba užima labai daug laiko. Vienas iš svarbiausių VBA bruožų yra tai, kad jums, programuotojui, leidžiama „pataisyti“ tuos kelis atvejus, kai jums atrodo, kad programoje CorelDraw tai padaryta neteisingai arba papildyti funkcijas kuo nors, kas tinka būtent jums. (6, 9)

## **Cadkey**

Tai firmos Micro Control System INC programų paketas, skirtas automatizuotam projektavimui ir braižymui trimatėje erdvėje. Viena iš jo taikymo sričių yra mechaninių detalių projektavimas ir braižymas. Brėžiniai atliekami greitai tiek dvimatėje tiek trimatėje erdvėje. Šis paketas leidžia automatizuoti projektavimo procesą, sutrumpinti dokumentacijos parengimo laiko sąnaudas, pateikti glaustą ir tikslią informaciją. Su Cadkey galima modeliuoti vientisus paviršius, braižyti, keistis duomenimis, redaguoti modelius. Taip pat galima redaguoti 3D modelius, sukurtus kitose grafinėse sistemose. Cadkey turi paprastą ir efektyvią meniu struktūrą. Pranešimų ir reikšmių eilutės padeda orientuotis sistemoje, o lengvai suprantamos komandos užtikrina aukštą darbo našumą. (3, 4, 10)

Savomis komandomis nepapildomas.

## **Solid Works**

Solid Works – galinga projektavimo priemonė, pritaikyta „Windows“ aplinkai ir skirta darbui su asmeniniais kompiuteriais bei jų tinkluose. Solid Works idealiai panaudoja erdvinio modeliavimo pranašumus: visišką parametrizaciją ir visišką dvipusį asociatyvumą; ji neriboja sudėtingų surinkimo komponentų skaičiaus, leidžia dirbti su bet kokios formos kūnais, tarpe jų ir su lankstytais, kurti fotovaizdus, sudėtingus brėžinius, apipavidalinti projekto dokumentaciją ir t.t. Dar vienas esminis Solid Works privalumas – pirmą kartą tokio lygio sistema yra rusifikuota: visos Solid Works vartotojo metodinės priemonės, meniu ir vartotojo sąsaja yra išversti į rusų kalbą.

Solid Works bazinis modeliavimo metodas yra erdvinių kūnų modeliavimas, tačiau tuo atveju, kai nepakanka modeliavimo galimybių, konstruktorius gali kurti sudėtingus paviršius, keletą paviršių sujungti į vieną, pjaustyti, išstempti ir t.t. Solid Works turi daug specialių priemonių, padedančių spręsti praktinius projektavimo uždavinius, atsižvelgiant į gamybos būdą. Su SolidWorks lengva pereiti iš 2D į 3D erdvę. (2, 15)

Savomis priemonėmis nepapildomas.

### **2.3. AutoCAD vartotojų klasifikacija**

#### **„Čainikai“**

Tai pradedantys vartotojai. „Čainikus“ galima tapatinti su kūdikiais, kurie mokosi vaikščioti, bet vėliau, galbūt, bėgios greičiau už visus pasaulyje. Visi mes esame „čainikai“, tik skirtingais klausimais. „Čainikus“ reikia mylėti, gerbti, ir kuriant programas, pasistengti palengvinti jų darbą ir pagreitinti AutoCAD sistemos įsisavinimą.

### **„Tetos“**

Tai atskira „čainikų“ grupė, dažniausiai tai mielos balzakiško amžiaus damos. Nuo paprastų „čainikų“ jos skiriasi tuo, kad puikiai žino tradicinių technologijų projektavimo taisykles. Nors, kai kada tai pradeda trukdyti, pavyzdžiui, „tetos“ sunkiai sugeba piešti objektus natūralaus dydžio. „Tetas“ galima lyginti su žmonėmis, kurie brandžiam amžiuje priversti jodinėti žirgais. Jos yra darbščios ir ištvermingos, nors tokiam amžiuje sunku naudotis kompiuteriu. „Tetas“ taip pat reikia mylėti, gerbti, semtis iš jų patirties ir stengtis rašyti programas taip, kad jos būtų patogios ir suprantamos ir „tetoms“.

### **„Beždžionė su granata“**

Galima palyginti su vaiku, kuris nemoka vaikščioti, bet jau bėgioja, be to kojos juos neša kur nori. Esminis bruožas – nereikia spėlioti ką ji padarys su granata. Granata atstoja laisvai rankose judančią pelę. Paprastai judinti pelę išmokstama greitai. Programuotojams „beždžionė su granata“ yra labai brangus laimikis, nes būtent ji tikrina programos stabilumą neteisingais vartotojų veiksmais.

### **„Normalūs vartotojai“**

Tai buvę „čainikai“ , išmokę dirbti. „Normalų vartotoją“ galima lyginti su paaugusiu vaiku, kuris moka vaikščioti ir jei prireikia gali bėgioti. Jis gali susitvarkyti su dauguma AutoCAD nustatymų ir įsisavinti bet kurias naujoves. „Normalių vartotojų“ dauguma, ir programos turi suteikti jiems galimybę greitai ir efektyviai dirbti su minimaliu nukrypimu nereikalingiems tyrinėjimams.

### **„Profai“**

Skirtumas nuo „normalių vartotojų“, kad jie nenormalūs. Tokių vartotojų skaičius bet kurioje teritorijoje ribotas. „Profai“ mano, kad bet kuri jiems duota programa yra išmokstama. Jie nuolatos kažką „sūpuoja“, „stato“, „griauna“. AutoCAD sistemą „profai“ moka tobulai, bet pats braižymas jiems nebeįdomus. Pasirausti sistemos „viduriuose“ – kitas reikalas. „Profai“ blogai baigia vartotojišką gyvenimą. Jie greitai perauga į programuotojus, „suserga“ lankymusi Internetiniuose forumuose, o galiausiai pradeda rašyti knygas, tokias kokias mes skaitome.

### **„Lameriai“**

Ypatingai nemalonus ambicingų „čainikų“ pogrupis, vadinantis save „profais“. „Lamerių“ galima tapatinti su vaiku, kuris jau moka vaikščioti, nors galvoja, kad jau moka bėgioti. Tėvai žino koks tai pavojingas periodas. Jei visi vaikai suauga, tai dauguma „lamerių“ lieka tokiais visą gyvenimą. Būtent apie „lamerius“ egzistuoja įvairūs programavimo juokeliai, kuriuos mėgsta pasakoti patys „lameriai“.

## „Kietuoliai“

Tai negausi degradavusių „profų“ grupė. Aktyvi veikla jiems įkyri, jie tampa apatiškais ir tik dėl pinigų greitai kuria projektus, nė kiek nesidomėdami aplinkinių poreikiais.

Programos vartotojų klasifikacija parodo kokios reikalingos naujos funkcijos, kad būtų palengvintas kiekvienos grupės vartotojų darbas. AutoCAD atsirado apie 1982m ir nuo to laiko beveik nekeičiamos funkcijų galimybės, labai mažai atsiranda naujų funkcijų, nors labai išsiplėtė sprendžiamų uždavinių ir vartotojų ratas. AutoCAD'ą reiktų kapitaliai perdaryti vartotojų funkcijų poreikio atžvilgiu, tačiau taip nedaroma dėl ypatingai didelio AutoCAD populiarumo ir dėl vartotojams atsirasiančių nepatogumų. (7)

### 2.4. AutoCAD komandų klasifikacija

- Brėžinių failų tvarkymo komandos (file): new, open, close...
- Redagavimo komandos (edit): cut, copy, paste...
- Vaizdo tvarkymo komandos (view): redraw, regen, zoom...
- Objektų įterpimo komandos (insert): block, external reference, raster image...
- Objektų formavimo komandos (format): layer, color, linetype, lineweight...
- Įrankių meniu komandos (tools): spelling, quick select, inquiry...
- Objektų braižymo komandos (draw): line, ray, construction line...
- Matmenų nurodymo komandos (dimension): qdim, linear, aligned, ordinate...
- Pakeitimų komandos (modify): erase, mirror, offset, array...
- Brėžinių langų išdėstymo komandos (window): cascade, tile horizontally,...
- Pagalbos sistemos komandos (help): AutoCAD help, what's new,...

Savo sukurtomis funkcijomis labiausiai papildysiu įrankių komandas, brėžinių braižymo, bei modifikavimo komandų grupes. (11)

### Išvados

1. AutoCAD sistema turi vienas iš geriausių modifikavimo priemonių. Tai viena iš svarbiausių šios sistemos pasisekimo priežasčių, nes daugelis panašių grafinių sistemų savomis funkcijomis nepapildomos.
2. AutoCAD sistema nuo jos sukūrimo pradžios mažai papildoma naujomis funkcijomis, nors vartotojų poreikiai išaugo. Todėl tikslinga papildyti sistemą aktualiomis funkcijomis.
3. AutoCAD sistema skirta įvairiems vartotojams, nuo pradedančiųjų iki profesionalų.
4. Atlikta vartotojų poreikų analizė leido išsiaiškinti trūkstamų funkcijų pobūdį.

### 3. Projektinė dalis

#### 3.1 Reikalavimų specifikavimas

##### 3.1.1. Projekto varovai

###### Projekto kūrimo pagrindas (pagrindimas)

AutoCAD universali Autodesk firmos sukurta ir plėtojama automatizuota projektavimo sistema, kurioje išplėstas trimatis modeliavimas ir vizualizavimas, ryšys su duomenų bazėmis, yra galimybė dirbti su keliais brėžiniais iš karto ir t.t. Ji naudojama įvairiose veiklos srityse: statyboje, mechanikoje, elektrotechnikoje, projektuojant žemėlapius, baldus, drabužius ir t.t. Be daugelio privalumų, kaip ir kiekviena tobula sistema AutoCAD turi ir trūkumų. Mūsų tikslas papildyti AutoCAD sistemą naujomis funkcijomis, taip padarant sistemą patrauklesne vartotojams.

###### Sistemos tikslai (paskirtis)

Naujos funkcijos įdiegiamos į senąjį meniu taip, kad jas būtų taip pat lengva pasirinkti pelės pagalba. Sistemos papildymas naujomis funkcijomis palengvins vartotojų darbą sutaupydamas jų laiką.

###### Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys

Užsakovas. AutoCAD sistemos vartotojai.

Pirkėjas. AutoCAD sistemos vartotojai.

Kiti sprendimus priimančiosios asmenys. Kauno Technologijos Universiteto informatikos fakulteto IFM 1/3gr. magistrantė Lina Koskutė.

###### Vartotojai

- Vartotojo kategorija - Universitetai, projektuotojai.
- Vartotojo sprendžiami uždaviniai (atliekamos funkcijos) – brėžinių projektavimas.
- Patirtis dalykinėje srityje – patyręs.
- Patirtis informacinėse technologijose – patyręs.
- Papildomos vartotojo charakteristikos – vartotojas turi mokėti dirbti AutoCAD sistema.

##### 3.1.2. Projekto Apribojimai

###### 3.1.2.1 Įpareigojantys apribojimai. (Mandated constraints)

###### Apribojimai sprendimui

Kuriamas papildymas turi veikti AutoCAD2004 ir naujesnėse versijose. Techninė įranga tenkinanti AutoCAD sistemos reikalavimus. Kuriamas papildymas turi turėti galimybę praplečiamumui ateityje.



### **Diegimo aplinka**

Reikalingas kompiuteris su įdiegta AutoCAD2004 ar naujesne versija. Operacinė sistema Windows 9x/2000/XP.

### **Bendradarbiaujančios sistemos**

Bendradarbiaujančių sistemų nėra.

### **Komerciniai specializuoti programų paketai**

Funkcionalumui palaikyti reikalinga AutoCAD sistema.

### **Numatoma darbo vietos aplinka**

Specialių reikalavimų nėra, elementari universitetinė arba biuro aplinka.

### **Sistemos kūrimo terminai**

1 lentelė

Eil.nr.	užduotis	Pabaigimo data
1	Reikalavimų specifikavimas	2006 10 15
2	Architektūros specifikavimas	2006 11 15
3	Detalios architektūros specifikacija	2006 12 15
4	Realizacija	2007 01 30
5	Testavimas	2007 02 20
6	Dokumentacijos ruošimas	2007 03 15
7	Taisymas ir užbaigimas	2007 05 15

### **Sistemos kūrimo biudžetas**

Sistema kuriama nekomerciniais tikslais. Universitetas suteikia galimybę naudotis reikalingomis sistemos kūrimui priemonėmis: dėstytojo medžiaga, kompiuterių klase su šiai sistemai reikalinga programine įranga, skaitykla, biblioteka.

#### **3.1.2.2 Terminų žodynas**

DXF – tekstinis brėžinio failo formatas.

Grafinis primityvas – taškas, tiesė, apskritimas ir t.t.

UML – unifikuota modeliavimo kalba.

#### **3.1.2.3.Svarbūs faktai ir prielaidos**

##### **Svarbūs faktai**

Kuriamas papildymas bus prieinamas visiems norintiems juo naudotis.

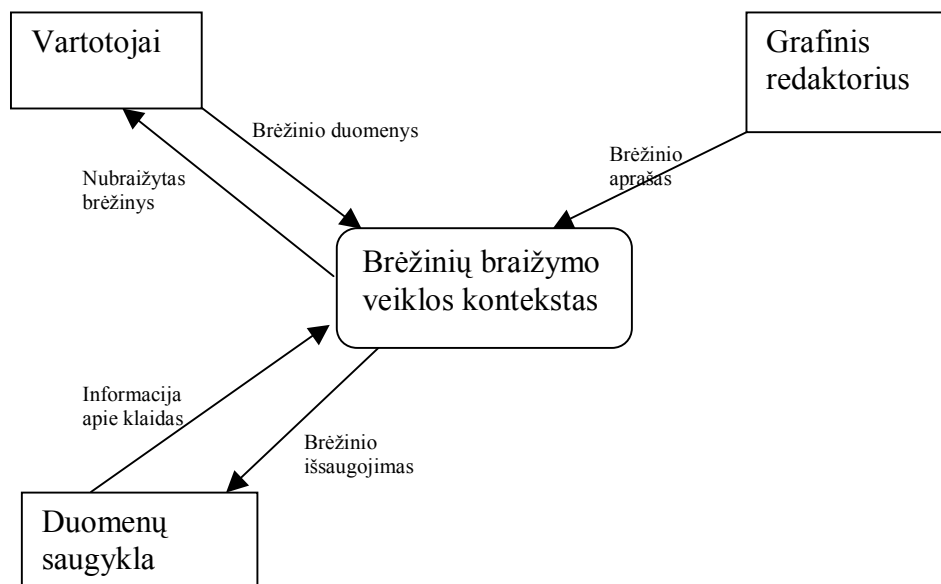
##### **Prielaidos**

Gali atsirasti nauja AutoCAD versija, kurioje jau bus dalis kuriamų funkcijų. Ne visos funkcijos gali būti panaudotos.

### 3.1.3. Funkciniai reikalavimai

#### 3.1.3.1. Veiklos sfera (The scope of the work)

Veiklos kontekstas (pateikiama konteksto diagrama)



1 pav. Veiklos kontekstas

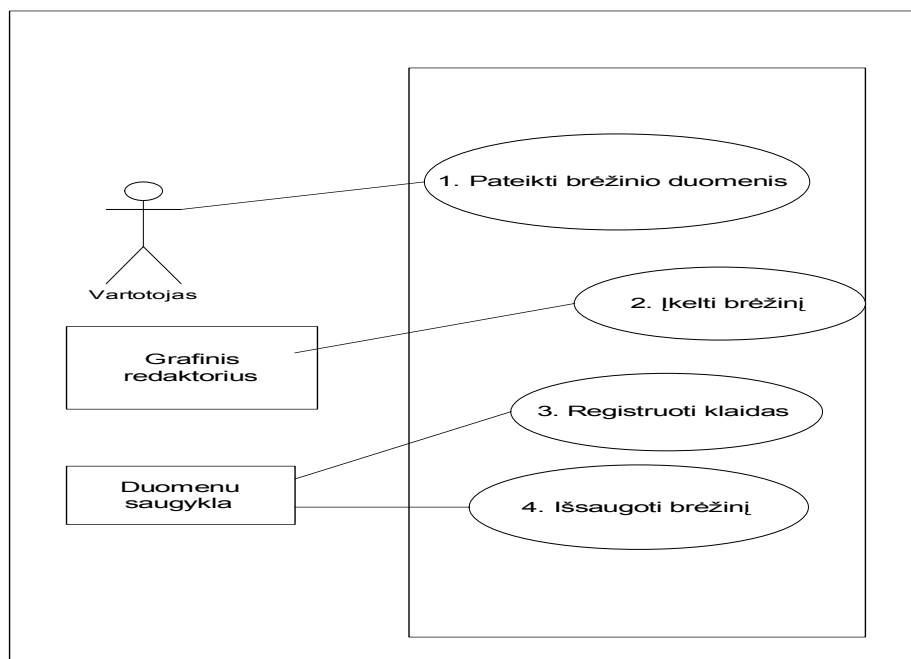
#### Veiklos padalinimas

2 lentelė veiklos įvykių sąrašas

Eil.nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	Vartotojas braižo brėžinį	Brėžinio duomenys (in) Nubraižytas brėžinys (out)
2	Įkeliamas brėžinys	Brėžinio duomenys(in)
3	Duomenų patikrinimas	Pranešimas dėl klaidingų duomenų (out)
4	Duomenų saugojimas	Išsaugotas brėžinys (out)

### 3.1.3.2 Produkto veiklos sfera (The scope of the product)

#### Sistemos ribos



2 pav. Panaudojimo atvejų diagrama

#### Panaudojimo atvejų sąrašas

**1. Panaudojimo atvejis** brėžinio duomenų pateikimas.

**Vartotojas/aktorius** vartotojas.

**Aprašas** apima procesą, kurio metu vartotojas įveda reikalingus duomenis.

**Prieš sąlygą** brėžinys nubraižytas.

**Sužadavimo sąlyga** pasirenkamas naujas brėžinys.

**Po-sąlyga** nubraižomas naujas brėžinys.

**2. Panaudojimo atvejis** įkelti brėžinį.

**Vartotojas/aktorius** braižytojas.

**Aprašas** apima procesą, kurio metu vartotojas įkelia brėžinį išsaugotą dwg formatu.

**Prieš sąlyga** sistemoje nėra brėžinio.

**Sužadavimo sąlyga** atsiranda poreikis įsikelti brėžinį iš kitur.

**Po-sąlyga** sėkmingai įkeltas brėžinys.

**3.Panaudojimo atvejis** registruoti klaidas.

**Vartotojas/aktorius** duomenų saugykla.

**Aprašas** apima procesą, kurio metu sistema informuoja, kad įvelta klaida.

**Prieš sąlyga** klaidų nėra.

**Sužadavimo sąlyga** įvedami duomenys su klaidomis.

**Po-sąlyga** užregistruota nauja klaida.

**4.Panaudojimo atvejis** išsaugojimas.

**Vartotojas/aktorius** vartotojas.

**Aprašas** apima procesą, kurio metu vartotojas išsaugo brėžinį duomenų saugykloje.

**Prieš sąlyga** brėžinio saugykloje nėra.

**Sužadavimo sąlyga** atsirado naujas brėžinys, kurį reikia išsaugoti.

**Po-sąlyga** vienu brėžiniu daugiau duomenų saugykloje.

### 3.1.3.3 Funkciniai reikalavimai ir reikalavimai duomenims

#### Funkciniai reikalavimai

**Reikalavimas #:** 1      **Reikalavimo tipas:** 12      **Įvykis/panaudojimo atvejis #:**1

**Aprašymas:** sistema turi leisti pasirinkus figūrą įvesti papildomus duomenis nubraižymui.

**Pagrindimas:** reikia skirtingų parametrų figūrų.

**Šaltinis:** vartotojas.

**Tinkamumo kriterijus:** galima bus braižyti skirtingų parametrų tą pačią figūrą.

**Priklausomybės:** nėra.

**Konfliktai:** nėra.

**Papildoma medžiaga:**

**Istorija:** užregistruotas 2005 spalio 20d.

**Reikalavimas #:** 2      **Reikalavimo tipas:** 12      **Įvykis/panaudojimo atvejis #:**1

**Aprašymas:** turi leisti keisti brėžinio duomenis.

**Pagrindimas:** vartotojui suklydus reikalingas pataisymas.

**Šaltinis:** vartotojas.

**Tinkamumo kriterijus:** galima perbraižyti figūrą.

**Priklausomybės:** 1

**Konfliktai:** nėra.

**Papildoma medžiaga:**

**Istorija:** užregistruotas 2005 spalio 20d.

**Reikalavimas #:** 3

**Reikalavimo tipas:** 12

**Įvykis/panaudojimo atvejis #:**3

**Aprašymas:** turi pranešti klaidas.

**Pagrindimas:** su netinkamais duomenimis negalima vykdyti operacijų.

**Šaltinis:** vartotojas.

**Tinkamumo kriterijus:** galima greitai pastebėti klaidas.

**Priklausomybės:** 1, 2

**Konfliktai:** nėra.

**Papildoma medžiaga:**

**Istorija:** užregistruotas 2005 spalio 28d.

**Reikalavimas #:** 4

**Reikalavimo tipas:** 12

**Įvykis/panaudojimo atvejis #:** 2

**Aprašymas:** galima įkelti brėžinį.

**Pagrindimas:** vartotojas turi turėti galimybę naujomis funkcijomis redaguoti anksčiau nubraižytus brėžinius.

**Šaltinis:** vartotojas.

**Tinkamumo kriterijus:** galima redaguoti ankstesnius brėžinius.

**Priklausomybės:** nėra.

**Konfliktai:** nėra.

**Papildoma medžiaga:**

**Istorija:** užregistruotas 2005 lapkričio 11d.

#### 3.1.4. Nefunkciniai reikalavimai

##### Reikalavimai sistemos išvaizdai (Look and feel)

**Reikalavimas #:** 7

**Reikalavimo tipas:** 13

**Įvykis/panaudojimo atvejis #:**1-4

**Aprašymas:** lengvai valdoma sąsaja.

**Pagrindimas:** vartotojui turi būti lengva valdyti punktus.

**Šaltinis:** vartojas.

**Tinkamumo kriterijus:** vartotojas nesivaržo naudotis.

**Priklausomybės:** nėra.

**Konfliktai:** nėra.

**Papildoma medžiaga:**

**Istorija:** užregistruotas 2006 balandžio 10d.

### Reikalavimai panaudojamumui (Usability)

**Reikalavimas #:** 8

**Reikalavimo tipas:** 14

**Įvykis/panaudojimo atvejis #:** 1-4

**Aprašymas:** paprasta naudotis.

**Pagrindimas:** neturi būti poreikio papildomiems apmokymams.

**Šaltinis:** vartotojas.

**Tinkamumo kriterijus:** vartotojas gali dirbti iškart.

**Priklausomybės:** nėra.

**Konfliktai:** nėra.

**Papildoma medžiaga:**

**Istorija:** užregistruotas 2006 balandžio 10d.

**Reikalavimas #:** 9

**Reikalavimo tipas:** 14

**Įvykis/panaudojimo atvejis #:** 1-4

**Aprašymas:** universalumas.

**Pagrindimas:** papildymui naudojamos standartinės funkcijos.

**Šaltinis:** vartotojas.

**Tinkamumo kriterijus:** galimybė naudotis visiems.

**Priklausomybės:** nėra.

**Konfliktai:** nėra.

**Papildoma medžiaga:**

**Istorija:** užregistruotas 2006 balandžio 10d.

### Reikalavimai vykdymo charakteristikoms (Performance)

**Reikalavimas #:** 10

**Reikalavimo tipas:** 17

**Įvykis/panaudojimo atvejis #:** 1-2

**Aprašymas:** 100% tikslumas.

**Pagrindimas:** brėžiniai turi būti tokių matmenų kokie nurodomi.

**Šaltinis:** vartotojas.

**Tinkamumo kriterijus:** tikslūs brėžiniai.

**Priklausomybės:** nėra.

**Konfliktai:** nėra.

**Papildoma medžiaga:**

**Istorija:** užregistruotas 2006 balandžio 10d.

**Reikalavimas #:** 11

**Reikalavimo tipas:** 17

**Įvykis/panaudojimo atvejis #:** 1-4

**Aprašymas:** efektyvumas.

**Pagrindimas:** neturi naudoti papildomų resursų trukdydamas kitoms sistemoms.

**Šaltinis:** vartotojas.

**Tinkamumo kriterijus:** darbas be papildomų resursų.

**Priklausomybės:** nėra.

**Konfliktai:** nėra.

**Papildoma medžiaga:**

**Istorija:** užregistruotas 2006 balandžio 10d.

#### **Reikalavimai veikimo sąlygoms (Operational)**

**Reikalavimas #:** 12

**Reikalavimo tipas:** 15b

**Įvykis/panaudojimo atvejis #:** 1-4

**Aprašymas:** papildymas turi veikti turimoje AutoCAD sistemos aplinkoje.

**Pagrindimas:** papildymas skirtas AutoCAD sistemai.

**Šaltinis:** vartotojas.

**Tinkamumo kriterijus:** sistema turi veikti ne senesnėje kaip AutoCAD2004 sistemoje.

**Priklausomybės:** nėra.

**Konfliktai:** nėra.

**Papildoma medžiaga:**

**Istorija:** užregistruotas 2006 balandžio 10d.

#### **Reikalavimai sistemos priežiūrai (Maintainability and portability)**

**Reikalavimas #:** 13

**Reikalavimo tipas:** 16c

**Įvykis/panaudojimo atvejis #:** 1-4

**Aprašymas:** galimybė įkelti į naujesnes AutoCAD versijas.

**Pagrindimas:** AutoCAD vis išleidžia naujas versijas.

**Šaltinis:** vartotojas.

**Tinkamumo kriterijus:** meniu veikia naujose versijose.

**Priklausomybės:** nėra.

**Konfliktai:** nėra.

**Papildoma medžiaga:**

**Istorija:** užregistruotas 2006 balandžio 10d.

### Reikalavimai saugumui (Security)

<b>Reikalavimas #:</b> 14	<b>Reikalavimo tipas:</b> 17	<b>Įvykis/panaudojimo atvejis #:</b> 1-4
<b>Aprašymas:</b> papildymas apsaugotas nuo neteisėto taisymo.		
<b>Pagrindimas:</b> vartotojas gali sugadinti tikslų brėžinių braižymą.		
<b>Šaltinis:</b> kūrėjas.		
<b>Tinkamumo kriterijus:</b> negalima iškraipyti figūrų.		
<b>Priklausomybės:</b> nėra		<b>Konfliktai:</b> nėra
<b>Papildoma medžiaga:</b>		
<b>Istorija:</b> užregistruotas 2006 balandžio 10d.		

### Kultūriniai-politiniai reikalavimai

<b>Reikalavimas #:</b> 15	<b>Reikalavimo tipas:</b> 18	<b>Įvykis/panaudojimo atvejis #:</b> 1-4
<b>Aprašymas:</b> politinis korektiškumas.		
<b>Pagrindimas:</b> negalima naudoti įžeidžiančių terminų, žargonų.		
<b>Šaltinis:</b> vartotojas.		
<b>Tinkamumo kriterijus:</b> naudojami žargonai gali atbaidyti kai kuriuos vartotojus.		
<b>Priklausomybės:</b> nėra.		<b>Konfliktai:</b> nėra.
<b>Papildoma medžiaga:</b>		
<b>Istorija:</b> užregistruotas 2006 balandžio 10d.		

### Teisiniai reikalavimai

<b>Reikalavimas #:</b> 16	<b>Reikalavimo tipas:</b> 7	<b>Įvykis/panaudojimo atvejis #:</b> 1-4
<b>Aprašymas:</b> turi vadovautis autorinių teisių įstatymu.		
<b>Pagrindimas:</b> sistemos papildymas turi būti teisėtas, negali būti kopija.		
<b>Šaltinis:</b> autorinių teisių įstatymas.		
<b>Tinkamumo kriterijus:</b> nekopijuojama.		
<b>Priklausomybės:</b> nėra.		<b>Konfliktai:</b> nėra.
<b>Papildoma medžiaga:</b>		
<b>Istorija:</b> užregistruotas 2006 balandžio 10d.		



### **3.1.5. Projekto išeiga**

#### **3.1.5.1. Atviri klausimai (problemos)**

Funkcijų sąrašas bei skaičius nuolat kinta. Sprendimas galimas šių metų pabaigoje.

#### **3.1.5.2. Egzistuojantys sprendimai**

##### **Pagamintos sistemos, kurios gali būti nupirktos**

Kuriamos funkcijos, kurių nėra sistemoje.

##### **Pagaminti komponentai, kurie gali būti panaudoti**

Tokių komponentų nėra.

##### **Galimas pakartotinas panaudojimas**

Kuriant papildymą, galima kai kurių elementų pagrindu kurti naujus.

#### **3.1.5.3. Naujos problemos**

##### **Problemos diegimo aplinkai**

Neturi kilti problemų įdiegiant papildymą į AutoCAD2004 versiją ir naujesnę.

##### **Įtaka jau instaliuotoms sistemoms**

Didelių problemų kilti neturėtų, nes naujesnės versijos dažniausiai palaiko senesnių versijų komandas.

##### **Neigiamas vartotojų nusiteikimas**

Didelio vartotojų priešiško kilti neturėtų, kadangi kuriamos pageidaujamos funkcijos. Galima neigiama vartotojų reakcija tik kaip įprasta vartotojams gavus naują produktą.

##### **Kliudantys diegimo aplinkos apribojimai**

Jokių specifinių apribojimų nėra. Reikalingas tik kompiuteris su AutoCAD programa.

##### **Galimos naujos sistemos sukeltos problemos**

Nenumatyta tokių problemų.

#### **3.1.5.4. Uždaviniai**

##### **Sistemos pateikimo žingsniai (etapai)**

3 lentelė

Nr.	produktas	Terminas
1	Reikalavimų specifikacija	2006 10 15
2	Architektūros specifikacija	2006 11 15
3	Detali architektūros specifikacija	2006 12 15
4	Testavimas	2007 03 10
5	Realizacija	2007 05 10

##### **Vystymo etapai**

Reikalavimų specifikacija – projekto pagrindinių apribojimų nustatymas, funkcinių ir nefunkcinių reikalavimų sudarymas.

Architektūros specifikacija – panaudojimo atvejų sudarymas remiantis UML.

Detalios architektūros specifikacija – detalios architektūros projektavimas.

Testavimas – pagal sudarytus metodus AutoCAD sistemos papildymo testavimas.

Realizacija – pilnas sistemos papildymo vykdymas.

### 3.1.5.5. Pritaikymas

#### Reikalavimai esamų duomenų perkėlimui

Duomenų perkėlimo atlikti nereikia.

#### Reikalingas duomenų transformavimas perkeliant į naują sistemą

Duomenų transformacijos nebus.

### 3.1.5.6. Rizikos

4 lentelė galimos sistemos kūrimo rizikos

Nr.	Rizikos faktorius	tikimybė	įtaka
1	Reikalavimų pasikeitimas	Vidutiniška	Rimta
2	Architektūros pasikeitimas	Vidutiniška	Rimta
3	Programinės įrangos gedimas	Žema	Leistina
4	Projekto dalyvių susirgimas	Žema	leistina

5 lentelė atsitiktinumų (rizikų) valdymo planas

Nr.	Rizikos faktorius	Sprendimas
1	Reikalavimų pasikeitimas	Reikia suderinti reikalavimus su užsakovu. Pasilikti laiko rezervą svarbiems reikalavimų pasikeitimams.
2	Architektūros pasikeitimas	Numatyti laiko rezervą.
3	Programinės įrangos gedimas	Naudoti atsarginę techninę įrangą.
4	Projekto dalyvių susirgimas	Numatyti laiko rezervą.

### 3.1.5.7. Kaina

Projekto kaina gali būti nustatyta tik apytiksliai.

Tačiau šis projektas nėra komercinis, todėl kainos nenustatinėsime.

### 3.1.5.8. Vartotojo dokumentacija ir apmokymas

Dokumentacijos tipas: vartotojo vadovas brėžinių projektavime. Rengiat dokumentaciją vartotojai nedalyvaus, ją rengs projekto kūrėjai. Apmokymai nebus rengiami.

### 3.1.5.9. Perspektyviniai reikalavimai (Waiting room)

Daugiakalbystės įvedimas.

## 3.2. Architektūros specifikacija

### 3.2.1. Įvadas

#### Dokumento paskirtis

Dokumentas skirtas sistemos architektūriniam aprašymui. Jame architektūra apibrėžiama įvairiais požiūriais, ir kiekvienam požiūriui pavaizduoti naudojamas atskiras modelis. Šio dokumento tikslas surinkti ir pateikti svarbius architektūrinius sprendimus, kurie buvo atlikti, projektuojant sistemą. Šis dokumentas tarnauja kaip bendravimo medžiaga tarp programinės įrangos architekto ir kitų komandos narių dėl architektūrinių sistemos kūrimo sprendimų. Šis dokumentas bus pagrindas sudarant sistemos detalią architektūrą.

#### Apibrėžimai ir sutrumpinimai

- DXF – tekstinis brėžinio failo formatas.
- Grafinis primityvas – taškas, tiesė, apskritimas ir t.t.
- UML – unifikuota modeliavimo kalba (Unified Modeling Language).

#### Apžvalga

Dokumentas aprašo AutoCAD programinės įrangos AutoCAD sistemos papildymo architektūrą. Sistemos nefunkciniai reikalavimai ir apribojimai pateikiami skyriuje „Architektūros tikslai ir apribojimai“. Panaudojimo atvejai pateikiami skyrelyje „Panaudojimo atvejų vaizdas“. Sistemos išskaidymas pateikiamas skyrelyje „Sistemos statinis vaizdas“. Procesai aprašomi skyrelyje „Sistemos dinaminis vaizdas“. Sistemos išdėstymas bei techninė įranga pateikiama „Išdėstymo vaizdas“. „Duomenų vaizdas“ skyrelyje pateikiama sistemos duomenų bazės struktūra. Skyriuje „Kokybė“ bus aprašoma kaip architektūra įtakoja sistemos išplečiamumą, pernešamumą, patikimumą ir pan.

### 3.2.2. Architektūros pateikimas

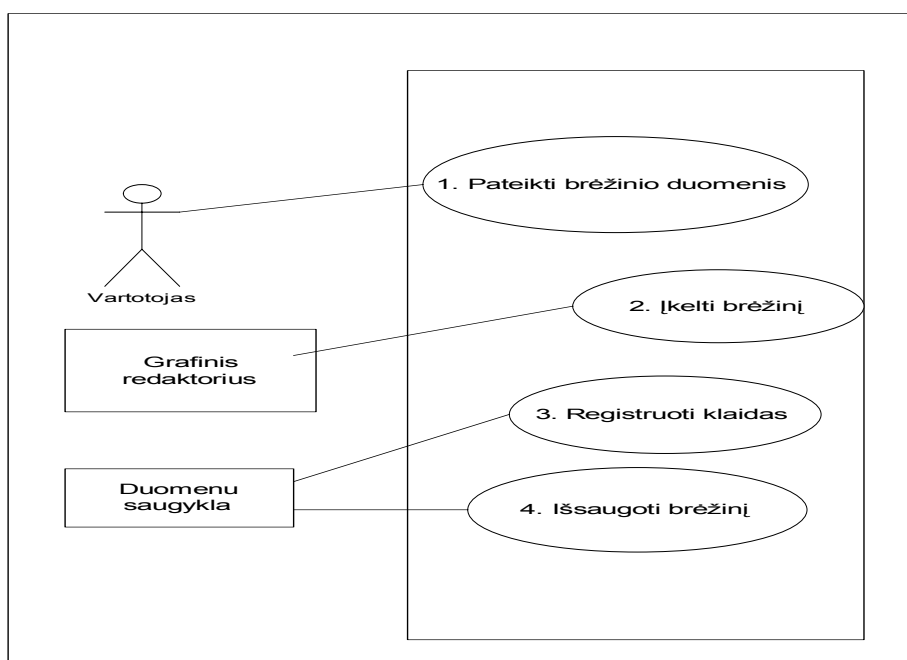
Dokumente sistemos architektūra pateikiama keliais vaizdais: panaudojimo atvejų, statinis, dinaminis ir išdėstymo. Šie vaizdai yra pateikiami naudojant unifikuotą modeliavimo kalbą (UML). Sistemos specifikacija pateikta vaizdais, kuriems įgyvendinti reikia šių UML diagramų:

- Panaudojimo atvejų vaizdas (panaudojimo atvejų diagrama Use case).
- Sistemos statinis vaizdas (klasių diagrama, skaidymas į paketus).
- Sistemos dinaminis vaizdas (būsenų, veiklos, sekų, bendradarbiavimo diagramos).
- Išdėstymo vaizdas (išdėstymo diagrama).

### 3.2.3. Architektūros tikslai ir apribojimai

- Kuriamas sistemos papildymas yra ne komercinis, jis bus pateikiamas kaip atviro kodo sistema.
- Architektūra turi būti parenkama taip, kad ją būtų galima lengvai išplėsti ar prijungti naujus komponentus.
- Sistemos papildymo kūrimui bus naudojama AutoLisp ir Visual Lisp kalbos.
- Sudarant sistemos architektūrą, turi būti atsižvelgta į būtinas programos vykdymo charakteristikas, apibrėžtas reikalavimų specifikacijoje.

### 3.2.4. Panaudojimo atvejų vaizdas



3 pav. Panaudojimo atvejų diagrama

**1. Panaudojimo atvejis** brėžinio duomenų pateikimas.

**Vartotojas/aktorius** vartotojas.

**Aprašas** apima procesą, kurio metu vartotojas įveda reikalingus duomenis.

**Prieš sąlyga** brėžinys n nubraižytas.

**Sužadinimo sąlyga** pasirenkamas naujas brėžinys.

**Po-sąlyga** nubraižomas naujas brėžinys.

## 2. Panaudojimo atvejis įkelti brėžinį.

**Vartotojas/aktorius** braižytojas.

**Aprašas** apima procesą, kurio metu vartotojas įkelia brėžinį išsaugotą dwg formatu.

**Prieš sąlyga** sistemoje nėra brėžinio.

**Sužadinimo sąlyga** atsiranda poreikis įsikelti brėžinį iš kitur.

**Po-sąlyga** sėkmingai įkeltas brėžinys.

## 3. Panaudojimo atvejis registruoti klaidas.

**Vartotojas/aktorius** duomenų saugykla.

**Aprašas** apima procesą, kurio metu sistema informuoja, kad įvelta klaida.

**Prieš sąlyga** klaidų nėra.

**Sužadinimo sąlyga** įvedami duomenys su klaidomis.

**Po-sąlyga** užregistruota nauja klaida.

## 4. Panaudojimo atvejis išsaugojimas.

**Vartotojas/aktorius** vartotojas.

**Aprašas** apima procesą, kurio metu vartotojas išsaugo brėžinį duomenų saugykloje.

**Prieš sąlyga** brėžinio saugykloje nėra.

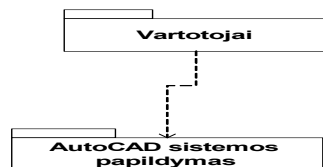
**Sužadinimo sąlyga** atsirado naujas brėžinys, kurį reikia išsaugoti.

**Po-sąlyga** vienu brėžiniu daugiau duomenų saugykloje.

### 3.2.5. Sistemos statinis vaizdas

#### Apžvalga

Kuriamos sistemos paketai:

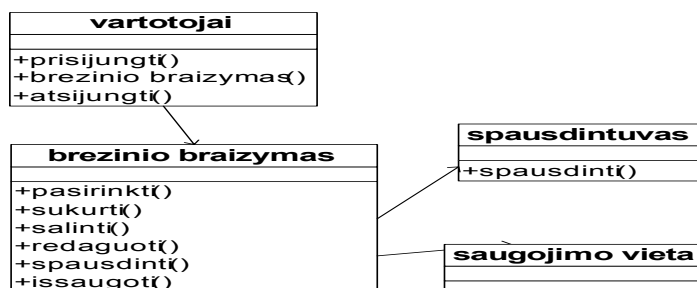


4 pav Suskaidymas į paketus

## Paketų detalizavimas

### Paketas Vartotojai

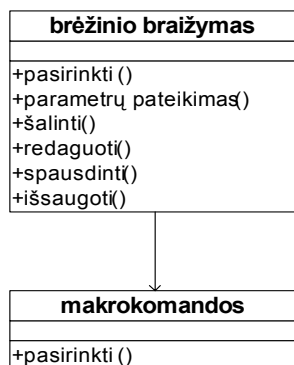
Paketą sudaro klasės per kurias bendraujama su sistema. Prisijungimas prie papildymo, papildymo punktų pasirinkimas, duomenų pateikimas.



5 pav. Paketas vartotojai

### Paketas AutoCAD sistemos papildymas

Sistemos papildymo funkcijos skirtos įvairių brėžinių braižymui, veiksams su jų duomenimis.

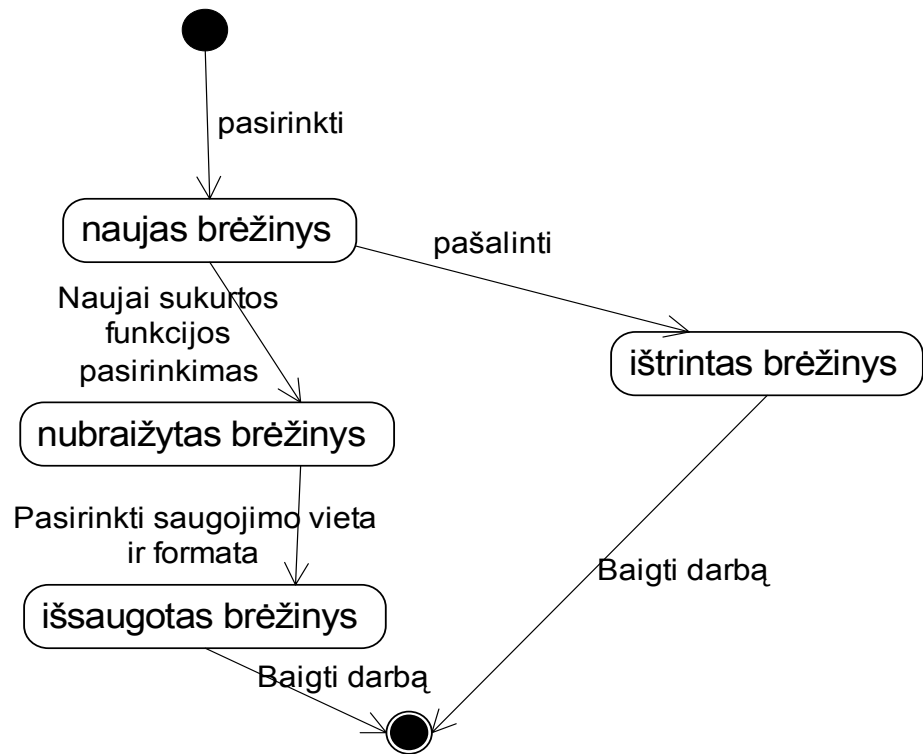


6 pav. Paketas sistemos papildymas

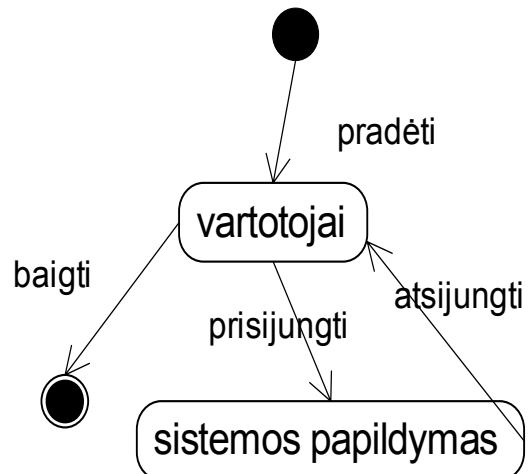
### 3.2.6. Sistemos dinaminis vaizdas

Šiame skyriuje pateikiamos sąveikos, būsenų ir veiklos diagramos.

#### 3.2.6.1. Būsenų diagramos



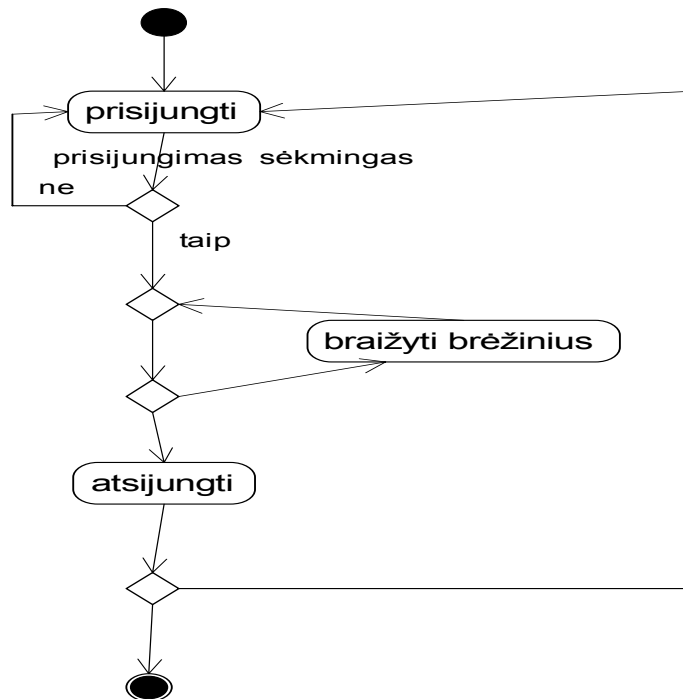
7 pav. Esysbės “brėžinys” būsenų diagrama



8 pav. Esysbės “vartotojai” būsenų diagrama

### 3.2.6.2. Veiklos diagramos

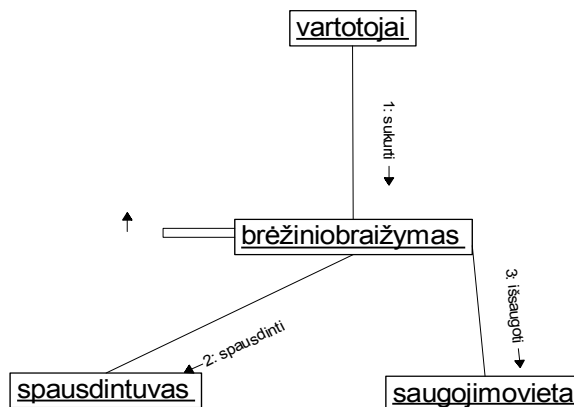
pateikiama vartotojo veiklos diagrama



9 pav. Vartotojų veiklos diagrama

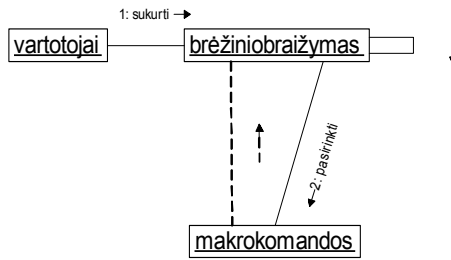
### 3.2.6.3. Sąveikos diagramos

bendradarbiavimo diagramos



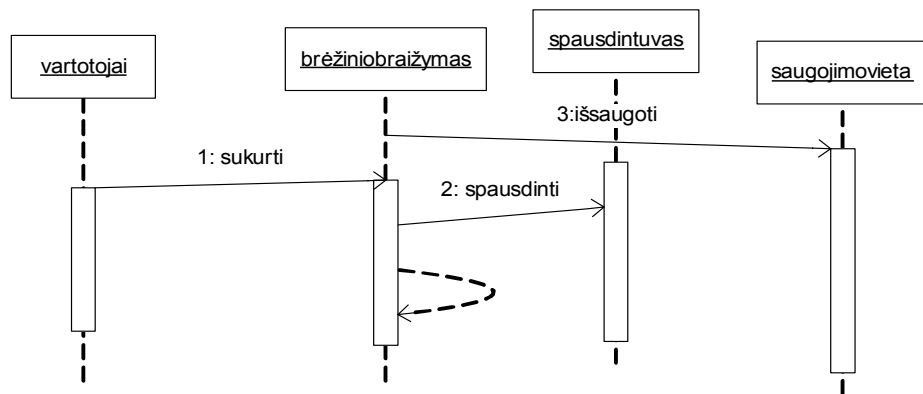
10 pav. Naujo brėžinio sukūrimas



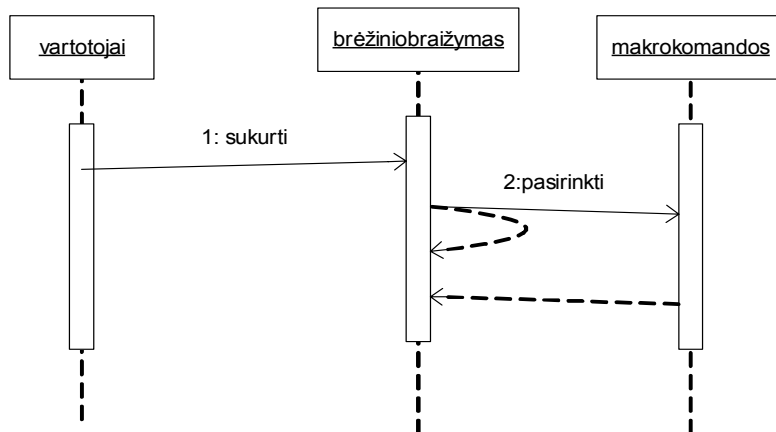


11 pav. Makrokomandų pasirinkimas

sekų diagramos

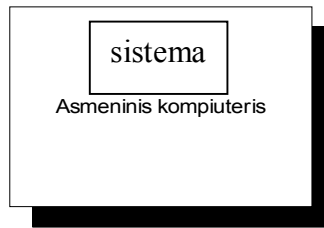


12 pav. Naujo brėžinio sukūrimas



13 pav. Makrokomandų pasirinkimas

### 3.2.7. Išdėstymo vaizdas



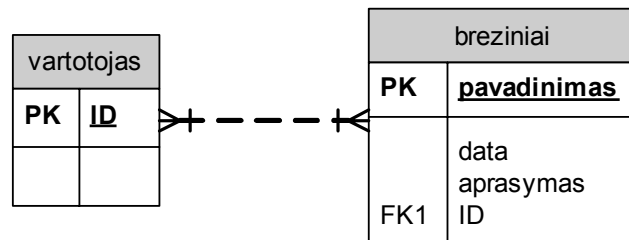
14 pav. Išdėstymo diagrama

#### Minimalūs reikalavimai techninei įrangai:

- Microsoft Windows 9x/2000, XP, AutoCAD 2004 ir naujesnė versija.
- Operatyvioji atmintis  $\geq 64$  MB
- 800\*600 skiriamosios gebos 256 spalvų VGA vaizduoklis ir vaizdo plokštė (rekomenduojama 1024\*768)
- Pentium 133 arba galingesnis procesorius

### 3.2.8. Duomenų vaizdas

AutoCAD duomenų bazė, kurioje saugoma tik informacija apie brėžinius bei brėžiniai.



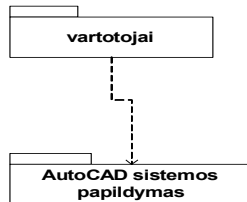
15 pav. Duomenų vaizdas

### 3.2.9. Kokybė

Pasirinkta architektūra leidžia nesunkiai ateityje papildyti kuriamą sistemą naujomis funkcijomis, pakartotinai jas panaudoti kitose sistemose ar panašiuose projektuose.

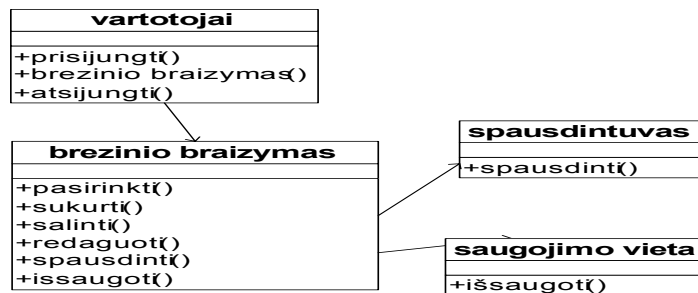
### 3.3 Detali sistemos architektūra

Sistema suskaidoma į šiuos paketus:



16 pav. Suskaidymas į paketus

#### 3.3.1. Vartotojai



17 pav. "Vartotojų" klasių diagrama

#### *Klasifikacija*

paketas

#### *Apibrėžimas*

Paketas nusako vartotojų sąveikavimą su sistema. Čia nusakyti veiksmai, kuriuos gali atlikti vartotojai.

#### *Atsakomybės*

Paketas skirtas užtikrinti, kad vartotojas galėtų lengvai atlikti veiksmus su sistema.

#### *Apribojimai*

Turi užtikrinti duomenų korektiškumą.

#### *Struktūra*

Komponentą sudaro klasės aprašytos pakete “Vartotojai” ir pateiktos diagramoje.

#### *Sąveikavimas*

Komponentas sąveikauja su papildymu. Pagal vartotojo pateiktus duomenis bei pasirinktus papildymo punktus nubraižomas brėžinys.

#### *Resursai*

Naudojama grafinė automatizavimo sistema ne senesnė kaip AutoCAD 2004.

#### *Skaičiavimai*

Aprašyti komponentą sudarančių klasių metodų aprašymuose. Jei metodų aprašymai nepateikiami, vadinasi jie yra suprantami arba paprasti.

#### *Sąsaja/eksportas*

AutoCAD sistemos sąsaja.

#### *3.3.1.1. vartotojai*

#### *Klasifikacija*

klasė

#### *Apibrėžimas*

Klasė skirta vartotojo darbui su sistema.

#### *Atsakomybės*

Brėžinio duomenų pateikimas.

#### *Apribojimai*

Turi užtikrinti duomenų korektiškumą.

#### *Struktūra*

Klasės struktūra pateikta klasių diagramoje.

#### *Sąveikavimas*

Klasė sąveikauja su klase brėžinio braižymas.

#### *Resursai*

Naudojama standartinė AutoCAD biblioteka.

#### *Skaičiavimai*

Pateikti klasių metodų aprašymuose.

#### *Sąsaja/eksportas*

Metodai prisijungti(), brėžinio braižymas(), atsijungti().

#### **Klasės metodai**

**Prisijungti( string vardas, string slaptažodis)**

Atsakomybės

Autentifikuojamas vartotojas sistemoje.

Skaičiavimai

Patikrinama ar egzistuoja vartotojas su nurodytu slaptažodžiu ir prisijungimo vardu. Jei ne, tuomet metama lentelė, kad blogi duomenys.

Sąsaja/eksportas

Vardas – vartotojo prisijungimo vardas

Slaptažodis – vartotojo prisijungimo slaptažodis.

### **Brėžinio braižymas(int brėžinio parametrai)**

Atsakomybės

Naujo brėžinio kūrimas pagal pateiktus parametrus.

Skaičiavimai

Patikrinama ar teisingi įvesti duomenys, ar jų pakanka norimo brėžinio sukūrimui. Jei trūksta kokių nors duomenų, ar jie nekorektiški išvedamas pranešimas „duomenys klaidingi“.

Sąsaja/eksportas

Brėžinio parametrai – vartotojo įvesti duomenys.

### **Atsijungti()**

Atsakomybės

Baigiamas vartotojo darbas su sistema.

Skaičiavimai

Pažymima, kad vartotojas atsijungęs nuo sistemos.

### *3.3.1.2. brėžinio braižymas*

#### *Klasifikacija*

Klasė

#### *Apibrėžimas*

Klasė skirta brėžinių kūrimui.

#### *Atsakomybės*

Pagrindinis sistemos komponentas, braižo įvairius brėžinius.

#### *Aprašymai*

Duomenys turi būti korektiški ir atitikti AutoCAD reikalavimus.

#### *Struktūra*

Klasės struktūra pateikta klasių diagramoje.

#### *Sąveikavimas*

Klasė naudojama vartotojo klasės.

## *Resursai*

Naudojama AutoCAD sistema.

## *Skaičiavimai*

Aprašyti komponentą sudarančių klasių metodų aprašymuose.

## *Sąsaja/eksportas*

Metodai pasirinkti(), sukurti(), šalinti(), redaguoti(), spausdinti(), išsaugoti().

## **Klasės metodai**

### **Pasirinkti (string makrokomanda)**

Atsakomybės

Naujos makrokomandos pasirinkimas.

Skaičiavimai

Pasirenkama funkcija brėžinio kūrimui.

Sąsaja/eksportas

Makrokomanda – vartotojo pasirinkta funkcija.

### **Sukurti (int matmenys)**

Atsakomybės

Naujo brėžinio sukūrimas.

Skaičiavimai

Tikrinama ar duomenys tinkami, jei ne, tuomet išmetamas pranešimas, kad kažką reikia keisti.

Sąsaja/eksportas

Matmenys – vartotojo pateikiami brėžinio matmenys.

### **Šalinti (string brėžinys)**

Atsakomybės

Pašalinti brėžinį.

Skaičiavimai

Pašalinamas kuriamas brėžinys arba tam tikri brėžinio duomenys.

Sąsaja/eksportas

Brėžinys – vartotojo sukurtas brėžinys.

### **Redaguoti (string brėžinys)**

Atsakomybės

Skirtas sukurto brėžinio redagavimui, matmenų keitimui.

Skaičiavimai

Tikrinama ar naujai įvesti duomenys atitinka reikalavimus.  
Sąsaja/eksportas

Brėžinys – tam tikras brėžinys.

### **Spausdinti (string brėžinys)**

Atsakomybės

Brėžinio spausdinimas spausdintuvu.

Skaičiavimai

Nubraižytas brėžinys spausdinamas pasirinktu spausdintuvu.

Sąsaja/eksportas

Brėžinys – vartotojo sukurtas brėžinys.

### **Išsaugoti (string failas)**

Atsakomybės

Brėžinio išsaugojimas faile.

Skaičiavimai

Sukuriamas DWG failas, kuris išsaugomas nurodytu vardu. Tikrinama ar failas tokiu vardu jau yra, ar užtenka saugojimui vietos diske.

Sąsaja/eksportas

Failas – kokiu vardu išsaugoti brėžinį, jei failas nurodytu vardu jau yra išvesti pranešimą.  
Nurodoma kur išsaugoti brėžinį.

#### *3.3.1.3 spausdintuvas*

##### *Klasifikacija*

Klasė

##### *Apibrėžimas*

Klasė skirta palengvinti brėžinių atsispausdinimą.

##### *Atsakomybės*

Nurodytų brėžinių spausdinimas.

##### *Apribojimai*

Nėra.

##### *Struktūra*

Struktūra patekta klasių diagramoje.

##### *Sąveikavimas*

Klasė naudojama brėžinio braižymo klasės.

##### *Resursai*

Naudoja kompiuteryje įdiegtą spausdinimo įrangą.

#### *Skaičiavimai*

Skaičiavimai aprašyti klasės metodų aprašymuose.

#### *Sąsaja/eksportas*

Metodai Spausdinti().

### **Klasės metodai**

#### **Spausdinti(brėžinys)**

Atsakomybės

Brėžinių spausdinimas.

Skaičiavimai

Spausdinamas dokumentas siunčiamas į spausdintuvą, pasinaudojant AutoCAD bibliotekas.

Sąsaja/eksportas

Brėžinys- vartotojo sukurtas ar pasirinktas brėžinys.

#### *3.3.1.4. saugojimo vieta*

#### *Klasifikacija*

Klasė

#### *Apibrėžimas*

Skirta brėžinių išsaugojimui.

#### *Atsakomybės*

Išsaugomas brėžinys vartotojo pasirinktoje vietoje.

#### *Apribojimai*

Brėžinių formatas turi atitikti reikalaujamą AutoCAD sistemos.

#### *Struktūra*

Struktūra pateikta klasių diagramoje.

#### *Sąveikavimas*

Naudojama klasės brėžinių braižymas.

#### *Resursai*

Klasė naudojasi AutoCAD baze bei bibliotekomis.

#### *Skaičiavimai*

Skaičiavimai aprašyti klasės metodų aprašymuose.

#### *Sąsaja/eksportas*

Metodai Išsaugoti().



## Klasės metodai

### Išsaugoti (string brėžinio duomenys)

Atsakomybės

Išsaugomas brėžinys pasirinktoje vietoje.

Skaičiavimai

Patikrinama ar galima išsaugoti brėžinį.

Sąsaja/eksportas

Brėžinio duomenys – sukurto brėžinio saugojimas tekstiniu formatu.

### 3.3.2. AutoCAD sistemos papildymas



18 pav. "AutoCAD sistemos papildymas" klasių diagrama

#### *Klasifikacija*

paketas

#### *Apibrėžimas*

Paketas atitinka pagrindinę sistemos dalį, jos funkcionalumą. Jame pateikiamos visos pagrindinės funkcijos, kurias gali atlikti sistema.

#### *Atsakomybės*

Sistemos papildymo funkcija - palengvinti vartotojo darbą su AutoCAD sistema.

#### *Apribojimai*

Turi būti užtikrintas komandų korektiškumas.

#### *Struktūra*

Komponentą sudaro klasės aprašytos pakete „sistemos papildymas“ ir pateiktos diagramoje.

#### *Sąveikavimas*

Sąveikauja su makrokomandomis.

### *Resursai*

Meniu saugomas AutoCAD pagrindiniame meniu(.mnu.).

### *Skaičiavimai*

Aprašyti komponentą sudarančių klasių metodų aprašymuose.

### *Sąsaja/eksportas*

Sąsaja menu.

#### *3.3.2.1. brėžinio braižymas*

### *Klasifikacija*

Klasė

### *Apibrėžimas*

Klasė skirta brėžinių braižymui.

### *Atsakomybės*

Pagrindinis sistemos komponentas, braižo įvairius brėžinius. Skirta pagerinti AutoCAD sistemos darbą.

### *Apribojimai*

Duomenys turi būti korektiški ir atitikti AutoCAD reikalavimus.

### *Struktūra*

Klasės struktūra pateikta klasių diagramoje.

### *Sąveikavimas*

Klasė naudojama vartotojo klasės.

### *Resursai*

Naudojama AutoCAD sistema.

### *Skaičiavimai*

Aprašyti komponentą sudarančių klasių metodų aprašymuose.

### *Sąsaja/eksportas*

Metodai pasirinkti(), sukurti(), šalinti(), redaguoti(), spausdinti(), išsaugoti().

### **Klasės metodai**

#### **Pasirinkti (string makrokomanda)**

##### *Atsakomybės*

Naujos makrokomandos pasirinkimas.

##### *Skaičiavimai*

Pasirenkama komanda brėžinio kūrimui.

Sąsaja/eksportas

Makrokomanda – vartotojo pasirinkta komanda.

### **Sukurti (int matmenys)**

Atsakomybės

Naujo brėžinio sukūrimas.

Skaičiavimai

Tikrinama ar duomenys tinkami, jei ne, tuomet išmetamas pranešimas, kad kažką reikia keisti.

Sąsaja/eksportas

Matmenys – vartotojo pateikiami brėžinio matmenys.

### **Šalinti (string brėžinys)**

Atsakomybės

Pašalinti brėžinį.

Skaičiavimai

Pašalinamas kuriamas brėžinys arba tam tikri brėžinio duomenys.

Sąsaja/eksportas

Brėžinys – vartotojo sukurtas brėžinys.

### **Redaguoti (string brėžinys)**

Atsakomybės

Skirtas sukurto brėžinio redagavimui, matmenų keitimui.

Skaičiavimai

Tikrinama ar naujai įvesti duomenys atitinka reikalavimus.

Sąsaja/eksportas

Brėžinys – tam tikras brėžinys.

### **Spausdinti (string brėžinys)**

Atsakomybės

Brėžinio spausdinimas spausdintuvu.

Skaičiavimai

Nubraižytas brėžinys spausdinamas pasirinktu spausdintuvu.

Sąsaja/eksportas

Brėžinys – vartotojo sukurtas brėžinys.

### **Išsaugoti (string failas)**

Atsakomybės

Brėžinio išsaugojimas faile.

## Skaičiavimai

Sukuriamas DWG failas, kuris išsaugomas nurodytu vardu. Tikrinama ar failas tokiu vardu jau yra, ar užtenka saugojimui vietos diske.

## Sąsaja/eksportas

Failas – koku vardu išsaugoti brėžinį, jei failas nurodytu vardu jau yra, išvesti pranešimą. Nurodoma kur išsaugoti brėžinį.

### 3.3.2.2. makrokomandos

#### *Klasifikacija*

Klasė

#### *Apibrėžimas*

Klasė aprašo kokiomis funkcijomis papildyta AutoCAD sistema.

#### *Atsakomybės*

Klasė atsakinga už įvairių funkcijų veikimą..

#### *Apribojimai*

Funkcijos turi būti korektiškos.

#### *Struktūra*

Struktūra pateikta klasių diagramoje.

#### *Sąveikavimas*

Klasė naudojama brėžinių braižymo klasės.

#### *Resursai*

Naudojama AutoLisp ir Visual Lisp kalbos.

#### *Skaičiavimai*

Skaičiavimai aprašyti klasės metodų aprašymuose.

#### *Sąsaja/eksportas*

Metodai pasirinkti().

### **Klasės metodai**

#### **Pasirinkti (string komanda)**

Atsakomybės

Funkcijos, kurią norima atlikti pasirinkimas.

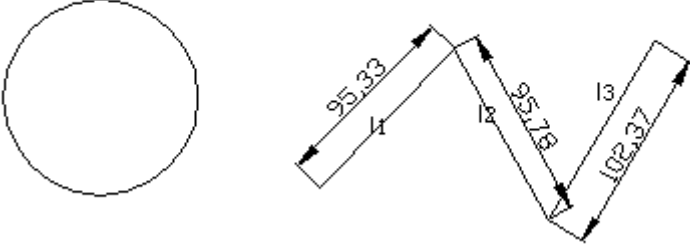
Skaičiavimai

Vykdoma pasirinkta funkcija, laukiama papildomų duomenų įvedimo.

Sąsaja/eksportas

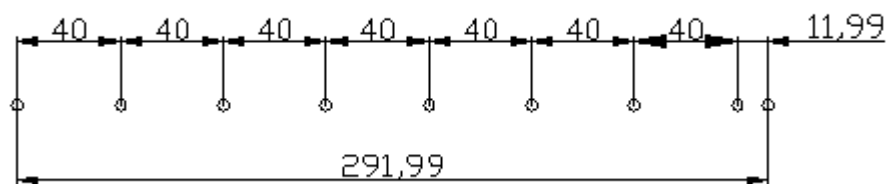
Funkcija – funkcijos vardas.

### Sukurtų funkcijų aprašas

Nr.	Pavadinimas	Argumentai	Funkcijos paskirtis
1	Lenght	Objektų sąrašas.  Rezultatas – objektų ilgių suma.	Skirta objektų ilgių sumai apskaičiuoti.  $F_{ob}(x) = s$ , čia $x$ pažymėtas objektas, $s$ - ilgis;  $F_a(x_1, x_2, \dots, x_n) = \sum_{i=1}^n s_i$ , $x_1, x_2, \dots, x_n$ - objektai, $s_i$ - i-ojo objekto ilgis;  $F_s(x_1, x_2, \dots, x_n) = \sum_{i=1}^n s - s_i$ , $x_1, x_2, \dots, x_n$ - objektai, $s_i$ - i-ojo objekto ilgis, $s$ - visas ilgis.
<p>Grafinė iliustracija</p> <div style="text-align: center;">  </div> <p style="text-align: center;">19 pav. Ilgių skaičiavimas</p> <p>Length funkcijos pagalba apskaičiuotas visų linijos segmentų bendras ilgis <math>l = l_1 + l_2 + l_3</math></p> <p>AutoCAD protokolas:</p> <pre> LENGTH [Object/Add/Subtract] :a Select objects: length 95.3288 Visas ilgis 95.3288 Select objects: length 95.7807 Visas ilgis 191.11 Select objects: length 102.372 Visas ilgis 293.481 Select objects: *Cancel*                     </pre>			
2	taskai	Du taškai.	Padalina atstumą tarp dviejų taškų į nurodytą intervalų skaičių arba į

		Rezultatas - taškais padalintas atstumas.	nurodytus atstumus, dalinimo vietose sudėdama taškus. $F(p_1, p_2, len, sk) =$ sudėti taškai, čia $p_1, p_2$ – nurodyti taškai; $len$ – intervalų ilgis; $sk$ – intervalų skaičius.
--	--	---	--

Grafinė iliustracija



20 pav. Sudeda taškus kai nurodomas segmentų ilgis

AutoCAD protokolas:

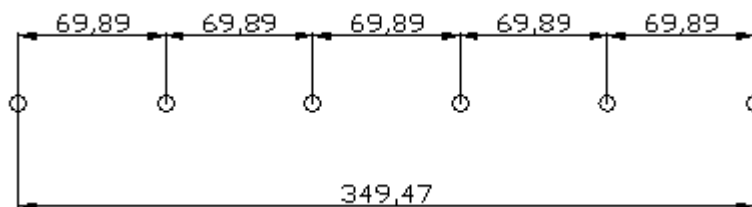
Command: taskai

Pick first point: Pick second point:

distance: 291.9929

length Yes No? y

Specify length of segments: 40



21 pav. Sudeda taškus, kai nurodomas segmentų skaičius

AutoCAD protokolas:

Command: taskai

Pick first point: Pick second point:

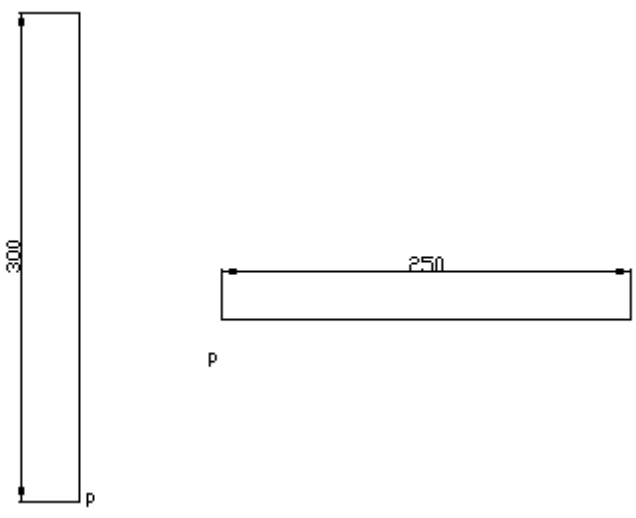
distance: 349.4677

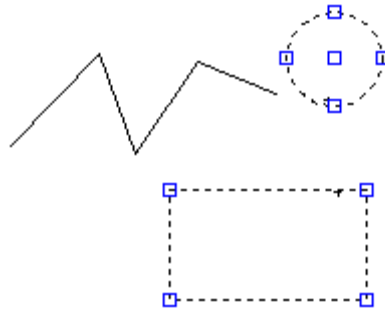
length Yes No? n

Enter number of segments: 5

Spacing is: 69.8935

3	Xline1	Linijos ilgis. Rezultatas – pasirinkto ilgio	Skirta braižyti konstrukcines linijas nurodant jų ilgį. $F(p, len) =$ konstrukcinė linija, $p$ –
---	--------	---	---

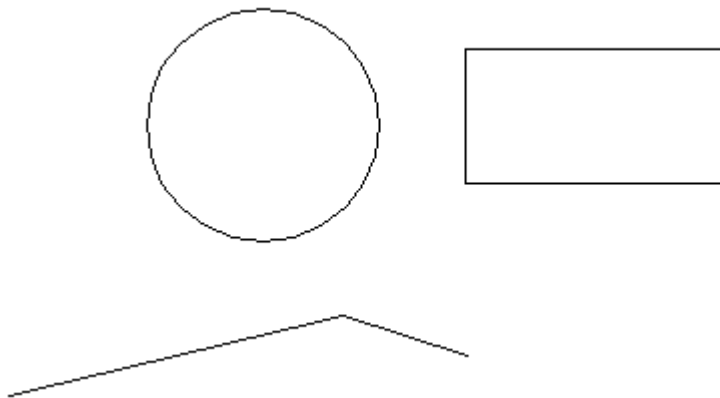
		konstrukcinė linija.	pradinis taškas, len – linijos ilgis.
Grafinė iliustracija			
			
22 pav. Konstrukcinės linijos			
AutoCAD protokolas:			
Command: ver			
Pick starting point for line			
Enter length:300			
line Specify first point:			
Specify next point or [Undo]:			
Specify next point or [Undo]:			
Command:			
Pick starting point for line *Cancel*			
Command: hor			
Pick starting point for line			
Enter length: 250			
line Specify first point:			
Specify next point or [Undo]:			
Specify next point or [Undo]:			
Command:			
Pick starting point for line *Cancel*			
4	lock	Objektų sąrašas. Rezultatas užrakintų objektų sąrašas.	Užrakina pasirinktus objektus. $F(x_1, x_2, \dots, x_n)$ = užrakinti objektai, $x_1, x_2, \dots, x_n$ – pažymėti objektai.
Grafinė iliustracija			



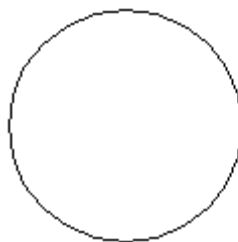
23 pav. Pažymėti objektai užrakinti

5	hid	Objektų sąrašas. Rezultatas paslėptų objektų sąrašas.	Paslepia pasirinktus objektus. $F(x_1, x_2, \dots, x_n) =$ paslėpti objektai, $x_1, x_2, \dots, x_n$ – pažymėti objektai.
---	-----	--	---

Grafinė iliustracija



24 pav. Prieš funkcijos hid panaudojimą

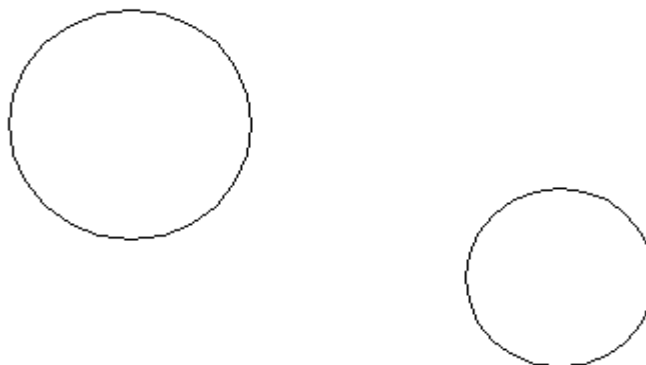


25 pav. Po funkcijos hid panaudojimo

6	trinti	Objektų sąrašas. Rezultatas – ištrinti objektai.	Skirta ištrinti pasikartojančius objektus. $F(x) =$ ištrintų objektų skaičius, $x$ -visas failas.
---	--------	---	--



Grafinė iliustracija



26 pav. Pasikartojantys objektai

AutoCAD protokolas:

Command: trintip

piesiniai trinami

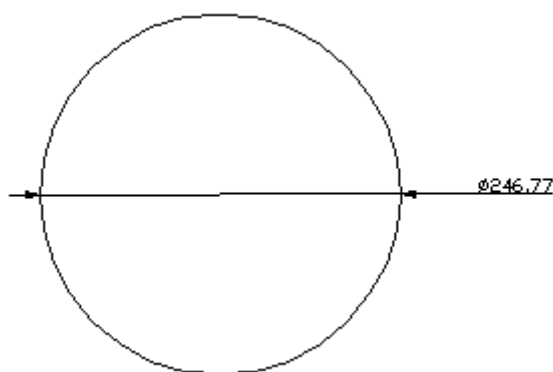
prasome palaukti

Objektu skaicius pries trynima 9

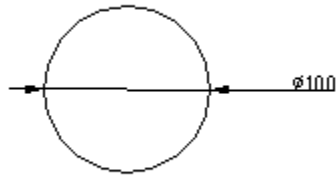
objektu skaicius po trynimo 5

7	editc	Apskritimas. Rezultatas – naujo diametro apskritimas.	Pakeičia pasirinkto apskritimo diametrą. Nurodžius naują diametrą reikia dar kartą pažymėti apskritimą. $F(x)$ = naujo diametro apskritimas, $x$ – apskritimas.
---	-------	--	---

Grafinė iliustracija



27 pav. Prieš funkcijos panaudojimą



28 pav. Po funkcijos panaudojimo

AutoCAD protokolas:

Command: editc

pasirinkite apskritima redagavimui

Diametras: 246.7681

Iveskite nauja diametra: 100

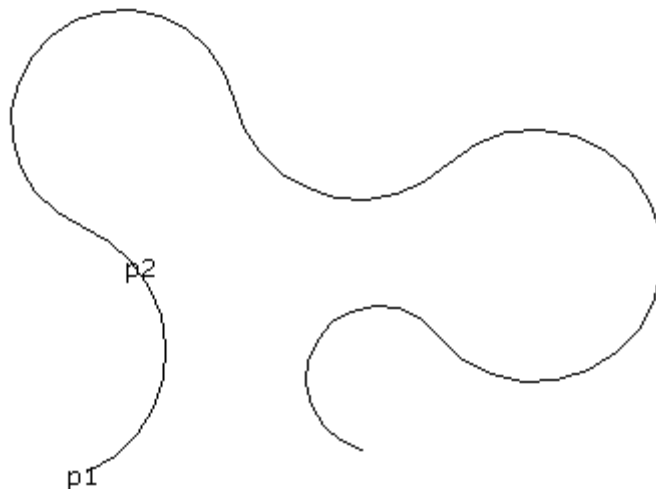
Select objects: 1 found

Select objects:

1 modified

8	arcline	Lanko pradžios, pabaigos taškai, kampas. Rezultatas – lankas.	Brėžia lankų liniją. $F(p1,p2, ang) = \text{lankai}$ , p1 – lanko pradžios taškas, p2 – lanko pabaigos taškas, ang – kampas.
---	---------	--	--

Grafinė iliustracija



29 pav. Lankų linija

P1 –pradžios taškas, p2 – vieno lanko pabaigos taškas ir naujo pradžios.

9	jpoints	Taškų sąrašas. Rezultatas – poliliniija.	Sujungia pažymėtus taškus į poliliniija. $F(p1,p2,...,pn)=\text{poliliniija}$ , p1,p2,...pn-taškai.
---	---------	---	--

Grafinė iliustracija



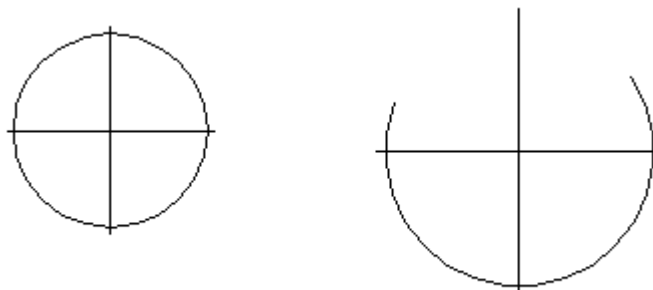
30 pav. Prieš funkcijos panaudojimą



31 pav. Po funkcijos panaudojimo

10	cenlines	Apkritimai ir lankai. Rezultatas – centro linija.	Nubraižo pasirinkto apskritimo ar lanko centro linijas. $F(x)$ = centro linijos.
----	----------	--	---


Grafinė iliustracija



32 pav. Centro linijos

11	txtbox	Objektų sąrašas. Rezultatas – aprėmintas tekstas.	Skirta pasirinkto teksto aprėminimui. $F(x_1, x_2, \dots, x_n)$ = aprėmintas tekstas, čia $x_1, x_2, \dots, x_n$ – pasirinkti objektai.
----	--------	--	--

Grafinė iliustracija

labas vakaras  


kaip sekasi

33 pav. Prieš funkcijos panaudojimą

labas vakaras

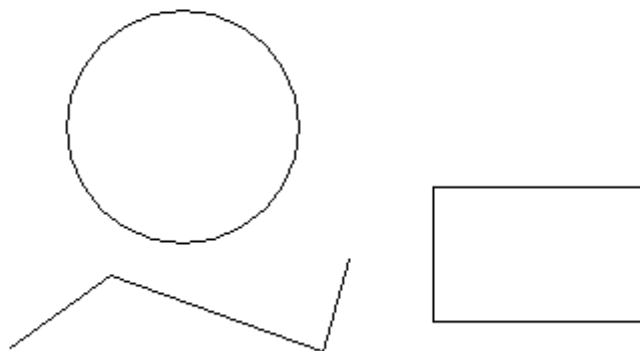


kaip sekasi

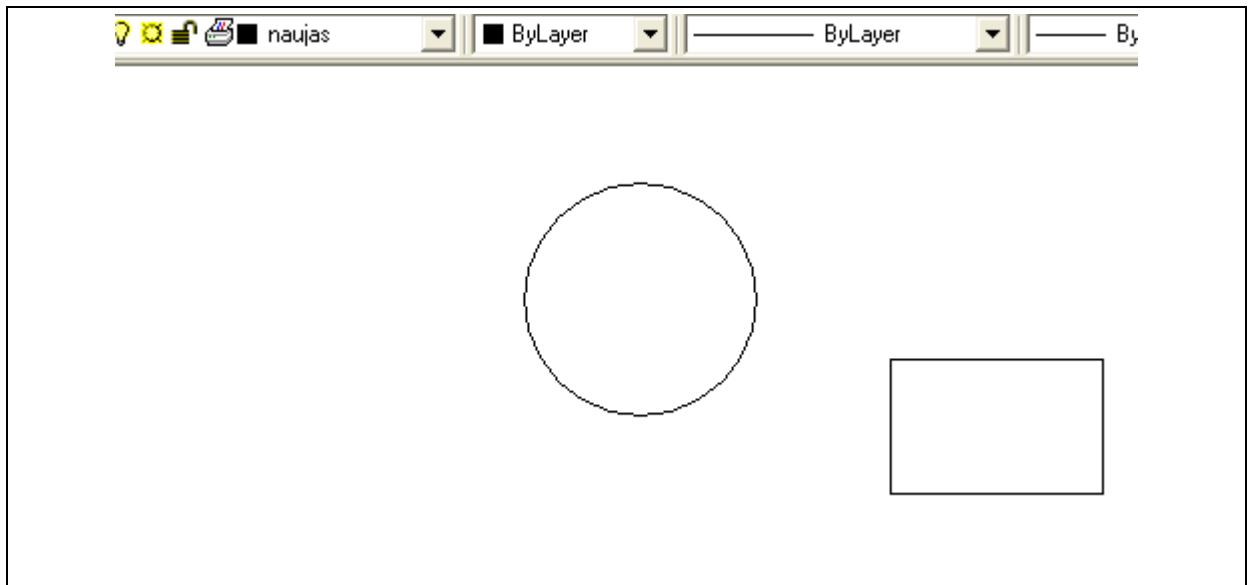
34 pav. Po funkcijos panaudojimo

12	ctl	<p>Objektų sąrašas.</p> <p>Rezultatas – nukopijuoti į pasirinktą sluoksnį objektai.</p>	<p>Kopijuoja pasirinktus objektus į pasirinktą sluoksnį.</p> <p><math>F(x_1, x_2, \dots, x_n) = cx_1, cx_2, \dots, cx_n, x_1, x_2, \dots, x_n</math> – pasirinkti objektai, <math>cx_1, cx_2, \dots, cx_n</math> – pasirinktų objektų kopijos kitame sluoksnyje.</p>
----	-----	---	--

Grafinė iliustracija



35 pav. Sluoksnio 0 vaizdas



36 pav. Sluoksnio naujas vaizdas po funkcijos panaudojimo

AutoCAD protokolas:

Command: ctl

Pasirinkite objektus kopijavimui

Select objects: 1 found

Select objects: 1 found, 2 total

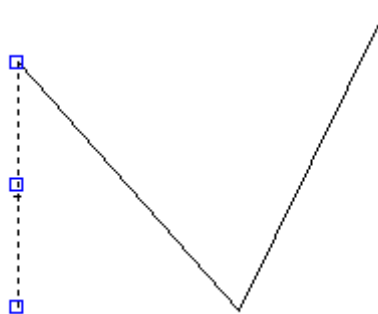
Select objects:

Pasirinkite sluoksnį į kurį kopijuosite objektus naujas

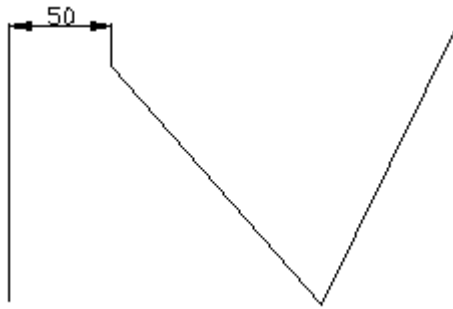
Command: '\_textscr

13	offset	Objektas. Lygiagrečiai perkopijuotas objektas.	Lygiagrečiai perkopijuoja ir ištrina originalą. $F(x)=\text{offset}(x)$ , x – pasirinktas objektas.
----	--------	---	--

Grafinė iliustracija



37 pav. Prieš funkcijos panaudojimą



38 pav. Po funkcijos panaudojimo

Pažymėta linija lygiagrečiai į kairę pusę perkopijuota atstumu 50.

14	poliliniija	Objektas(linija, lankas, poliliniija). Rezultatas – prijungta poliliniija.	Prijungia poliliniija prie lanko, linijos ar poliliniijos. $F(x)=nx$ , $x$ – pasirinktas objektas, $nx$ – objektas su prijungta poliliniija.
----	-------------	---	---

Grafinė iliustracija



39 pav. Prieš funkcijos panaudojimą



40 pav. Po funkcijos panaudojimo

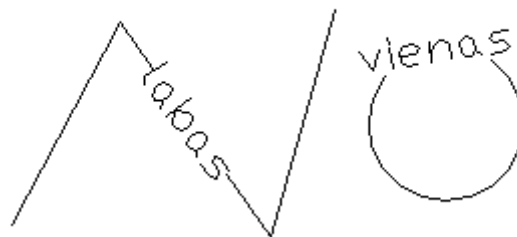
Poliliniija prijungiama nuo taškų p.

15	inserttxt	Objektas. Rezultatas – objektas su įterptu tekstu.	Įterpia tekstą į liniją. $F(x) = nx$ , $x$ – pasirinktas objektas, $nx$ – pasirinktas objektas su įterptu tekstu.
----	-----------	---	--

Grafinė iliustracija



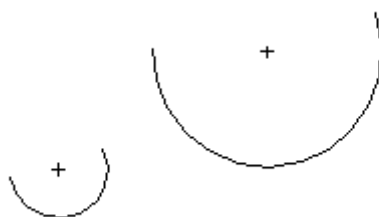
41 pav. Prieš funkcijos panaudojimą



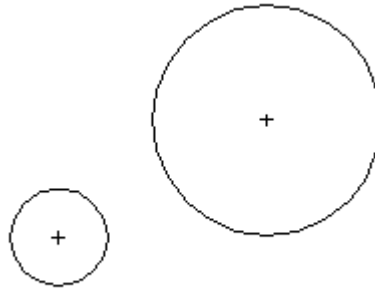
42 pav. Po funkcijos panaudojimo

16	tocircle	Objektas. Rezultatas – apskritimas.	Keičia lanką į apskritimą. $F(x) = \text{apskritimas}$ , $x$ – lankas.
----	----------	--	---

Grafinė iliustracija



43 pav. Prieš funkcijos panaudojimą



44 pav. Po funkcijos panaudojimo

17	keisti	Objektų sąrašas. Rezultatas – pakeistas tekstas.	Pakeičia pasirinktą tekstą kitu. $F(x_1, x_2, \dots, x_n) =$ pakeistas tekstas, $x_1, x_2, \dots, x_n$ – pasirinkti objektai.
----	--------	--	---

Grafinė iliustracija

šiandien graži diena

kaip sekasi

45 pav. Prieš funkcijos panaudojimą

šiandien saulėta diena

kaip sekasi

46 pav. Po funkcijos panaudojimo

AutoCAD protokolas:

Command: keistiteksta

Ieškomas tekstas: graži

Pakeisti tekstu: saulėta

Select objects: 1 found

Select objects: 1 found, 2 total

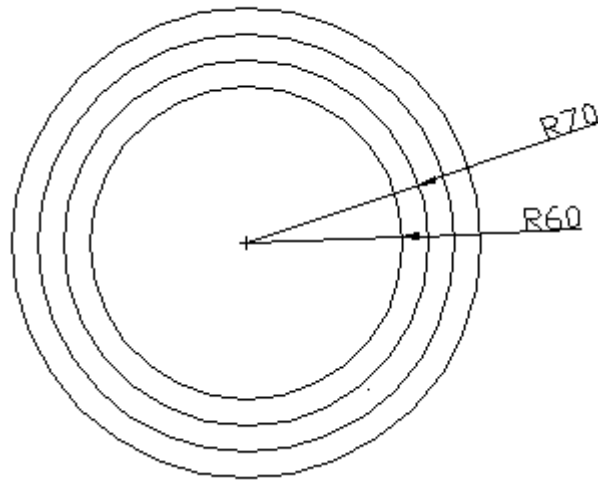
Select objects:

Pakeista 1 teksto objektas(i). Current text style: "Standard" Text height: 12

18	ccircle	Taškas, spindulys.	Braižo koncentrinus apskritimus. $F(p_1, r) =$ apskritimai, $p_1$ – apskritimo centro taškas, $r$ – apskritimo spindulys.
----	---------	--------------------	---



Grafinė iliustracija

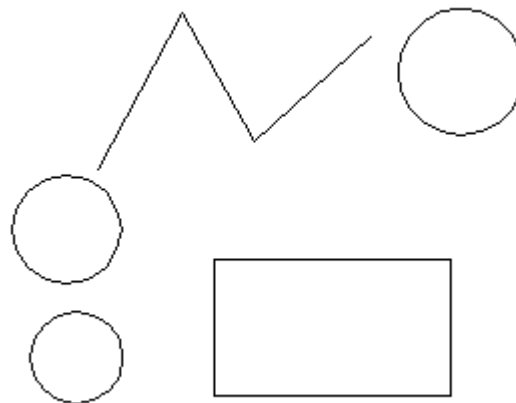


47 pav. Koncentriniai apskritimai

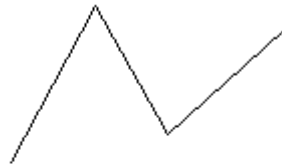
Nubraižyti 4 koncentriniai apskritimai, offset atstumu 10.

19	obdelete	<p>Objektų sąrašas, tipas.</p> <p>Rezultatas – ištrinti nurodyto tipo objektai.</p>	<p>Trina pasirinkto tipo objektus iš brėžinio.</p> <p><math>F(x_1, x_2, \dots, x_n, t)</math> = ištrinti t-tipo objektai, <math>x_1, x_2, \dots, x_n</math> – pasirinkti objektai, t – trinamų objektų tipas.</p>
----	----------	---	---

Grafinė iliustracija



48 pav. Prieš funkcijos panaudojimą



49 pav. Po funkcijos panaudojimo

Ištrinti apskritimai.

AutoCAD protokolas:

Command: obdelete

Pasirinkite objektus kuriuos norite ištrinti: \_erase

Select objects: 3 found

Select objects:

Command: 3 objektai ištrinti

Pasirinkite objektus trynimui:

### Naujų funkcijų atitikmenys AutoCAD'e

Darbe sukurta funkcija	AutoCAD funkcija	Darbe sukurtų funkcijų privalumai
arcline	arc	AutoCAD'e galima nubraižyti vieną lanką, darbe sukurta funkcija braižo kiek norime sujungtų lankų.
ccircle		Panašios funkcijos nėra.
cenlines	line	Centro linijas galima nubraižyti line pagalba.
ctl	copy	AutoCAD'e galima kopijuoti objektus į kitą vietą, darbe sukurta funkcija leidžia kopijuoti objektą į kitą sluoksnį.
editc	stretch	Darbe sukurtoje funkcijoje galima nurodyti diametro dydį.
hid	Off layer	AutoCAD'e galima paslėpti visą

		sluoksnį su jame esančiais objektais, darbe sukurta funkcija leidžia paslėpti pavienius objektus.
inserttxt		Panašios funkcijos nėra.
jpoints		Panašios funkcijos nėra.
keistiteksta	text	AutoCAD' galima pakeisti tekstą funkcijos text pagalba parenkant po vieną objektą, darbe sukurta funkcija leidžia pakeisti keletą teksto objektų iš karto juos visus pažymėjus.
lenght	distance	AutoCAD'o funkcija skaičiuoja atstumą tarp dviejų taškų, darbe sukurta funkcija gali apskaičiuoti vieno ar kelių objektų ilgių sumą, pridėti pasirinkto objekto ilgį prie sąrašo arba atimti jį.
lock	Lock layer	AutoCAD'e galima užrakinti visą sluoksnį, darbe sukurta funkcija leidžia pasirinkti pavienius objektus, nebūtina užrakinti viso sluoksnio.
obdelete		AutoCAD'e galima ištrinti visus pažymėtus objektus, darbe sukurtos funkcijos pagalba galima ištrinti visus norimo tipo objektus.
ofsetd	offset	AutoCAD'e lygiagrečiai kopijuoti
poliliniija	polyline	AutoCAD'e galima nurodyti pabaigos tašką prie kurio jungti, darbe sukurta funkcija automatiškai nustato pabaigos tašką pažymėjus objektą netoli pabaigos.
taskai	Divide, measure	AutoCAD'o funkcijos taškais

		padalina objektus, o darbe sukurta funkcija padalina atstumą tarp taškų.
tocircle		Panašios funkcijos nėra.
trintip		AutoCAD'e tokios funkcijos nėra.
txtbox		Panašios funkcijos nėra.
Xline1	xline	AutoCAD'o funkcija brėžia neriboto ilgio linijas, darbe sukurta funkcija leidžia pasirinkti norimą ilgį.

## Išvados

1. Sukurtos funkcijos skirtos vartotojų galimybėms padidinti.
2. Sistemos papildymas palengvina ir papildo darbą su AutoCAD sistema.
3. AutoCAD sistema papildoma 19 naujų funkcijų.
4. Naujomis funkcijomis naudotis nesudėtinga, jos aprašytos naudojant standartinį AutoCAD'o funkcijų aprašymo stilių.

## 4. Vartotojo dokumentacija

### 4.1. Sistemos funkcinis aprašymas

AutoCAD vartotojai pasigenda daugumos funkcijų. Kuriama sistema paprasta, pašalina kai kuriuos vartotojų nepatogumus, palengvindama jų darbą.

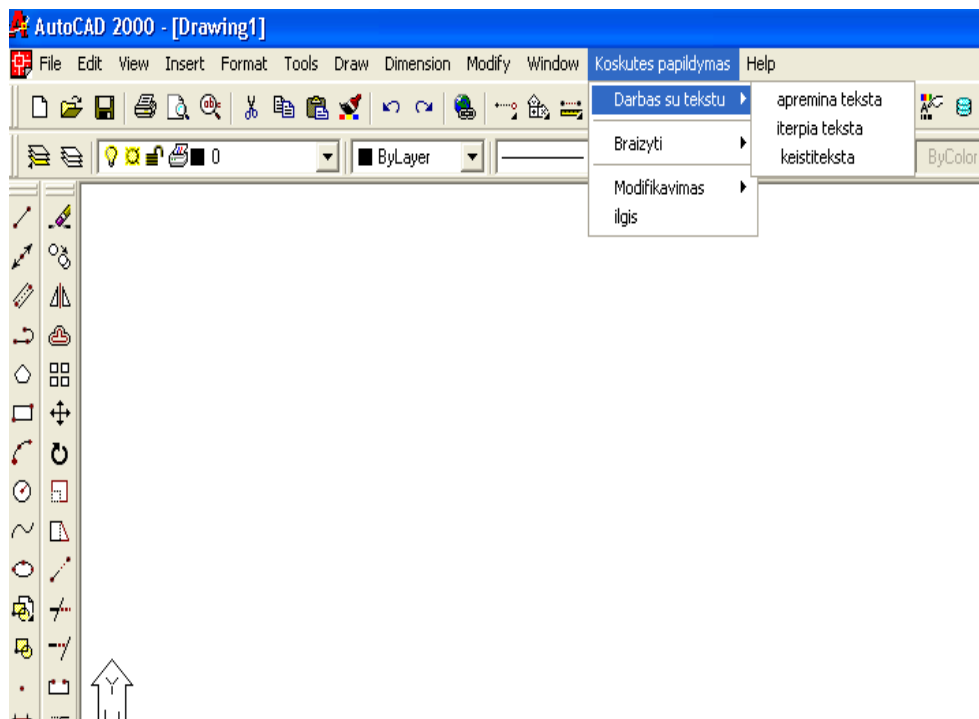
AutoCAD meniu papildomas šiomis naujomis funkcijomis:

- Objektų ilgių skaičiavimas (Length)
- Tarp dviejų taškų sudeda taškus (Taskai)
- Norimo ilgio xline (Xline1)
- Objektų užrakinimas (Lock)
- Objektų paslėpimas (Hid)
- Trina\_pasikartojančius (trintip)
- Keičia apskritimų dydį (editc)
- Lankų juosta (arcline)
- Taškų jungimas į poliliniją (jpoints)
- Apskritimo ar lanko centro linijos (cenlines)
- Aprėmina tekstą (txtbox)

- Kopijuoja objektus į sluoksnį (ctl)
- Offset su originalo ištrynimu (ofsetd)
- Prijungia polilinią (polilinija)
- Įterpia tekstą (inserttxt)
- Lanką pakeičia į apskritimą (tocircle)
- Pakeičia tekstą (keistiteksta)
- Koncentriniai apskritimai (ccircle)
- Trina pasirinkto tipo objektus (obdelete)

## 4.2. Sistemos vadovas

### Menu vaizdas



Norint vykdyti komandą reikia iš meniu punkto ją pasirinkti. Funkcijų veiksmams aprašyti projektinėje dalyje.

## 4.3. Sistemos instaliavimas

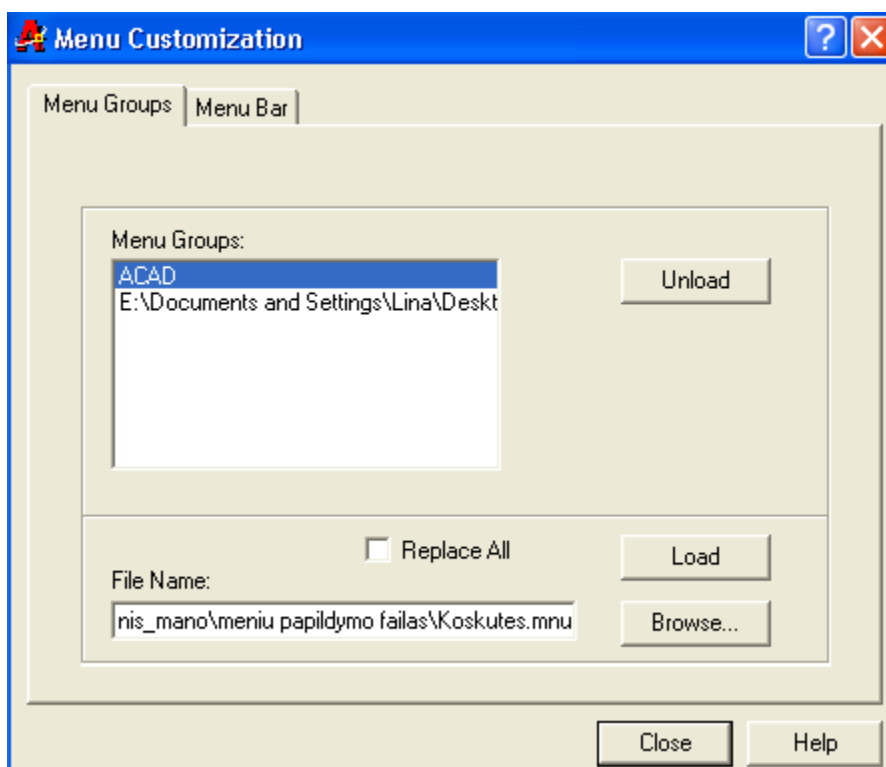
Meniu papildymas saugomas CD, jį reikia įsikelti į savo AutoCAD sistemą. Failus arcline.lsp, ccircle.lsp, ctl.lsp, editc.lsp, hid.lsp, cenlines.lsp, inserttxt.lsp, jpoints.lsp, keistiteksta.lsp, length.lsp, lock.lsp, obdelete.lsp, ofsetd.lsp, polilinija.lsp, taskai.lsp, tocircle.lsp, trintip.lsp, txtbox.lsp, xline1.lsp reikia persikopijuoti į AutoCAD'o SUPPORT katalogą. Meniu papildymo failą Koskutes.mnu įsikelti į darbinį AutoCAD meniu:

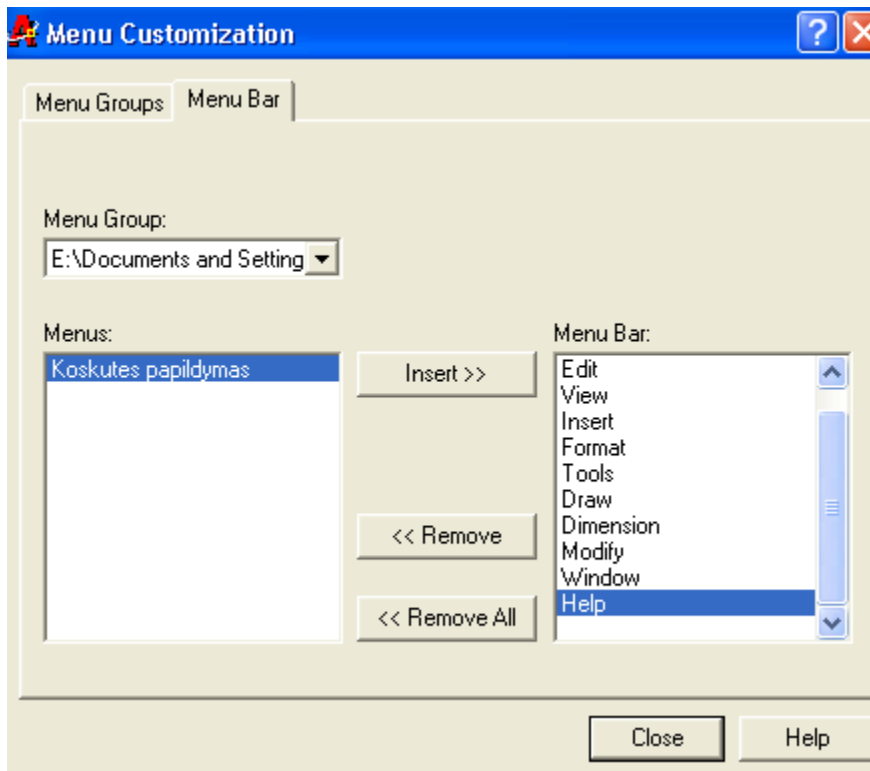
1. Komanda **MENU** įkelti darbinį meniu acad.mnu, jei prieš tai buvo įkeltas kitas meniu

2. **Tools** → **Customize menus...**

3. Atverto dialogo lango **Menu Customization** skyriuje **Menu Groups** paspaudus komandinį mygtuką **Browse...** atvertame lange **Select Menu File** pažymėti failą ...\\Koskutes.mnu, paskui **Open**
4. Dialogo lange **Menu Customization** paspausti mygtuką **Load**
5. Suaktyvinti skyrių **Menu Bar**, kurio sąrašė menu bar pateiktas visų įkeltų skyrių sąrašas.
6. Išskleidžiamame sąrašė **Menu Group** parinkti grupę ...\\Koskutes.mnu, o sąrašė **Menu Bar** pažymėti skyrių prieš kuri norite įterpti papildymą
7. Sąrašė **Menus** pažymėti **Koskutes papildymas** ir paspausti mygtuką **Insert**

Papildymo meniu skyrius panaikinti galima modifikuojant darbinio meniu failo tekstą, arba meniu komanda **MENUUNLOAD** (Tools →Customize Menus...).





## 5. Eksperimentinis tyrimas

Kiekviena funkcija testuota atskirai. Testavimui naudota AutoCAD sistema 2004. Stebėtas funkcijų veikimas, jų korektiškumas. Pastebėtos klaidos ištaisytos. Darbe sukurtos funkcijos turi daugiau galimybių, patogesnės naudoti. Funkcijos suskirstytos į keletą grupių: darbui su tekstu (txtbox, inserttxt, keistiteksta), skirtos braižyti (cenlines, taskai, xline1, polilinja, arcline, ccircle), modifikavimui (jpoints, editc, tocircle, ctl, lock, hid, trintip, obdelete, ofsetd), ilgio apskaičiavimams (length).

## 6. Išvados

1. AutoCAD sistema turi vienas iš geriausių modifikavimo priemonių. Tai viena iš svarbiausių šios sistemos pasisekimo priežasčių, nes daugelis panašių grafinių sistemų savomis funkcijomis nepapildomos.
2. AutoCAD sistema nuo jos sukūrimo pradžios mažai papildoma naujomis funkcijomis, nors vartotojų poreikiai išaugo. Todėl tikslinga papildyti sistemą aktualiomis funkcijomis.
3. Atlikta vartotojų poreikų analizė leido išsiaiškinti trūkstamų funkcijų pobūdį.
4. Sukurtos funkcijos (19 funkcijų) skirtos vartotojų galimybės padidinti. Naujos funkcijos palengvina darbą su AutoCAD sistema.

5. Naujomis funkcijomis naudotis nesudėtinga, jos aprašytos naudojant standartinį AutoCAD funkcijų aprašymo stilių.
6. Naujos funkcijos sukurtos AutoLisp ir VisualLisp programavimo kalbomis.
7. Sukurtą sistemos papildymą galima papildyti naujomis funkcijomis, nenaudojamas funkcijas pašalinti, pakeisti naujomis.



## 7. Literatūros sąrašas

1. Deke McClelland *Photoshop 5 žaliems*. 1999
2. Dundulis, R. *Kompiuterinis projektavimas I*. 2005
3. Gaidys, R. *Kompiuterinė grafika Cadkey*. 1992
4. *Kompiuterinė braižyba. Automatizuoto projektavimo sistema. Cadkey struktūra*. 1994
5. Lenkevičius, A.; Matickas, J. *Kompiuterinė grafika*. 2007
6. Lūžienė, T. *Kompiuterinė grafika*. 2006
7. Сергей Зуев Николай Полещук *САПР на базе AutoCAD - как это делается*. 2004
8. Sokas. *Programavimas VBA kalba. Mokomoji knyga*. Vilnius, 2005.
9. Steve Brain *CorelDraw 12 Oficialus vadovas*. 2004.
10. [www.cadkey.com](http://www.cadkey.com)
11. <http://eif.viko.lt/uimg/file/Malma/ACAD01.pdf> žr. 2007-03-12
12. [http://www.ik.ku.lt/lessons/diplom/zilyte-diplom/acd\\_7.htm](http://www.ik.ku.lt/lessons/diplom/zilyte-diplom/acd_7.htm) žr 2006-01-26
13. [www.kc.gf.vu.lt/Paskaitos/MK/3-Photoshop-1.ppt](http://www.kc.gf.vu.lt/Paskaitos/MK/3-Photoshop-1.ppt) žr 2008-03-08
14. <http://www.softkey.lt/catalog/program.php?ID=21316&progdsc=long>
15. [www.solidworks.com](http://www.solidworks.com)

## 8. Terminų ir santrupų žodynas

.mnu – AutoCAD meniu failas.

AutoLisp – funkcinė dirbtinio intelekto programavimo kalba.

DXF – tekstinis brėžinio failo formatas.

Grafinis primityvas – taškas, tiesė, apskritimas ir t.t.

Grafinis redaktorius – redaktorius apdorojantis grafinius primityvus.

UML – unifikuota modeliavimo kalba.

VBA – Visual Basic for Applications, programavimo kalba skirta pradedantiesiems.

VisualLISP – programavimo kalba.

XML – kalba skirta paskirstyti informaciją internete.

## 9. Priedai

### Programinis kodas

#### 1. Funkcija ilgių apskaičiavimui (length)

```
(defun o-length () ;funkcija pavienių objektų ilgiui apskaičiuoti
(vl-load-com)
  (setq ssent (ssget „:S“))
  (setq count 0)
  (setq x_object (vlax-ENAME->VLA-Object (ssname ssent count)))
  (setq x_length (vlax-curve-getdistatparam
                  x_object
                  (vlax-curve-getendparam x_object)
                  ) ; apskaičiuoja objekto ilgį

  ) ; setq pabaiga
  (setq ilgis x_length)
  (setq k 0)
  (princ „\nlength „)
  (princ ilgis)
  (princ)
)
(defun a-length () ; funkcija prie jau turimo ilgio prideda pasirinkto objekto ilgį
(vl-load-com)
  (if (= k 0)
    (setq suma ilgis)
    (if (= k 2)
      (setq suma suma1)
      (setq suma 0)) )
  (while (/= ssent null)
    (setq ssent (ssget „:S“))
    (setq count 0)
    (setq x_object (vlax-ENAME->VLA-Object (ssname ssent count)))
    (setq x_length (vlax-curve-getdistatparam
                    x_object
                    (vlax-curve-getendparam x_object)
                    ) ; of vlax-curve-getdistatparam pabaiga

    ) ; setq pabaiga
    (setq ilgis1 x_length)
    (setq suma (+ suma ilgis1 )
    ) ; setq pabaiga
  (setq k 1)
  (princ „\nlength „)
  (princ ilgis1)
  (princ „ „)
  (princ „Visas ilgis“)
  (princ „ „)
  (princ suma)
  (princ))
)
(defun s-length () ; funkcija iš turimo ilgio atima pasirinkto objekto ilgį
(vl-load-com)
  (if (= k 0)
    (setq suma1 ilgis)
```

```

( if (= k 1)
  (setq suma1 suma)
  (setq suma1 0)) )
(while (/= ssent null)
  (setq ssent (ssget „:S“))
  (setq count 0)
  (setq x_object (vlax-ENAME->VLA-Object (ssname ssent count)))
  (setq x_length (vlax-curve-getdistatparam
                  x_object
                  (vlax-curve-getendparam x_object)
                  ) ;vlax-curve-getdistatparam pabaiga
  ) ; setq pabaiga
  (setq ilgis2 x_length)
  (setq suma1 (- suma1 ilgis2))
  )
(setq k 2)
  (princ „\nlength „)
(princ ilgis2)
(princ „ „)
(princ „visas ilgis“)
(princ „ „)
(princ suma1)
(princ))
)
(defun C:length () ; pagrindine programa
  (initget 0 „Object Add Subtract“)
  (setq xx (getkword „\n [Object/Add/Subtract] :“))
  (cond ((or (eq xx „Object“)
            (eq xx nil))
        (o-length))
        ((or ( eq xx „Add“)
            (eq xx nil))
        (a-length))
        (T
        (s-length))
        )
  )
)
(princ „\nType \“length\“ to start“)
(princ)

```

## 2. Funkcija sudedanti taškus tarp dviejų taškų (taskai)

```

(defun c:taskai ()
  (setvar „cmdecho“ 0)
  (setq P1 (getpoint „Pick first point: „)
  P2 (getpoint P1 „Pick second point: „))
  (setq d (distance P1 P2)) ;nustato atstuma tarp tasku
  (prompt (strcat „\ndistance: „ (rtos d))) atspausdina atstuma
  (initget 1 „Yes No“)
  (setq wr (getkword „\nlength Yes No? „))
  (if (= wr „Yes“)
    (progn

```

```

    (setq len (getdist „\nSpecify length of segments: „)
      sk1(/ d len)
      sk2( fix sk1)
      sk3(* len sk2))
    (if (/= d sk3)(setq sk (fix(+ 1 sk2) ))) )
    (if (= wr „No“)
      (progn
        (setq sk (getint „\nEnter number of segments: „)
          len (/ d sk)); apskaiciuoja koks gali buti segmentu ilgis
        (prompt (strcat „\nSpacing is: „(rtos len)“\n“)))
        (setq ANG (angle P1 P2))
        (setvar „blipmode“ 0)
        (setvar „snapang“ ANG)
        (command „POINT“ (polar P1 ANG len)
          „ARRAY“ (polar P1 ANG len) „ „R“ „ (- sk 1) len)
        (setvar „blipmode“ 1)
        (setvar „snapang“ 0)
      )
    )

```

### 3. Funkcija braižanti norimo ilgio konstrukcines linijas (ver,hor)

```

(defun C:VER ()
  ;(START)
  (setvar „osmode“ 32 )
  (setvar „orthomode“ 1) ;ijungia Ortho režimą
  (setq D „T“)
  (while D
    (setq P1 (getpoint „\nPick starting point for line „)); linijos pradžios taškas
    (setq X (car P1))
    ( setq len (getdist „\nEnter length:“)); xline ilgis
    (setq p2(cadr p1))
    (setq y (+ p2 len))
    (setvar „osmode“ 0 )
    (command „line“ P1 (list X y) „“)
    (setvar „osmode“ 32 )
  ) ; end of while
  (princ)
  (end)
) ; end of function
(defun C:HOR ()
  ; (START)
  (setvar „osmode“ 32 )
  (setvar „orthomode“ 1) ;ijungia Ortho režimą
  (setq D „T“)
  (while D
    (setq P1 (getpoint „\nPick starting point for line „))
    (setq Y (cadr P1))
    ( setq len (getdist „\nEnter length: „))
    (setq p2(car p1))
    (setq x(+ p2 len))
    (setvar „osmode“ 0 )
    (command „line“ P1 (list x Y) „“)
    (setvar „osmode“ 32 )
  ) ; end of while

```

```

(princ)
(end)
) ; end of function
(defun *error* (msg)
  (if (/= MSG „Function cancelled“)
    (princ (strcat „\nAutoLISP Error: „, MSG))
    (terpri))
  )
)
(DEFUN)
(princ)
)
(defun START ()
  (setq OLDERR *error*)
  (setq S_CMD (getvar „cmdecho“))
  (setvar „cmdecho“ 0)
  (setq S_OSM (getvar „osmode“))
  (setvar „osmode“ 0) ; NONE
  (setq S_Blip (getvar „blipmode“))
  (setvar „blipmode“ 0)
  (setq S_ORTH (getvar „orthomode“))
  (setvar „orthomode“ 0) ;išjungia ortho režimą
  (graphscr)
  (setq CL (getvar „clayer“))
)
(defun END ()
  (setvar „blipmode“ S_Blip)
  (setvar „orthomode“ S_ORTH)
  (setvar „osmode“ S_OSM)
  (command „layer“ „s“ CL „“)
  (setq *error* OLDERR)
  (setvar „cmdecho“ S_CMD)
  (prompt „\nType: VER for vertical construction lines.“)
  (prompt „\nType: HOR for vertical construction lines.“)
  (princ)
)
(prompt „\nType: VER for vertical construction lines.“)
(prompt „\nType: HOR for vertical construction lines.“)
(princ);

```

#### 4. Funkcija užrakinanti objektus (lock)

```

(defun c:LOCK ()
  (setvar „cmdecho“ 0)
  (setq appl „LOCKED“)
  (if (not (tblsearch „appid“ appl)) (regapp appl) )
  (initget „Lock Unlock“)
  (setq wr (getkword „\nLock, Unlock : „))
  (cond ( (= wr „Unlock“) (RESTORE-LAYR „Locked“) )
    ( (= wr „Lock“) (CHANGE-LAYR „Locked“) )
  )
)
(princ)
);defun

```

(defun CHANGE-LAYR (layr-name); naujo sluoksnio sukurimui i kuri patalpinami uzrakinti objektai

```

(setq ss (ssget))
(if ss (progn
(command „_layer“ „_n“ layr-name (strcat „_“ wr) layr-name „“)
(setq count 0)
(while (setq s1 (ssname ss count))
  (setq dxf (entget s1))
  (setq old-name (cdr (assoc 8 dxf)))
  (setq xd (list (LIST -3 (LIST appl
                        (cons 1002 „{,„)
                        (cons 1003 old-name)
                        (cons 1002 „,}“))
                )))
  (setq dxf (subst (cons 8 layr-name) (assoc 8 dxf) dxf))
  (setq dxf (append dxf xd))
  (entmod dxf)
  (setq count (+ 1 count))
)
));if
);defun
(defun RESTORE-LAYR (layr-name)
(setq ss (ssget „_x“ (list (cons 8 layr-name)) ))
(if ss (progn
  (setq count 0)
  (if (= layr-name „Locked“) (command „_layer“ „_Unlock“ layr-name „“) )
  (while (setq s1 (ssname ss count))
    (setq dxf (entget s1 (list appl)))
    (setq xd (car (cdr (assoc -3 dxf))))
    (setq old-name (cdr (nth 2 xd)))
    (setq dxf (subst (cons 8 old-name) (assoc 8 dxf) dxf))
    (setq dxf (reverse (cdr (reverse dxf))))
    (entmod dxf)
    (setq count (+ 1 count))
  )
  (if (= layr-name „Locked“) (command „_layer“ „_Lock“ layr-name „“) )
))
(princ count) (princ „ elements“) (princ)
);defun

```

### 5. Funkcija paslepianči objektus(hid)

```

(defun c:HID ()
(setvar „cmdecho“ 0)
(setq appl „HID“)
(if (not (tblsearch „appid“ appl)) (regapp appl) )
(initget „ON OFF“)
(setq wr (getkword „\nON OFF „))
(cond ( (= wr „ON“) (RESTORE-LAYR „HID“) )
      ( (= wr „OFF“) (CHANGE-LAYR „HID“) ) ; paslepiamas objektas ikeliamas i nauja
sluoksni
)
(princ)
);defun
(defun CHANGE-LAYR (layr-name)
(setq ss (ssget))

```

```

(if ss (progn
(command „_layer“ „_n“ layr-name (strcat „_“ wr) layr-name „“)
(setq count 0)
(while (setq s1 (ssname ss count))
  (setq dxf (entget s1))
  (setq old-name (cdr (assoc 8 dxf)))
  (setq xd (list (LIST -3 (LIST appl
                        (cons 1002 „{,„)
                        (cons 1003 old-name)
                        (cons 1002 „}“)
                      )))
  (setq dxf (subst (cons 8 layr-name) (assoc 8 dxf) dxf))
  (setq dxf (append dxf xd))
  (entmod dxf)
  (setq count (+ 1 count))
)
));if
);defun

```

```

(defun RESTORE-LAYR (layr-name)
(setq ss (ssget „_x“ (list (cons 8 layr-name)) ))
(if ss (progn
  (setq count 0)
  (while (setq s1 (ssname ss count))
    (setq dxf (entget s1 (list appl)))
    (setq xd (car (cdr (assoc -3 dxf))))
    (setq old-name (cdr (nth 2 xd)))
    (setq dxf (subst (cons 8 old-name) (assoc 8 dxf) dxf))
    (setq dxf (reverse (cdr (reverse dxf))))
    (entmod dxf)
    (setq count (+ 1 count))
  )
)
)
(princ count) (princ „ elements“) (princ)
);defun

```

## 6. Funkcija trinanti pasikartojančius objektus(trinti)

```

(defun DXF (A) (cdr (assoc A (entget ONE))))

(defun COMPLEX ()
  (while (not (wcmatch (DXF 0) „*END*“))
    (setq ONE (entnext ONE))
    (ELEM (append ELEM (list ONE)))
    (CHECK))
  )
)
(defun CORECT (A)
  (cond ((= (type (cdr A)) ,list)
    (cons (car A) (mapcar ,(lambda (X) (atof (rtos X 2 8))) (cdr A))))
    ((member (type (cdr A)) (list ,INT ,REAL))
    (cons (car A) (mapcar ,(lambda (X) (atof (rtos X 2 8))) (list (cdr A))))
    (t A)))
)
(defun CHECK ()
  (foreach X (entget ONE)

```



```

(if (not (member (car X) (-2 -1 5 6 8 62 100)))
  (setq TEMP (cons (CORECT X) TEMP))))
(defun C:trinti ()
  (setq CM (getvar „cmdecho“)
        LISTA NIL
        NEW NIL
        ONE NIL
        TEMP NIL
        OLD (ssget „_X“))
  (setvar „cmdecho“ 0)
  (cond
    (OLD
     (command „_-layer“ „_u“ „*“ „“)
     (prompt „\npiesiniai trinami „)
     (prompt „\nprasome palaukti \n“)
     (while (cond (ONE (setq ONE (entnext ONE)))
                  (t (setq ONE (entnext))))
            (setq ELEM (append ELEM (list ONE)))
            (CHECK)
            (if (or (and (= (DXF 0) „INSERT“) (= (DXF 66) ,1)) (= (DXF 0) „POLYLINE“)
                    (COMPLEX))
                (if (member TEMP LISTA)
                    (foreach X ELEM (entdel X)))
                (setq LISTA (cons TEMP LISTA)
                      TEMP NIL
                      ELEM NIL)
                (cond (ZNACZNIK (setq ZNACZNIK NIL) (princ „\r\“))
                      (t (setq ZNACZNIK (princ „\r/“)))) ;end while
                (prompt (strcat „\nObjektu skaicius pries trynima „ (itoa (sslength OLD))))
                (prompt
                 (strcat „\nobjektu skaicius po trynimo „ (itoa (sslength (ssget „_X“)))))
                (t (prompt „\nnerasta pasikartojanciu elementu „)))
            (setvar „cmdecho“ CM)
            (princ))
     (prompt „\nLoaded new command trinti „)
     (princ)

```

## 7. Funkcija redaguojanti apskritimų dydį (editc)

```

(defun C:editc ()
  (defun editc_err (S)
    (if (/= S „Function cancelled“)
        (princ (strcat „Error: „ S)) )
    (setq *error* OLD_ERR)
    (setvar „CMDECHO“ OLD_CMD)
    (princ)
  )
  (setq OLD_ERR *error*
        OLD_CMD (getvar „CMDECHO“)
        *error* EDITc_ERR
        ENT nil
  )
  (setvar „CMDECHO“ 0)
  (command „.undo“ „mark“)

```



```

(setq pabaiga2 (* 0.03125 (getvar „LTSCALE“)))
(command
  „pline“ pradzia „w“ „0.0“ pabaiga2 pabaiga1 „w“ „0.0“ „0.0“ „arc“
)
(princ)
)

```

### 9. Funkcija sujungianti taškus į poliliniją(jpoints)

```

(defun C:jpoints ()
  (prompt „\npasirinkite taskus sujungimui“)
  (setq SSET
    (ssget ,((0 . „POINT“)))
    length (sslenght SSET) ;nustato sarašo sset ilgi
    STPT (getpoint
      „\nspragtelekite salia jungiamos linijos“
    )
  )
)
(prompt „\njungia taškus“)
(command „PLINE“)
(repeat
  length
  (setq
    COUNT 0
    ENTNAME1 (ssname SSET 0)
    ENDPT1 (cdr (assoc 10 (entget ENTNAME1)))
    DIST1 (abs (distance STPT ENDPT1))
    ENTNAME2 (ssname SSET COUNT)
  )
  (while ENTNAME2
    (setq
      ENDPT2 (cdr (assoc 10 (entget ENTNAME2)))
      DIST2 (abs (distance STPT ENDPT2))
    )
    (if (< DIST2 DIST1)
      (progn
        (setq
          DIST1 DIST2
          ENDPT1 ENDPT2
          ENTNAME1 ENTNAME2
        )
      )
    )
  )
  (setq
    COUNT (+ COUNT 1)
    ENTNAME2 (ssname SSET COUNT)
  )
)
(command ENDPT1)
(ssdel ENTNAME1 SSET)
(setq STPT ENDPT1)
)
(command „“)
(prompt „\nsujungimas pabaigtas.....“)

```

)

### 10. Funkcija braižo apskritimo centro linijas(cenlines)

```
(defun C:CENLINES ()
  (prompt „Select circle or arc! „)
  (terpri)
  (command „select“ „si“ pause)
  (setq PT (ssget „P“))
  (setq PT (ssname PT 0))
  (setq PT (entget PT))
  (setq R (assoc 40 PT))
  (setq PT (cdr(assoc 10 PT)))
  (setq R (cdr R))
  ;set primary centerline overlap ratio
  (setq R (* 2 R 1.25))
  (cond
    ((< R 0.9)
     (setq LN „center2“ LFAC 1.5))
    ((< R 7.0)
     (setq LN „center“ LFAC 1.0))
    ((< R 500.0)
     (setq LN „centerx2“ LFAC 0.85))
  )
  (setq R (* R LFAC))
  (command „line“ PT (polar PT 0 R))
  (command)
  (setq PT2 (polar PT 0 (/ R 2)))
  (command „move“ „l“ „“ PT2 (polar PT2 pi (/ R 2)))
  (command „change“ „l“ „“ „p“ „lt“ LN „“)
  (command „array“ „l“ „“ „p“ PT 2 90 „“)
); end
```

### 11. Funkcija aprėminanti tekstą(txtbox)

```
(defun c:txtbox ()
  (vl-load-com)
  (defun *ERROR* (err)
    (princ)
  )
  (setvar „cmdecho“ 0)
  (setq ent „Something“)
  (while ent
    (while(not(or(= EnTyp „ATTRIB“)(= EnTyp „ATTDEF“)(= EnTyp „TEXT“)(= EnTyp „MTEXT“)))
      (if(not(= EnTyp nil))(prompt „\nPasirinktas ne atributas ar tekstas“))
      (if(setq ent (entsel „\nPasirinkite atributus ar teksta: „))
        (setq EnTyp (cdr (assoc 0 (entget (car ent)))) ;nustato objekto tipą
              EnPt (cadr ent) ;objekto taško koordinatės
              ent (car ent) ; Objekto vardas
              SubE (car (nentselp „“ EnPt))
              SenTyp (cdr (assoc 0 (entget SubE))))
        ); setq
      (setq ent nil EnTyp nil SenTyp nil)
    ); if
  (cond
```

```

(= SenTyp „ATTRIB“)(setq ent SubE EnTyp SenTyp SubE nil SenTyp nil)
(= EnTyp „DIMENSION“);kai objektas dimensija
(progn
  (princ „\nTipas = Dimensija.“)
  (setq Blk ent
    ent SubE
    Dobj (vlax-ename->vla-object Blk)
    EnTyp SenTyp
    SubE nil
    SenTyp nil
  );setq
);progn
)
(ent(princ „\nPasirinkto objekto tipas: „)(princ EnTyp)) ;atspausdina pasirinkto objekto
tipą
);cond
(setq elist (entget ent) ;pateikia duomenis apie elementą sąrašo pavidalu
)
(setq EOBJ (vlax-ename->vla-object ent)
  ELA (getvar „clayer“
);;setq
);;while
(setq rot (vlax-get-property EOBJ ,Rotation) ;nustato objekto posūkį
  h (vlax-get-property EOBJ ,Height) ;nustato objekto aukštį
  pt (vlax-safearray->list(vlax-variant-value (vlax-get-property EOBJ ,InsertionPoint)))
);; gražina sąrašė pasirinkto teksto idėjimo taškų reikšmes
)
(cond
  ((= EnTyp „MTEXT“);kai tipas tekstas
    (progn
      (if Blk (setq h (*(vlax-get-property Dobj ,TextGap)2)))
      (setq AtPt (vlax-get-property EOBJ ,AttachmentPoint))
      (cond
        ((or(= 1 AtPt)(= 4 AtPt)(= 7 AtPt))(setq pt (polar pt(+ rot PI)/( h 2))))
        ((or(= 2 AtPt)(= 5 AtPt)(= 8 AtPt))
          (setq pt (polar pt(+ rot PI)/( (+ (cdr(assoc 42 elist))h) 2))) ;nustato horizontalia
          kryptimi užimamą teksto vietą
        )
        ((or(= 3 AtPt)(= 6 AtPt)(= 9 AtPt))
          (setq pt (polar pt(+ rot PI)(+ (cdr(assoc 42 elist))/( h 2))))
        )
      )
    )
  (cond
    ((or(= 1 AtPt)(= 2 AtPt)(= 3 AtPt))(setq pt (polar pt(+ rot(/ PI 2))/( h 2))))
    ((or(= 4 AtPt)(= 5 AtPt)(= 6 AtPt))
      (setq pt (polar pt(+ rot(/ PI 2))/( (+ (cdr(assoc 43 elist))h) 2))) ;nustato vertikalio
      kryptimi užimamą tekstą
    )
    ((or(= 7 AtPt)(= 8 AtPt)(= 9 AtPt))
      (setq pt (polar pt(+ rot(/ PI 2))(+ (cdr(assoc 43 elist))/( h 2))))
    )
  )
)
)
)

```

```

(setq lr (polar pt rot (+ (cdr(assoc 42 elist))h))
  ul (polar pt (- rot(/ PI 2))(+ (cdr(assoc 43 elist))h))
  ur (polar lr (- rot(/ PI 2))(+ (cdr(assoc 43 elist))h))
)
(vl-cmdf „pline“ „non“ pt „non“ ul „non“ ur „non“ lr „c“) ;piešia rėmus
);;progn
)
((or(= EnTyp „ATTRIB“)(= EnTyp „TEXT“))
  (progn
    (if(= EnTyp „ATTRIB“)
      (setq
        elist(subst(cons 73 (cdr(assoc 74 elist)))(assoc 74 elist)elist) ;
        elist(subst(cons 0 „TEXT“)(assoc 0 elist)elist)
      )
    )
    (vl-cmdf „ucs“ „Object“ ent)
    (setq tb (textbox elist)
      ll (list(-(car(car tb))/( h 6))(-(cadr(car tb))/( h 6)))
      ur (list(+ (car(cadr tb))/( h 6))(+(cadr(cadr tb))/( h 6)))
    )
    (vl-cmdf „rectang“ „non“ ll „non“ ur) ;piešia rėmus
    (vl-cmdf „ucs“ „p“)
  )
); EnTyp = „ATTRIB“ or „TEXT“
); cond
(setq Box (vlax-ename->vla-object (entlast)))
(vl-catch-all-apply „vla-put-Linetype (list Box „CONTINUOUS“) );nustatomas linijos tipas
continuous
(vl-catch-all-apply „vla-put-ConstantWidth (list Box (/ h 20)) );rėmo storis
(setq EnTyp nil SenTyp nil)
)
(princ)
)

```

## 12. Funkcija kopijuojanti objektus į pasirinktą sluoksnį (ctl)

```

;programa kopijuoja objektus į pasirinktą sluoksnį
(defun C:ctl ()
  (setq pt1 ,(0.0 0.0 0.0))
  (setq pt2 ,(0.0 0.0 0.0))
  (setvar „cmdecho“ 0)
  (prompt „Pasirinkite objektus kopijavimui“)
  (setq ent(ssget)) ;primityvai iš brėžinio parenkami po vieną
  (setq nlaer(getstring „Pasirinkite sluoksnį iš kurį kopijuosite objektus ,)“) ;pasirenkamas
naujas sluoksnis
  (command „copy“ ent „“ pt1 „“ pt2) ;kopijuoja objekto taškus
  (command „change“ ent „“ „P“ „LAYER“ nlaer „“) ;pakeičia sluoksnio pavadinimą
  (princ)
)

```

## 13. Funkcija offset su originalo ištrynimu (ofsetd)

```

(defun c:ofsetd()
  (setq dist(getdist(strcat „\n Nurodykite offset atstuma <“) ;atstumo kuriuo kopijuoti
pasirinkimas
  (rtos (getvar „OFFSETDIST“)2 2) ;atspausdiną sisteminį offset atsumą

```

```

„>:“
);strcat
);getdist
)
(if (not dist)
(setq dist(getvar „OFFSETDIST“)) ;jei nepasirenkamas joms atstumas kopijuoja standartiniu
atstumu
); if pabaiga
(while (setq ent(entsel „\npasirinkite objekta: „))
(setq ent (car ent)) ;nustatomas objekto vardas
(setq pt (getpoint „\nPazymekite taska i kuria puse offset: „))
(command „offset“ dist ent pt „“) ;kopijuoja pasirinktu atstumu pasirinkta objekta i pasirinkta
puse
(entdel ent) ;istrina objekto originala
)
(princ)
)

```

#### 14. Funkcija prijungianti prie lanko, linijos, polilinijos nauja polilinija(polilinija)

```

(defun C:polilinija () ;pradedama braizyti polilinija nuo lanko, linijos ar polilinijos pabaigos
(setq osm (getvar „osmode“))
(setvar „OSMODE“ 1)
(setq pt (entsel „\nPazymekite objekta netoli pabaigos „))
(setq endpt (osnap (cadr pt) „endp“)) ;grazina susiejimo komanda susieta taska su
objekto pabaigos tasku
(setq ename (car pt)) ;grazinamas is sarsa pt objekto vardas
(setq etype (cdr (assoc 0 (entget ename)))) ; nustatomas objekto tipas
(setq elaer (cdr (assoc 8 (entget ename)))) ;nustatomas sluoksniu pavadinimas
(setvar „OSMODE“ osm)
(setvar „CLAYER“ elaer)
(if (or (wcmatch etype „ARC“) (wcmatch etype „LINE“)) ;patikrinama ar objekto
tipas yra lankas arba linija
(progn
(command „pedit“ ename „Y“ „“) ;redaguoja su polilinija objekta
(setq ename (entlast))
) ;progn pabaiga
) ;if pabaiga
(if (wcmatch etype „POLYLINE“) ;jei polilinija
(command „convertpoly“ „L“ ename „“) ;optimizuoja polilinija
) ;if pabaiga
(setq edata (entget ename)) ;grazinami duomenys apie objekta ename sarsa
pavidalu
(setq etype (cdr (assoc 0 edata))) ;grazinamas objekto tipas
(if (= etype „LWPOLYLINE“)
(progn
(if (= (cdr (assoc 70 edata)) 0) ;standartinis polilinijos tipas
(progn
(setq P1 (list (car endpt) (cadr endpt)))
(setq P11 (cdr (assoc 10 edata))) ;grazina objekto tasko koordinates
(setq P12 (cdr (assoc 10 (reverse edata)))) ;suraso sarsa edata atbula tvarka
(setq dist1 (distance P1 P11)) ;grazina atstuma nuo pasirinkto objekto
pabaigos iki tasko kuri pasirenkame
(setq dist2 (distance P1 P12))

```

```

    (if (< dist1 dist2)
      (setq endpt P11)
      (setq endpt p12)
    );if
    (command „PLINE“ endpt) ;nuo pabaigos taško brėžia poliliniją
    (while (/= (getvar „cmdactive“) 0) ;sisteminio kintamojo grąžinimas
      (command PAUSE)
    );while pabaiga
    (setq elast (entlast)) ;grąžinamas paskiausiai nubraižyto primityvo vardas
    (if (wcmatch etype „*POLYLINE“) ;jei tipas yra poliliniija
      (command „pedit“ ename „J“ elast „“ „“) ;braižoma poliliniija
      (command „pedit“ ename „y“ „J“ elast „“ „“)
    );if
  );progn
(progn
  (terpri) ;pereinama spausdinti į naują eilutę
  (prompt „Poliliniija uždara. Atidarykite ir bandykite iš naujo „)
  ); if
); if
); progn
(progn
  (terpri) ;pereinama spausdinti į naują eilutę
  (prompt „Jūs turite pasirinkti liniiją, lanką ar poliliniiją „)
  ); progn
);if
);defun

```

### 15. Funkcija įterpianti tekstą į liniiją(inserttxt)

```

(defun C:inserttxt ()
  (setq OS1 (getvar „osmode“))
  (setvar „osmode“ 512)
  (setq l1 (entsel „\Pasirinkite liniija: „)) ;grąžina sąrašą kurio pirmas elementas primityvo
  vardas,o antrasis tasko koordinates kur jis yra
  (setq l2 (entget (car l1))); gražinama informacija apie primityvą sąrašo pavidalu
  (setq l3 (cdr (assoc 0 l2))); iš sąrašo l2 ištraukia primitybo tipą
  (setq P1 (cdr l1)); gražina pasirinktos liniijos taško koordinates
  (setq P1 (car P1)); priskiria x koordinatę p1
  (if (= l3 „POLYLINE“)
    (setq P2 (cadr l1)
      P3 (getpoint „\pazymekite taska pasikartojimui „))
    (if (= l3 „CIRCLE“)
      (setq P2 (cadr l1)
        P3 (cdr (assoc 10 l2)))
      (if (= l3 „ARC“)
        (setq P2 (cadr l1)
          P3 (cdr (assoc 10 l2)))
        (setq P2 (cdr (assoc 10 l2))
          P3 (cdr (assoc 11 l2)))
        )
      )
    )
  )
  (if (= l3 „CIRCLE“)
    (setq A1 (+ (angle P2 P3) (/ pi 2)))
  )

```



```

    (if (= l3 „ARC“)
      (setq A1 (+ (angle P2 P3) (/ pi 2)))
      (setq A1 (angle P2 P3))
    )
  )
  (if (> A1 (/ pi 2)) (setq A2 (angtos (+ A1 pi) 0 4)))
  (if (<= A1 (* pi 1.5)) (setq A2 (angtos (+ A1 pi) 0 4)))
  (if (<= A1 (/ pi 2)) (setq A2 (angtos (* A1 1) 0 4)))
  (if (> A1 (* pi 1.5)) (setq A2 (angtos (* A1 1) 0 4)))

  (if (/= SIZE nil)
    (setq TEXT_SIZE (getdist (strcat „\nNurodykite teksto dydi <“
      (rtos SIZE)
      „>: „
    );;strcat
    );;getdist
    );;setq
    (setq TEXT_SIZE (getdist „\nNurodykite teksto dydi: „))
  );;if
  (if (= TEXT_SIZE nil)
    (setq TEXT_SIZE SIZE)
    (setq SIZE TEXT_SIZE)
  );if
  (if (= TEXT_SIZE nil)(setq TEXT_SIZE (* 2.50 dsc ))) ;jei neįvedamas teksto dydis rašoma
  standartiniu 2.5 dydžiu
  (setq Txt (getstring „\nIveskite teksta: „))
  (command „TEXT“ „M“ P1 TEXT_SIZE A2 Txt)
  (setvar „osmode“ OS1)
  (setq LASTxt (entget (entlast))); gražina paskiausiai braižytą primitivą
  (setq Txt1 (textbox LASTxt))
  (setq P4 (car Txt1))
  (setq P5 (cdr Txt1))
  (setq P5 (car P5))
  (setq P6 (distance P4 P5))
  (setq P7 (/ P6 2))
  (setq A3 (- A1 pi))
  (setq p8 (polar P1 A1 P7))
  (setq p9 (polar P1 A3 P7))
  (command „ERASE“ „L“ „“); ištrina liniją
  (command „BREAK“ P1 „F“ P8 P1)
  (command „BREAK“ P1 P9)
  (command „OOPS“); atsato ką tik ištrintą tekstinį objektą
  (princ)
  )

```

## 16. Funkcija keičianti lanką į apskritimą(tocircle)

```

(defun c:tocircle ()
  (while (not (setq a (entsel „\npasirinkite lanką: „))))
  (setq ent (entget (car a)))
  (if (= (cdr (assoc 0 ent)) „ARC“) ;jei primitivas lankas
    (progn
      (setq ce (cdr (assoc 10 ent))); primitivo pradžios taško koordinatės
      (setq r (cdr (assoc 40 ent))); lanko spindulys
    )
  )

```

```

(setq lyer (cdr (assoc 8 ent))); sluoksnis kuriame yra lankas
(if (cdr (assoc 62 ent))
  (setq spalva (cdr (assoc 62 ent))); lanko spalva
  (setq spalva 256)
)
(if (cdr (assoc 6 ent))
  (setq lt (cdr (assoc 6 ent))); linijos tipas
  (setq lt „BYLAYER“); linijos tipas pagal sluoksnį
)
(setq typ (cons 0 „CIRCLE“) ce (cons 10 ce) r (cons 40 r)
  lyer (cons 8 lyer) spalva (cons 62 spalva) lt (cons 6 lt)
)
(entdel (car a))
(entmake (list typ ce r lyer spalva lt))
(prompt „\nlankas pakeistas apskritimu „)
)
(prompt „\nNe lankas“)
)
(redraw)
(princ)
)

```

### 17. Funkcija pakeičianti pasirinktą tekstą kitu tekstu (keisti)

;klaidos atveju išveda pranešimą apie klaidą

```

(defun MYERR (MSG)
  (if (and
    (/= MSG „Function cancelled“)
    (/= MSG „console break“)
    (/= MSG „quit / exit abort“)
  )
    (progn (princ „\nError: „) (princ MSG))
  ) ;if pabaiga
  (RESTORE)
  (princ)
)
(defun RESTORE ()
  (command „undo“ „e“)
  (setq *ERROR* OLDER)
  (setvar „cmdecho“ 1)
)
;-----radianus verčia į laipsnius
(defun RTD (A) (/ (* A 180.0) pi))
;-----pagrindinė programa
(defun C:keisti ()
  (setvar „cmdecho“ 0)
  (command „undo“ „g“)
  (setq OLDER *ERROR*)
  (setq *ERROR* MYERR); seną klaidą pakeičia į vartotojo
  (setq SEARCHSTRING (getstring t „\nIeškomas tekstas: „))
  (setq SEARCHLEN (strlen SEARCHSTRING)); teksto ilgis
  (setq REPLACESTRING (getstring t „\nPakeisti tekstu: „))
  (setq REPLACELEN (strlen REPLACESTRING))
  (setq SS (ssget)) ;objektai parenkami po vieną

```

```

(setq SLEN (sslength SS))
(setq CTR 0) ;sąrašo elemento indeksas
(setq STRCNT 0) ;skaičiuoja kiek pakeista lelementu
(while (< CTR SLEN)
  (setq SSN (ssname SS CTR)) ;gražinamas iš sąrašo elementas su indeksu ctr
  (setq ELIST (entget SSN))
  (if (or (= „DIMENSION“ (cdr (assoc 0 ELIST)))
          (= „TEXT“ (cdr (assoc 0 ELIST))))
      ) ;or pabaiga
      (RPTEXT) ;kreipiasi į paprogramę
  ) ;if pabaiga
  (if (= „INSERT“ (cdr (assoc 0 ELIST)))
      (RPATTRIB SSN) ;kreipiasi į paprogramę
  ) ; if pabaiga
  (if (= „MTEXT“ (cdr (assoc 0 ELIST)))
      (RPMTEXT) ;kreipiasi į paprogramę
  ) ;if
  (setq CTR (1+ CTR))
) ;while
(RESTORE)
(princ)
) ;defun
;-----paprogramė apdorojanti paprastą ir matmenų tekstą
(defun RPTEXT ()
  (setq TXT (cdr (assoc 1 ELIST)))
  (setq TEXTLEN (strlen TXT))
  (setq RPCTR TEXTLEN)
  (while (>= RPCTR 0)
    (setq PART1 (substr TXT 1 (- TEXTLEN RPCTR))) ;dalis kuri iki pakeitimo
    (setq PART2 (substr TXT (1+ (- TEXTLEN RPCTR)) SEARCHLEN)) ;dalis kurią reikia
    pakeisti
    (setq PART3 (substr TXT (+ SEARCHLEN (1+ (- TEXTLEN RPCTR))))); dalis po
    pakeitimo
    (if (= PART2 SEARCHSTRING)
        (progn
          (setq TXT (strcat PART1 REPLACESTRING PART3)) ;sujungia eilutes į vieną
          (setq ELIST (subst (cons 1 TXT) (assoc 1 ELIST) ELIST))
          (entmod ELIST)
          (setq STRCNT (1+ STRCNT))
          (setq RPCTR (- RPCTR REPLACELLEN))
          (princ (strcat „\rPakeistas „ (itoa STRCNT) „ teksto objektas(i)).“))
        ) ;progn
        (setq RPCTR (1- RPCTR)) ;else
    ) ;if
  ) ;while
) ;defun
;-----paprogramė apdorojanti atributus
(defun RPATTRIB (ENT )
  (setq TEMPENT ENT)
  (setq ENT (entnext ENT)) ;sekančiam primityvui suteikiamas vardas ent
  (if (= NIL ENT)
      (setq ENT TEMPENT)
  )
)

```

```

    (setq ELIST (entget ENT)) ;attributes
  ) ;_ end of if
(while (= „ATTRIB“ (cdr (assoc 0 ELIST)))
  (setq TXT (cdr (assoc 1 ELIST)))
  (setq TEXTLEN (strlen TXT))
  (setq RPCTR TEXTLEN)
  (while (>= RPCTR 0)
    (setq PART1 (substr TXT 1 (- TEXTLEN RPCTR)))
    (setq PART2 (substr TXT (1+ (- TEXTLEN RPCTR)) SEARCHLEN))
    (setq PART3 (substr TXT (+ SEARCHLEN (1+ (- TEXTLEN RPCTR)))))
    (if (= PART2 SEARCHSTRING)
      (progn
        (setq TXT (strcat PART1 REPLACESTRING PART3))
        (setq ELIST (subst (cons 1 TXT) (assoc 1 ELIST) ELIST))
        (entmod ELIST)
        (entupd ENT) ;primityvo regeneracija
        (setq STRCNT (1+ STRCNT))
        (setq RPCTR (- RPCTR REPLACELLEN))
        (princ (strcat „\rPakeistas „ (itoa STRCNT) „ teksto objektas.“))
      ) ;progn
      (setq RPCTR (1- RPCTR))
    ) ;if
  ) ;while
  (setq ENT (entnext ENT))
  (setq ELIST (entget ENT))
  ) ;while
) ;defun
;-----paprogramė keičianti didelės apimties tekstą
(defun
  RPMTEXT ( )
  (setq TXT „“ )
  (setq REPLACEMENT NIL)
  (setq LA (assoc 8 ELIST) ;sluoksnius
  IN (assoc 10 ELIST)
  HT (assoc 40 ELIST) ;teksto aukštis
  WD (cdr (assoc 41 ELIST)) ;plotis
  AT (assoc 71 ELIST) ;prtvirtinimo taškai point
  ST (assoc 7 ELIST) ;stilius
  AN (RTD (cdr (assoc 50 ELIST))) ;posūkio kampas
  CL (cdr (assoc 62 ELIST)) ;spalva
  ) ;_ end of setq
  (foreach
    ELEM ELIST
    (if (= 3 (car ELEM))
      (setq TXT (strcat TXT (cdr ELEM))) ;concatenate all group 3
    ) ;text strings [if present]
  ) ;_ end of foreach
  (setq TXT (strcat TXT (cdr (assoc 1 ELIST)))) ;with group 1 string
  (setq TEXTLEN (strlen TXT))
  (setq RPCTR TEXTLEN)
  (while (>= RPCTR 0)
    (setq PART1 (substr TXT 1 (- TEXTLEN RPCTR)))

```

```

(setq PART2 (substr TXT (1+ (- TEXTLEN RPCTR)) SEARCHLEN))
(setq PART3 (substr TXT (+ SEARCHLEN (1+ (- TEXTLEN RPCTR))))))
(if (= PART2 SEARCHSTRING)
  (progn (setq REPLACEMENT „T“)
    (setq TXT (strcat PART1 REPLACESTRING PART3))
    (setq STRCNT (1+ STRCNT))
    (princ (strcat „\rPakeista „ (itoa STRCNT) „ teksto objektas(i).“))
    (setq RPCTR (- RPCTR REPLACELLEN))
  ) ;progn
  (setq RPCTR (1- RPCTR)) ;else
) ;if
) ;while
(if REPLACEMENT
  (if (< (strlen TXT) 2048) ;negali tekstas būti didesnis nei 2048 simboliai
    (progn
      (entdel SSN)
      (command
        „,mtext“ „0,0“ ;temp insert point
        „,rotation“ AN ;rotation angle
        „,width“ WD ;width
        „,xxx“ ;temp text string
        “
      )
      (if (/= NIL CL)
        (command „,chprop“ „l“ „ „,c“ CL „“)
        (command „,chprop“ „l“ „ „,c“ „,bylayer“ „“)
      ) ;if
      (setq NEWELIST (entget (entlast)))
      (setq NEWELIST (subst (cons 1 TXT) (assoc 1 NEWELIST) NEWELIST))
      (setq NEWELIST (subst LA (assoc 8 NEWELIST) NEWELIST) )
      (setq NEWELIST (subst IN (assoc 10 NEWELIST) NEWELIST))
      (setq NEWELIST (subst HT (assoc 40 NEWELIST) NEWELIST) )
      (setq NEWELIST (subst AT (assoc 71 NEWELIST) NEWELIST) )
      (setq NEWELIST (subst ST (assoc 7 NEWELIST) NEWELIST))
      (entmod NEWELIST)
    ) ;progn
    (prompt „MTEXT viršija maksimalų ilgį“)
  ) ;if
) ;if replacement
) ;defun
(princ)

```

### 18. Funkcija braižanti koncentrinus apskritimus

```

(defun C:ccircle ()
  (setq P1 (getpoint "\nApskritimo centro taskas: "))
  (setq R (getdist "\nSpindulys: "))
  (setq offset (getdist "\nNurodykite offset atstuma, jei nenurodysite spausdins atstumu 0: "))
  (if (not offset)
    (setq offset 0) ;jei nepasirenkamas joks atstumas kopijuoja atstumu 0
  ); if pabaiga
  (repeat (getint "Koncentrinu apskritimu skaicius: ") ;spausdina tiek kiek pasirenkama
    apskritimu
    (command "circle" P1 R)
  )
)

```

```

    (setq R (+ R offset))
  )
  (princ)
)

```

### 19. Funkcija ištrinanti visus pasirinkto tipo objektus(obdelete)

```

(defun C:obdelete ()
  (setq ent (entsel "\nPasirinkite objektus kuriuos norite ištrinti: ")); gražinamas pasirinkto
  primityvo vardas ir koordinatės kur jis yra
  (while ent
    (setq xent (car ent)); primityvo vardas
    (setq obj (entget xent)); primityvas vardu xent išrenkamas iš GDB ir gražinamas sąrašo
    pavidalu
    (foreach x1 '(0 2 6 7 8 62)
      (if (assoc x1 obj)
        (setq flt (cons (assoc x1 obj) flt))
        );if
      );foreach
    (cond
      ((and (= (cdr(assoc 0 obj)) "INSERT")
        (wcmatch (cdr(assoc 2 obj)) "*"*)
        )
      )
      (setq flt (subst (cons 2 "*"*) (assoc 2 flt) flt));sąrašo flt senas elementas pakeičiamas
      nauju
      (reverse flt);surikiuoja sąrašo flt elementus priešinga tvarka
      )
      ((= (cdr(assoc 0 obj)) "DIMENSION")
      (setq flt '((0 . "DIMENSION")))
      )
      );cond
      (setq visi (ssget "x" flt));grąžina primityvų rinkinį susidedantį iš objektų flt esančių
      brėžinyje
      (command "_erase" visi "")
      (if (> (sslength visi) 1)
        (prompt (strcat (itoa (sslength visi)) " objektai ištrinti"));atspausdina keli objektai ištrinti
        (prompt (strcat "1 objektas ištrintas"));spausdina jei brėžinyje buvo tik vienas toks objektas
        );if
        (setq visi nil flt nil);
        (setq ent (entsel "\nPasirinkite objektus trynimui: ")); iš naujo galima pasirinkti tipą
      );while
      (PRINC)
    )
  )

```

### Meniu papildymo failas

```

***POP1
**Pakopinis meniu
[Koskutes papildymas]
[->Darbas su tekstu]
[apremina teksta]^C^C+
(if (not(load "txtbox"))(load "txtbox"));+
txtbox;
[iterpia teksta]^C^C+
(if (not (load "inserttxt"))(load "inserttxt"));+
inserttxt;

```

```

[<- keistiteksta]^C^C+
(if(not( load "keistiteksta"))(load "keistiteksta"));+
keistiteksta;
[--]
[->Braizyti]
[centro linijos]^C^C+
(if(not( load "cenlines"))(load "cenlines"));+
cenlines;
[sudeda taskus]^C^C+
(if(not( load "taskai"))(load "taskai"));+
taskai;
[->xline1]^C^C+
(if(not( load "xline1"))(load "xline1"));+
xline1;
[vertikali]^C^C+
ver;
[<-horizontali]^C^C+
hor;
[polilinja]^C^C+
(if(not( load "polilinja"))(load "polilinja"));+
polilinja;
[lanku linija]^C^C+
(if(not( load "arcline"))(load "arcline"));+
arcline;
[<- koncentriniai apskritimai]^C^C+
(if(not( load "ccircle"))(load "ccircle"));+
ccircle;
[--]
[->Modifikavimas]
[sujungia taskus]^C^C+
(if(not( load "jpoints"))(load "jpoints"));+
jpoints;
[keicia apskritima]^C^C+
(if(not( load "editc"))(load "editc"));+
editc;
[lankas i apskritima]^C^C+
(if(not( load "tocircle"))(load "tocircle"));+
tocircle;
[kopijuoja i sluoksni]^C^C+
(if(not( load "ctl"))(load "ctl"));+
ctl;
[uzrakina]^C^C+
(if(not( load "lock"))(load "lock"));+
lock;
[paslepti]^C^C+
(if(not( load "hid"))(load "hid"));+
hid;
[trinti pasikartojancius]^C^C+
(if(not( load "trintip"))(load "trintip"));+
trintip;
[trina]^C^C+
(if(not( load "obdelete"))(load "obdelete"));+

```

```
obdelete;  
[<-ofsetd]^C^C+  
(if (not (load "ofsetd"))(load "ofsetd"));+  
ofsetd;  
[ilgis]^C^C+  
(if (not (load "length"))(load "length"));+  
length;
```