

შავი ზღვის საერთაშორისო უნივერსიტეტი

კომპიუტერული ტექნოლოგიების და საინჟინრო საქმის  
ფაკულტეტი

მეცნიერული გამოთვლები პროგრამირების ენა PERL-ის  
საშუალებით

მუსტაფა სიმსეკი

სადოქტორო დისერტაციის ავტორეფერატი  
კომპიუტერულ მეცნიერებაში

თბილისი / 2013

**ხელმძღვანელი : პროფ. დოქტ. ნიაზი არი**

**ექსპერტები :**

პროფ. დოქტ. ნოდარ მომცელიძე . . . . .

ასოც. პროფ. დოქტ. გიორგი მანდარია . . . . .

**ოპონენტები:**

პროფ. დოქტ. ანა კობიაშვილი . . . . .

პროფ. დოქტ. ალექსანდრე ედიბერიძე . . . . .

## ნაშრომის ზოგადი მახასიათებლები:

### ნაშრომის თემატური აქტუალობა

PERL არის მაღალგანვითარებული (მაღალი დონის), ზოგად მიზნებზე ორიენტირებული, ინტერპრეტირებადი, დინამური პროგრამირების ენა. PERL თავდაპირველად 1987 წელს ლარი უოლმა განავითარა, როგორც ზოგად მიზნებზე ორიენტირებული UNIX ტექსტური ფორმატის (შაბლონური ჩაწერის) ენა ანგარიშების უფრო გამარტივებული წარმოებისა და დამუშავების მიზნით. ამ დროიდან, მან მრავალი ცვლილება და შესწორება განიცადა და პროგრამებს შორის ფართოდ პოპულარულ ენად იქცა. ლარი უოლი განაგრძობს ძირითადი ენის განვითარებაზე და მის უახლეს Perl 6 ვერსიაზე მუშაობას. Perl-მა ბევრი მახასიათებელი პროგრამირების სხვა ენებიდან იმსხვავება, როგორცაა C, Shell (შაბლონური ჩაწერის (sh)), AWK (ტექსტის ფორმატირების ენა), ეს ენა უზრუნველყოფს ტექსტის მძლავრი დამუშავების შესაძლებლობებს მრავალი თანამედროვე Unix პროგრამული საშუალებებით წარმოებული მონაცემების სიგრძივი მიმართულების გარეშე, ტექსტური ფაილების მარტივი მანიპულაციების გაიოლების მიზნით. Perl-მა შეიძინა ფართოდ აღიარებული პოპულარობა ადრეულ 1990-იან წლებში, როგორც CGI (შაბლონური ჩაწერის) ენა, ნაწილობრივ, მისი გრამატიკული გარჩევა-განხილვის წარმოების უნარების გამო.

CGI-ს (საერთო შესასვლელი ინტერფეისი) გარდა, Perl გამოიყენება გრაფიკული პროგრამირების, სისტემური ადმინისტრირების, ქსელური პროგრამირების, ფინანსური საქმიანობის, ბიოინფორმატიკისა და სხვა მიზნებისთვის. Perl-ს დაერქვა “შვეიცარული ჯარის ჯაჭვური პროგრამირების ენა” თავისი მოქნილობისა და სიმძლავრის გამო. იგი ასევე მოიხსენიება, როგორც “არხში ბეჭდვა-გადაცემა, რომელიც აქცევს ინტერნეტს ერიანად” თავისი საყოველთაობისა და განცდილი არაეფექტურობის გამო.

განხილულია Perl პროგრამირების ენის გამოყენების მექანიზმი ურთიერთგადამკვეთ სფეროებში და საგანმანათლებლო სწავლებისა და სწავლის პროცესის აინჟინრო საქმის კვლევის ისეთ არეალში, როგორც მათემატიკაა,

მოცემულ სადისერტაციო (PhD) თეზისებში შერჩევა განხორციელდა Perl გაფართოებული ფორმატიდან იმ ტექნიკური სტუდენტებისთვის, როგორცაა გრაფიკული პროგრამირება, სისტემური ადმინისტრირება, ქსელური პროგრამირება, საფინანსო საქმე, ბიოინფორმატიკა და სხვა.

ზემოთ ჩამოთვლილ სფეროებში Perl-ის შერჩეული შესაძლებლობებით სარგებლობა სტუდენტებს უფრო მარტივი, სწრაფი, მთავონებეყული და ურთიერთშეთანხმებული მუშაობის საშუალებას უქმნის. ეს აჩვენებს ახალ ტექნოლოგიურ მიღწევებს მოცემულ სფეროში. საგანმანათლებლო პროცესი და მოცემული ტექნოლოგიით სარგებლობა ძალზე მარტივია.

### **Perl-ის ისტორია:**

1987 წლიდან, Perl-ის განვითარების დასაწყისიდან დღემდე, მიმდინარეობს მისი შინაარსის გაუმჯობესება კვლევების შედეგად.

ქვემოთ მოყვანილია განვითარების დაწვრილებითი ინფორმაცია:

**1987:** Perl-ის ბაზისური იდენის შექმნა.

**1988:** Perl 2 გამოშვება.

**1989:** პრაქტიკულად გამოშვებულია Perl 3 და ემატება მხარდაჭერი საშუალებები ბინარული (ორობითი) მონაცემების ნაკადისთვის.

**1991:** გამოიცა “Perl პროგრამირება” და იქცა ენის ფაქტობრივ მითითებად. აგრეთვე, იგივე 1991 წელს, გამოიშვა Perl 4. დაგამოქვეყნდა ვერსიების წიგნი.

**1993:** ხდება Perl 4.036 ვერსიის გამოშვება.

**1094:** გამოდის Perl 5. 1994 წლის მაისში ფუძნდება Perl 5 პორტერების საფოსტოსია და ხდება Perl 5 მუშაობის კოორდინაცია სხვადასხვა პლატფორმებისთვის პორტირებაში და აგრეთვე, 1994 წლის 17 ოქტომბერს Perl 5.000 ვერსიის გამოშვება.

**1995:** 1995 წლის 13 მარტს ხდება Perl 5,001 გამოშვება. 1995 წლის 26 ოქტომბერს საფუძველი ეყრება მრავალმხრივი Perl არქივირების ქსელს (CPAN), როგორც საცავს Perl-ის მოდულებისთვის.

**1996:** 1996 წლის 29 თებერვალს გამოდის Perl 5.002 ახალი პროტოტიპების თავისებურებით. 1996 წლის 25 ივნისს გამოდის Perl 5.003, უსაფრთხოებრივი გამოყენებისთვის.

**1997:** 1997 წლის 15 მაისს გამოდის Perl 5.004, რომელიც სხვა მრავალ საშუალებასთან ერთად მოიცავს უნივერსალურ პაკეტს, Perl-სანიჭებს საბაზისო ობიექტს ავტომატური წარმოებისთვის, რომელიც ახდენს ყოველივეს კლასიფიცირებასა და აძლევს მას მოდულების ვერსიების მოთხოვის უნარს. მეორე მნიშვნელოვანი განვითარება CGI.pm მოდულის ჩართვაში მდგომარეობს, რომელმაც Perl გახადა პოპულარული, როგორც CGI შაბლონური ჩაწერის ენა. Perl ასევე, Microsoft Windows და სხვა ოპერატიული სისტემების პროგრამული ამუშავების მხარდამჭერია.

**1998:** 1998 წლის 22 ივლისს გამოდის Perl 5.005.

**2000:** 2000 წლის 22 მარტს გამოდის Perl 5.6.

**2002:** 2002 წლის 18 ივლისს პირველად ხდება Perl 5.8 გამოშვება.

**2008:** 2008 წლის 14 დეკემბერს გამოდის Perl 5.8 ბოლო, Perl 5.8.9 ვერსია.

**2010:** 2010 წლის 12 აპრილს წარმოებს Perl 5.12.0 გამოშვება.

სულ რამდენიმე წელში მნიშვნელოვნად გაიზარდა კეისის გამოყენების არეალი. იგი განსაკუთრებით ფართოდ გამოიყენება ინფორმაციული

ტექნოლოგიების (IT) ადმინისტრირებაში. ეს თეზისები შეიძლება გამოყენებული იქნეს უნივერსიტეტის ლექტორების მიერ და კვლევით პროექტებში, როგორც ეს მოცემულ სადისერტაციო თეზისებშია ნაჩვენები.

აქამდე მრავალი პროგრამირების ენა გამოიყენებოდა. თუმცა Perl-ისა და მისი შერჩეული ნაწილების უპირატესობები თავსაჩინოდ ილუსტრირებულია წინამდებარე თეზისებში.

კომპიუტერული მეცნიერების სფეროში Perl-ის ირგვლივ ჩატარდა უამრავი კონფერენცია და გამოიცა მრავალი პუბლიკაცია.

Perl-ის ექსპანსია, როგორც PDL, გაუმჯობესდა კვლევის ტექნიკების თავსაზრისით. სწორედ ამიტომაცაა, რომ მოცემულ თეზისებში ყურადღების კონცენტრირება ძირითადად, ამ საგანზე ხდება (იხ. მითითებული ლიტერატურა). მომავალ წლებში Perl-ის ირგვლივ ბუბლიკაციები კიდევ უფრო გაიზრდება. მზარდია ამგვარი პუბლიკაციების მოთხოვნა და მხარდაჭერა პერიოდული გამოცემების მხრიდან.

Perl-ის სისტემა წარმოადგენს ღია წყაროს პროდუქტს. მისი მომხმარებლები (მეცნიერები, აკადემიკოსები, სტუდენტები, აკადემიური პერსონალი და მასწავლებლები) არიან, რომლებსაც შეუძლიათ სტანდარტული დოკუმენტების მოძიება. Perl-ის მრავალი დოკუმენტის პეციალურ სფეროებში გამოიყენება, როგორც ამ თეზისებშია ნაჩვენები.

ეს თეზისები ახდენენ იმის დემონსტრირებას, თუ მეცნიერების, ინსტრუქტორებისა და სტუდენტებისთვის, Perl-ის გამოყენებით სულ მცირე, რისი გაკეთება არის შესაძლებელი მეცნიერულ პროგრამირებაში მათემატიკური ამოცანების გადასაწყვეტად.

მოცემულ თეზისებში ნაჩვენებია მეთოდები მეცნიერებისთვის, აკადემიური მუშაკების, ინსტრუქტორებისა და სტუდენტებისთვის. შერჩეული მაგალითები წარმოგიდგენენ სხვა გამოყენების ნიმუშებსაც. ეს ნაშრომი, რამდენადაც ჩემთვის ცნობილია, წარმოადგენს უნიკალურ გამოკვლევას. მოცემული დისერტაცია (PhD)

ტიპური საწყისი სახალი კვლევებისთვის Perl-ის გამოყენების ზემოთ აღნიშნულ არეალში. მითითებულ სფეროებში მოცემული ნაშრომი შეიძლება გამოდგეს მომავალი მეცნიერული და აკადემიური კვლევების საფუძვლად. ეს შეიძლება გახდეს სპეციალობები “ათათურქ ალათუს საერთაშორისო უნივერსიტეტის” საზღვრებში, სადაც კომპიუტერული მეცნიერების თეორია და პრაქტიკა და კომპიუტერული ტექნოლოგიები აქტიურად გამოიყენება განათლების სისტემასა და ინდუსტრიულ პროექტებში.

### **საგნობრივი ბმულები (კავშირები):**

თეზისების თემა წარმოადგენს ე.წ. case-study-ის, ანუ Perl-ის ურთიერთგადასაკვეთ სფეროებში კონკრეტული შემთხვევის კვლევას, გამოყენებადს მათემატიკურ და კომპიუტერულ მეცნიერებებში. ასევე ტესტირებულია მისი სწავლების, სწავლისა და პრაქტიკული მანქანური ინჟინერიის პედაგოგიური პროცესებიც.

### **ნაშრომის მიზნები და ამოცანები:**

ნაშრომის მიზანია აჩვენოს Perl-ის გამოყენების უპირატესობები მეცნიერების, ინსტრუქტორებისა და სტუდენტებისთვის მათემატიკური პრობლემების გადასაჭრელად, ამ ნაშრომით გამიზნული გვაქვს გავსწავლოთ ახალი კვლევებისთვის, აგრეთვე, ხელი შევუწყოთ Perl-ის სწავლებისა და სწავლის მიმდინარეობას, რაც საგანმანათლებლო პროცესს ბევრად უფრო მარტივს, სწრაფსა და შემოქმედებითს გახდის.

მოცემული ნაშრომით გამიზნული გვაქვს მოცემულ სფეროში გამოინახოს შემდეგი პრობლემების გადაჭრის გზები:

- მეცნიერულ და საგანმანათლებლო პროცესებში Perl-ის გამორჩეული მნიშვნელობის განმარტება;
- ინფორმაციის წყაროს ურთიერთდაკავშირებული მომხმარებლების სიახლოვის, ფამილარულობის შექმნა;
- პრობლემის ეფექტური გადაწყვეტის ძიება;

- ამგვარი გადაწყვეტის პრეზენტაცია, პრაქტიკული გატარება;
- ტექნიკური საშუალების მომზადება, რომელიც ურთიერდაკავშირებულ მომხმარებლებს უზრუნველყოფს ახალი კვლევებისა და აკადემიური კვლევა-ძიების ხელსაყრელი შესაძლებლობებით.

### **ნაშრომის მეთოდოლოგია:**

ნაშრომის მეთოდოლოგია ეფუძნება კომპიუტერულ ტექნიკებს, კომპიუტერულ მეცნიერებასა და მათემატიკასცოცხალი მაგალითების გამოყენებით.

### **ნაშრომის მეცნიერული სიახლე:**

- კომპიუტერული მეცნიერებისა და მათემატიკის მეთოდების ერთობლივი ამუშავება;
- შაბლონების მოდელის განვითარება Perl-ის გამოყენებისთვის;
- მხარდამჭერიინსტრუმენტის განვითარება საგანმანათლებლო პროცესისა და კვლევების წარმოებისთვის;
- პირველი ნაბიჯის განვითარება Perl-ის სფეროში (დომეინში) ახალი მეცნიერული და აკადემიური კვლევებისთვის.

### **ნაშრომის პრაქტიკული შედეგები:**

მკვლევარებს ეძლევათ შაბლონი, ნიმუში თავიანთი კვლევების საწარმოებლად. Perl-ის სტრუქტურებიმათ დაეხმარებათ პრობლემის გადაწყვეტის ტექნიკების უკეთ გაგებაში.

მომხმარებლებს Perl-ის დახმარებით ექმნებათ მრავალი სათანადო მაგალითისა და გრაფიკის, პასუხების ჩამონათვალისა, თუ ანიმაციის ილუსტრირების შესაძლებლობა. მათეძლევათ უნარი მეტად ისარგებლონ კომპიუტერით გაშუალებული სწავლებით, ანუ მოახდინონ სასწავლო პროცესის აგება კონსტრუქტივისტური თეორიის შესაბამისად, რომელიც დღეს ასეთი პოპულარულობით სარგებლობს. მათ შეეძლებათ დაეუფლონ სწავლების



ისეთ მეთოდებს, როგორცაა პრობლემაზე ბაზირებული და ჯგუფური კვლევის მეთოდები.

### **ნაშრომის საიმედოობა:**

მაგალითებისა და შაბლონების საიმედოობა საკმაოდ მაღალია და დამოუკიდებელი ოპერატიული სისტემებისაგან. ყველა მაგალითის ტესტირება მრავალგზის წარმატებით განხორციელებულია,

### **ნაშრომის სტრუქტურა და მოცულობა:**

დისერტაცია შედგება ოთხი ნაწილის, დასკვნის, მითითებული ლიტერატურისა და დანართისაგან. ნაშრომი 150 გვერდიანია, შეიცავს დიაგრამებს, ცხრილებსა და დანართებს. ლიტერატურის სია მოიცავს 199 დასახელებას, ამათგან მხოლოდ 40 მოცემულია ამ მოკლე კრებულში).

### **ნაშრომის შინაარსი**

**ნაწილი 1** წარმოადგენს შესავალს ნაშრომის საფუძვლების შესახებ. იგი მოიცავს კომპიუტერული პროგრამირების სისტემების განვითარების ისტორიას, საინფორმაციო ტექნოლოგიური საშუალების (ე.წ. სოფტის (Software), მეცნიერული და საგანმანათლებლო პროცესის შეფასებას კომპიუტერებთან, კომპიუტერული პროგრამირების სისტემებთან, კომერციულ და ღია წყაროს პროგრამებთან, აღწერის ენებთან, Perl და PDL-ის საშუალებით ჩატარებულ კვლევებთან მიმართებაში.

**ნაწილი 2** შეეხება Perl-ის პროგრამირების ენას. ამ ნაწილში ჩვენ წარმოვადგინეთ ლიტერატურის ჩამონათვალი მოცემულ საკითხებზე:

**ცვლადის ტიპები:** სკალები, რაოდენობრივი ცვლადები, მიმდევრობითი ცვლადები, ცალმხრივად კვოტირებული მიმდევრობები, ორმაგად კვოტირებული მიმდევრობები, კლავიატურიდან მიღებული მნიშვნელობები, წყობის მითითება, ორგანზომილიანი წყობა.

**მართვის, კონტროლის სტრუქტურა:** განაცხადის ბლოკები, თუ/არ განაცხადი, პირობითი, მანამ/ვიდრე განაცხადი, გააკეთე მანამ/ვიდრე არ განაცხადი, იმდენად, რამდენადაც განაცხადი, თითოეულისთვის განაცხადი (The Foreach Statement).

**შეტანა/გმოტანა:** ფაილის გახსნა წასაკითხად/დასაწერად, ფაილის დახურვა, ფაილიდან კითხვა, სიისა და სკალური კონტექსტი, ფაილისთვის მიწერა, ახლომდებარე დოკუმენტები (Hrer Dokuments), ბინარული ფაილების კითხვა და წერა.სავარჯიშო და გადაწყვეტა.

**ფუნქციები:** ფუნქციებზე მითითება, ცვლადების ლოკალიზება, სავარჯიშო და გადაწყვეტა.

**დასრულებული პროგრამის დაწერა:** ნიუტონის მიზეზის მოძიების მეთოდი, მაგალითი, დროითი პროგრამა.

**ფაილისა და დირექტორიის (კატალოგის) მანიპულაცია:** დირექტორიები, დირექტორიებში (კატალოგების) შეღწევა (Accessing Directories), დირექტორიების შეცვლა, მოგზაურობა (Globing), დირექტორიების შექმნა და დატოვება (Creating and Remainig Directories), დირექტორიის სამართავები (Directory Handles), დირექტორიის (კატალოგის) სამართავის გახსნა და დახურვა (Openning and Closing a Directory Handle), დირექტორიის სამართავის წაკითხვა. ფაილები, ფაილის გადანაცვლება (ადგილმდებარეობის შეცვლა), ფაილის სახელის გადარქმევა, ფაილის ტესტი.

**პროცესი:** სისტემისა და Exec-ის გამოყენება, პროცესების გამოყენება, როგორც ფაილსამართავებისა (Filehandles).

**რეგულარული ექსპრესიები:** რეგულარული ექსპრესიების კონცეპტები (ცნებები), მარტივი რეგულარული ექსპრესია, სათანადო (წყვილადი) რეგულარული ექსპრესიის ტესტირება, სუროგატები, ბრძანებების გაყოფა და გაერთიანება, ფუნქციების გაყოფა, ფუნქციების გაერთიანება.

**მოდულები:** მოდულების გამოყენება, მოდულების შექმნა, ქვეშაბლონები.

**ობიექტზე ორიენტირებული პროგრამირება:** ობიექტზე ორიენტირებული პროგრამირების ცნება, პაკეტი, როგორ შეიქმნას ობიექტები, შვილების შექმნა, ობიექტზე ორიენტირებული პროგრამირების კლასები, Perl ობიექტების რეპრეზენტაცია, პოლიმორფიზმი, მემკვიდრეობა.

### **Perl-ის სტრუქტურა:**

Perl წარმოადგენს ზოგად მიზნებზე ორიენტირებულ პროგრამირების ენას, თავდაპირველად გამოიზნულს ტექსტური მანიპულაციების საწარმოებლად, მაგრამ 2010 წლიდან დაიწყო მისი გამოყენება ამოცანების უფრო ფართო სპექტრის შესასრულებლად, როგორცაა სისტემური ადმინისტრირება, WEB განვითარება, ქსელური პროგრამირება, თამაშები, ბიონფორმატიკა და GUI განვითარება. აღნიშნულიენა გამოიზნულია პრაქტიკული მოხმარებისთვის (ეფექტური, ამომწურავი, მარტივი გამოყენების), ისევე როგორც სასიამოვნო, ლამაზი ვიზუალური აღქმის (პაწაწინა, მოხდენილი (ელეგანტური), მინიმალური) თვალსაზრისით. მისი მთავარი თავისებურებები მიმართულია მულტიფუნქციური (მრავალმხრივი) პროგრამირების პარადიგმების მხარდაჭერაზე (პროცედურული, ობიექტზე-ორიენტირებული და ფუნქციონალური სტილების მხრივ), მიმართავს გაანგარიშების მეხსიერების მენეჯმენტს (ციკლური შემოწმების ნაგვის კოლექტორის გარეშე), გააჩნია ჩამონტაჟებული მხარდაჭერი საშუალებები ტექსტის პროცესირებისთვის და სამმხრივი მოდულებისფართო კოლექცია.

Perl-ის სტრუქტურა ფართოდ გამოიყენება მოცემულ თეზისებში.

### **Perl-ის მახასიათებლები, თავისებურებები:**

Perl-ის მთელისტრუქტურა ფართოდ C-დან წარმოსდგება. Perl-ის თავისი ბუნებით, პროცედურულია: ცვლადებით, ექსპრესიებით, დანიშნულებრივი (დავალების) განცხადებებით, უსაზღვრო (არალიმიტური) დაკავშირების ბლოკებით, მართვის (კონტროლის) სტრუქტურებითა და ქვეშაბლონებით.

Perl გარკვეულ თავისებურებებს იღებს აგრეთვე, shell პროგრამირებიდან, (პროგრამა-გარსიდან).ყველა ცვლადი აღინიშნება წამყვანი sigils(?).Perl იღებს სიას Lisp-დან (Lisp წარმოადგენს პროგრამირების ენას ლოგიკური პროგრამებისთვის), დანაწევრებას (“ასოციაციური წყობა”) AWK-დან (ტექსტის პროცესირების ენა) და რეგულარულ ექსპრესიებს sed-დან.ესბევრად ამარტივებსა და უფრო გაიოლებულს ხდის გრამატიკული გარჩევა-განხილვის, ტექსტის მართვისა და მონაცემების მენეჯმენტის მრავალ ამოცანას. Perl-ის ყველა ვერსია ავტომატურად ახდენს მონაცემების ტიპიზაციასა და მეხსიერების ავტომატურ მართვას. ინტერპრეტატორმა იცის ყოველი მონაცემის ობიექტის ბეჭდვისა და დაგროვება-შენახვას მოთხოვნები პროგრამაში; იგი ათავსებს ანერისმათ, აუცილებელიაანგარიშების მიმართვის გამოყენებით (ამდენად მას არ შეუძლია საქმე ჰქონდეს მიმდებარეცილკულარულ მონაცემთა სტრუქტურებთან ხელით ჩარევის გარეშე).

მკვლევართამოთხოვნილებები შეირჩევა Perl(PDL) თავისებურებებიდან.

### **Perl აპლიკაცია (განაცხადი):**

Perl არის მრავალი და სხვადასხვა, მრავალფეროვანი აპლიკაციების (განაცხადების) მატარებელი.რომელიც შედგენილია მრავალი სტანდარტული და სამმხრივი მოდულების ხელმისაწვდომობის შედეგად. პროგრამისტები Perl-ს იყენებენ CGI ფორმატის ჩანაწერებისთვის. Perl ცნობილია, როგორც ერთ-ერთი “სამი P-თაგანი“(Python და PHP-ისგვერდით), იგია ყველაზე პოპულარული დინამიური ენა წერითი Web აპლიკაციების გასაფორმებლად. იგი აგრეთვე, გახლავთ ერთ-ერთი ინტეგრალური კომპონენტი პოპულარული LAMP (Software bundle, Linux, Apache, MySQL და Perl/ Python/PHP)გადაწყვეტების ერთობლიობისა Web განვითარების მიზნით.

Perl-ს იყენებენ, როგორ მწება ენას, რომელიც ერთად ამაგრებს, ერთმანეთთან აბამს სისტემებსა და ინტერფეისებს, რომლებიც საგანგებოდ არ არის შედგენილი, სპეციფიკურად ინტერპრეტაციისათვისან “მონაცემების სამართავად”, იგი ახდენს მონაცემთა დიდი მიმდევრობების კონვერტირებას, ან

გაშვებას იმგვარი ამოცანების შესასრულებლად, როგორცაა ანგარიშების შექმნა. კომბინაციები Perl-ს აქცევს ერთ-ერთ პოპულარულ, უნივერსალური მიზნების მქონე, მულტიფუნქციურ ენად სისტემური ადმინისტრირებისთვის, ნაწილობრივ, იმიტომაც, რომ მოკლე პროგრამების შეტანა და ფაშვება შესაძლებელია მხოლოდ ერთი ბრძანების სტრიქონით.

Perl-ის კოდი შესაძლებელია გახდეს პორტაბელური Windows და UNIX ოპერატიული სისტემების გავლით.

Perl-ის კოდი გამოიყენება ინფორმაციული ტექნოლოგიური ინსტრუმენტისე.წ. სოფტის (Software) მომხმარებლებისთვის (როგორც COTS (კომერციული ოპერატიული სისტემები) და წინასწარი შეკვეთის) საინფორმაციო ტექნოლოგიის შეფუთვისა და ტექნიკური აღჭურვილობის სამშენებლო და საქმიანი დოკუმენტების გაიოლების მიზნით.

გრაფიკული მოხმარების ინტერფეისები (GUIs) შეიძლება განვითარდეს Perl-ის გამოყენების შედეგად. მაგალითად, Perl/Tk ზოგადი დანიშნულების მქონეა, რომელიც შესაძლებელს ხდის მომხმარებლების ინტერაქციას Perl შაბლონური ჩანაწერების (სკრიპტების) მეშვეობით. ამგვარი ინტერაქცია შეიძლება იყოს როგორც სინქრონული, ასევე ასინქრონული ხელახალი, უკან დაბრუნებული გამოძახებების, (callbacks) გამოყენებით GUI განახლების მიზნით.

Perl ფართოდ გამოიყენება აგრეთვე, საფინანსო საქმესა და ბიოინფორმატიკაში, სადაც იგი მნიშვნელობას იძენს აპლიკაციის სწრაფი განვითარებისა, თუ უარყოფის უნარისა და მონაცემთა ვრცელი ერთობლიობების მართვის შესაძლებლობების გამო..

ეს თეზისები ახდენენ გრაფიკული მოხმარების ინტერფეისის პრეზენტაციას Perl-ის პროგრამისთვის.

ეს პროგრამა იღებს შესატანი ინფორმაციის გარკვეულ პარამეტრებს კლავიატურიდან და შემდეგ ბეჭდავს კვადრატში ფაილის შემდეგ მნიშვნელობებს.

```
#!/usr/bin/Perl-w

$val='0';

while ($val ne 'end'){

print "please enter a number (end to finish) \n";

$val=<STDIN>;
```

```
chomp($val);
push(@values,$val);
}
pop (@values);
open (FILE,"> squares.txt");
foreach $val (@values) {
print FILE "The square of $val is ",$val*$val,'\n';
}
close(FILE);
```

**ნაწილი 3** წარმოგიდგენს ცნობებს Perl მონაცემთა ენის შესახებ (PDL).

ამ ნაწილში ჩვენ მიმოვიხილეთ მოცემულ საკითხებზე არსებული ლიტერატურა და აღვწერეთ აპლიკაციების ყველა მათემატიკური თემა PDL-ში.

**PDL-ის შესავალი:** რას წარმოადგენს PDL, უპირატესობები და ნაკლოვანებები, PDL-ის ინსტალაცია.

**PDL შექმნა:** რა არის piddle, მარტივი piddle-ები, რთული piddle-ები.

**PDL მონაცემთა ტიპები:** ტიპიური კონვერსია და შექმნა, რთული ტიპები.

**არითმატიკა:** ბაზისური, მაღალი დონის.

**მოდულიზაციური piddle-ები:** დაყენების ფუნქციები (პარამეტრები) (Set function), კვეთის ფუნქცია (Slice function), ბაზისური კვეთა (Basic Slicing), მოდიფიკაციური კვეთა (Modifying Slicing), უფრო სპეციფიკური კვეთა (Advanced Slicing), სიის ფუნქცია (List function), სიის ინდექსების ფუნქცია (Listindices function), სამაგრის ფუნქცია (Clip function), მაკეტის ფუნქცია (Dummy function), ზესამაგრის ფუნქცია (Hclip function), ქვესამაგრის ფუნქცია (Lclip function), One2nd ფუნქცია, სათამაშო კოჭების ღერძის ფუნქცია (Dice and Dice \_Axis function), MV, გაცვლისა და ხელახალი დალაგების ფუნქცია (Reorder function), განზომილებების კომბინაცია ჯგუფური ფუნქციის (Clump function) მეშვეობით.

**PDL გრაფიკები:** 2D გრაფიკები, პლოტერის (გრაფამგების) ბრძანებების პარამეტრები, ისარი, ისრის ზომები, ღერძი, ღერძის ფერი, საზღვრები (Borders), Charsize, შევსების ტიპი (Filltype), ფონტი, შტრიხირება, ორმხრივ კიდებზე სწორება (Justify), ხაზის სტილი, ხაზის სისქე, პლოტის პოზიცია, სიმბოლო, აღმართვა პლოტის არეში (Setting up the plot area), ასპექტი, ფანჯრის სიგანე (WindowWidth), ფანჯრის X ზომები (WindowXSize), ფანჯრის Y ზომები (WindowYSize), ხაზებისა და პლოტის (მონაკვეთის) წერტილების ხაზვა, იმიჯების (გამოსახულებების) ეკრანზე გამოტანა (Displaing imiges), ფერების პანელი/სოლი (Colour bar/wedge), დაყენება (Set), ადგილმდებარეობის შეცვლა

(Displacement), სიგანე (Width), წინა პლანის ფონი (Eoreground), უკანა პლანის ფონი (Background), იარლიყი (Lable), პორიზონტალური პლოტები და ვექტორის ველები (Contour plots and vector fields), სახაზავი ფორმები, წრეები, ელიფსი, ტექსტი და ლეგენდები, ტექსტი, X პოზიცია(Xpos), Y პოზიცია(Ypos), სიგანე, წონა, ტექსტის ფრაქციები(TextFraction), ტექსტის შეცვლა (TextShift), ვერტიკალური სივრცე (VertSpace),ფერების გამოყენება, 3D გრაფიკები, პარამეტრული გრაფიკები, 3D გრაფიკული ობიექტები, GIMP-ის გამოყენება, Direct Pixel Access.

**მათემატიკა PDL-ით:**რიცხვები, ოპერატორები, კვადრატული ფესვი, კარტეზიული პროდუქტი (Cartesian product), პოლინომინალები, პარაბოლა, ტრიგონომეტრიული ფუნქციები, ურთიერთქმედების ფუნქციები, გადახრისგანსაზღვრა, სერიული მსაზღვრელები, ინვერსიული ფუნქციები, რადიალური კოორდინატული კონვერსიები, კოორდინატთა სისტემა, 3D კუთხური კონვერსიები, ლოგარითმი, რთული რიცხვი, კომბინაცია, პერმუტაცია, ალბათობა, დერივაცია (ცდომილება), ინტეგრალი, მატრიცები.

**რა არის PDL?** Perl წარმოადგენს მონაცემების ენას. PDL რიცხვითი გამოთვლების ენაა Perl-ის რაოდენობრივი (რიცხვითი) გაფართოების ბაზაზე.PDL (“Perl მონაცემთა ენა”) აძლევს სტანდარტულ Perl-ს კომპაქტური დაგროვების, შენახვისა და ვრცელი N-განზომილებიანი მონაცემთა მიმდევრობების, მასივების სწრაფი მანიპულირების უნარს, რომლებიც მეცნიერული გამოთვლების განუყოფელ ბირთვს შეადგენენ.მას შეუძლია მარტივი სახით სწრაფი მატრიცული კალკულირება.

მოცემული იდეა იმაში მდგომარეობს, რომ Perl-ის თავისუფალ, მიმდევრობაზე ორიენტირებულ, ციფრულ ენად ისეთი კომერციული პაკეტების მსგავსად, როგორცაა IDL და MatLab.მომხმარებელს შეუძლია ჩაწეროს მარტივი ექსპრესიები Perl-ის ერთიანი რიცხვითი მიმდევრობების მყისიერი მანიპულირების მიზნით..მარტივი ინტერაქტიური shell (perldl) პროგრამა-გარსი უზრუნველყოფილია ბრძანებათა სტრიქონის (PDL) მოდულთან ერთობლივი ამუშავებითPerl ჩანაწერებში (შაბლონებში, სკრიპტებში) მოხმარებისთვის.



არსებობს PDL გამოყენების ორი ხერხი: ამუშავება Perl პროგრამიდან და ინტერაქტიურად.

### **PDL უპირატესობები:**

- იგი თავისუფალი და ღია წყაროს ინფორმაციული ტექნოლოგიური ინსტრუმენტია.
- იძლევა მატრიცული და რამდენიმე განზომილებიანი გამოთვლების განხორციელების შესაძლებლობას; არ ჩერდება მეორე განზომილებაზე, არამედ უშვებს გამოთვლებში ბევრად მეტი განზომილების ჩართვას. ამ თვალსაზრისით, იგი სრულყოფილად ჩაფიქრებულია და შესანიშნავად ახდენს სკალირებას.
- გამოიირჩევა ძლიერი სისწრაფით.
- გრაფიკული ინტერფეისი საკმაოდ მოხერხებულია.
- 2D გრაფიკები დაფუძნებულია PGPLOT ბაზაზე, რომელიც საკმაოდ ადეკვატურია როგორც პუბლიკაციების ხარისხის მახასიათებლების, ასევე ყოველდღიური პლოტირების (გრაფაგების) მოთხოვნებისთვის.
- გრაფიკები ემყარება MESA ბიბლიოთეკას, ძალზე მოხერხებულია მარტივი სახით სამგანზომილებიანი პლოტირებისთვის და ახდენს რეალური დროის განკარგვას პლოტირების მონაცემთა როტაციისთვის.

### **PDL ნაკლოვანებები:**

- ხელმისაწვდომი ციფრული ბიბლიოთეკა გამოთვლების საწარმოებლად არ არის ძალიან ფართო. ეს თავისებურება სწრაფად ცვალებადია, რამდენადაც ბევრი, GSL-ის ან SLATEC-ის მსგავსი რიცხვითი შაბლონი იხმება PDL-ში.
- იგი თვითონ არ უზრუნველყოფს რთული რიცხვებით გამოთვლებს.
- PDL-ის 2.004 ვერსიაში კომპლექსური პაკეტი დამატებულია, მაგრამ იგი უბრალოდ, არითმათიკულ მოქმედებებს რთულ რიცხვებზე მარტივად ვერ აწარმოებს.

- PDL-ით პროგრამირება მისი შესაძლებლობების სრული ამოქმედებისთვის სხვა ენებთან შედარებით უფრო მეტად საჭიროებს აბსტრაქტულ აზროვნებას,

### **PDL ილუსტრაციის მაგალითები:**

PDL იყენებს გარკვეულ არითმეტიკულ ოპერაციებსა და ფუნქციებს. ესენია: 'მიმატების (add), გამოკლების (subtract), გამრავლების (multiply), გაყოფის (divide) მოქმედებები.

#### **მაგალითი 1.**

ეს მაგალითი აჩვენებს, თუ როგორ გამოიყენება PDL-ის არითმეტიკული ოპერატორები.

```
perl> $x=pdl([5,4])
```

```
perl> $y=pdl([4,3])
```

```
perl> $x*$y
```

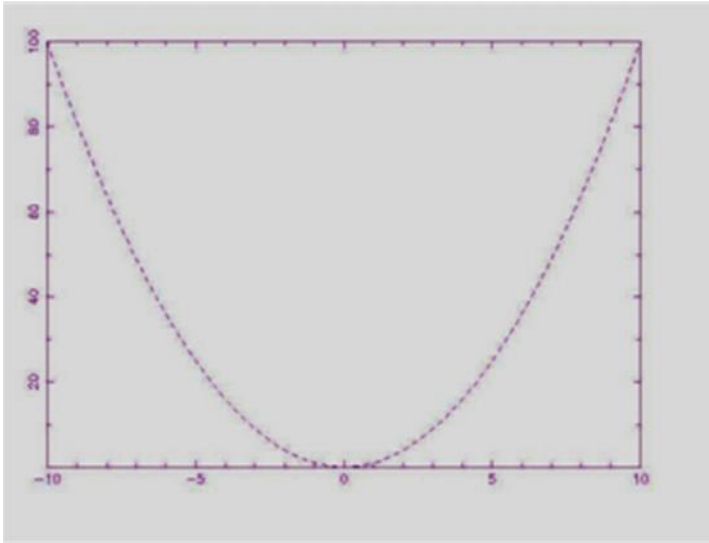
```
[20,12]
```

#### **მაგალითი 2:**

შემდეგი მაგალითი გამოითვლის და აგებს პარაბოლის წერტილებს.

```
perl> $x=zeroc(100)->xlinvals(-10,10);
```

```
perl> line $x, $x*$x, {LINESTYLE=>'DASHED', colour-green};
```



### **PDL-ის გრაფიკები:**

მონაცემთა ანალიზის ნებისმიერი პაკეტის ცენტრალურ მოთხოვნას წარმოადგენს მონაცემთა ვიზუალიზაციის შესაძლებლობა. ამ ამოცანას PDL-დანაც განსხვავებული სახითასრულებს სხვა პაკეტებთან შედარებით, რომლებშიც ჩაშენებული ტიპის გრაფიკების ბიბლიოთეკა არ გამოიყენება, სანაცვლოდ, იგი იყენებს სხვა თავისუფლად ხელმისაწვდომ ექსტერნალურ პაკეტებს. ამ თავში ჩვენ ყურადღების ფოკუსირებას მოვახდენთ 2D ბლოტირების პაკეტზე, PGPLOT და 3D გრაფიკების აგების დეტალიზებაზე.

### **2D გრაფიკები**

ორგანზომილებიანი გრაფიკები PDL-ში იგება ჩვეულებრივ, PDL::Graphics::PGPLOT მოდულით. გრაფიკების ფუნქციებისთვის გამოიყენება მძლავრი PGPLOT ბიბლიოთეკა. ამისთვის უნდა ვისარგებლოთ PDL::Graphics::PGPLOT პაკეტით ბრძანებების მისაცემად. ეს შესავალი ინტერაქტიურობასა და perldl-ის გამოყენებაზეა აგებული.

```
perldl>use PDL::Graphics::PGPLOT;
```

2D ფუნქციის პლოტირებისთვის გამოიყენება `line` ბრძანება. ქვემოთ მოცემული მაგალითი დახატავს ხაზს.

```
perldl> x=pdl([0,1,2])  
perldl> $y=pdl([4,5,6])  
perldl> line([$x,$y])
```

ახლა საჭიროა გრაფიკების აპარატის გახსნა, რომლის ჩამონათვალიც მცირეა და მოითხოვს PGPLOT მხარდაჭერას.

აქ PDL იყენებს ნორმალური პლოტის ფანჯარას, რომლის კვლავ გამოყენებაც შეიძლება.

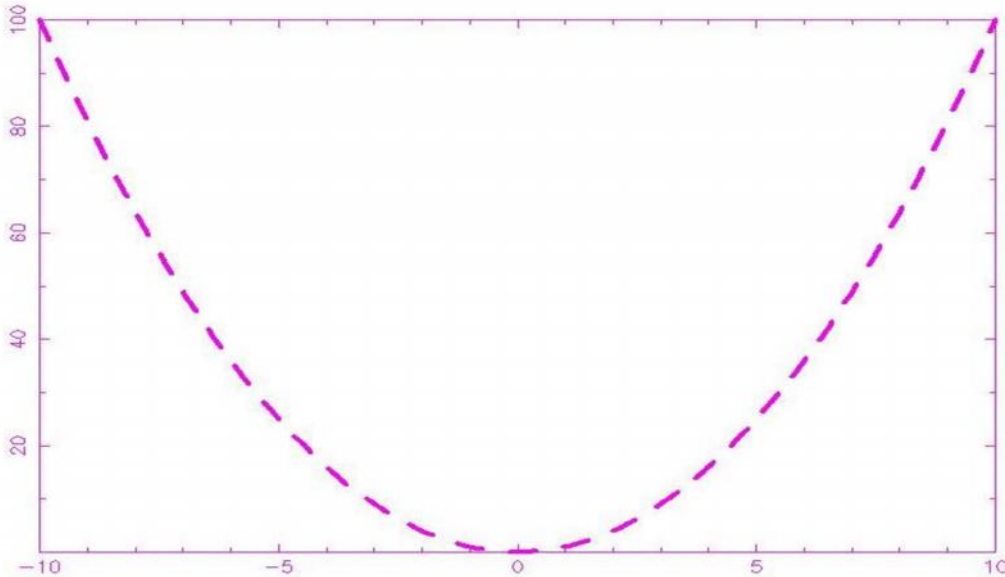
```
perldl>dev('/xs')
```

თავდაპირველად საჭიროა ცვლადის განსაზღვრა რაიმე გრაფიკის ასაგებად. პირველი გეგმა წარმოადგენს მარტივად პარაბოლის აგებას და Gaussian (ზარი) ფუნქციის ამუშავებას. ახლა კი საჭიროა რაიმე `x` ცვლადის შემოღება, რომელიც როგორც პოზიტიური, ასევე ნეგატიური მნიშვნელობის იქნება.

```
perldl>$x=zeros(100)->xlinvals(-10,10)
```

იგი ქმნის 100 ელემენტის გამოსახულებას -10-დან დაწყებული 10-ით დამთავრებული. მას შეუძლია პარაბოლის გამოსახვა.

```
perldl>line $x,$x*,$x,{LINESSTYLE=>'DASheD',colour=>green} (x,y)
```



### მაგალითი 3:

დააყენეთ მოხაზულობის (კონტურის) პარამეტრები (option Countours) სასურველი გამოსახულების დონეებისა, როგორც `pidle` და იარლიყები (Labels) არაწესიერ მიმდევრობათა წყობის მისაღებად.

```

use PDL;

PDL::Graphics::PGPLOT;

dev('/xs');

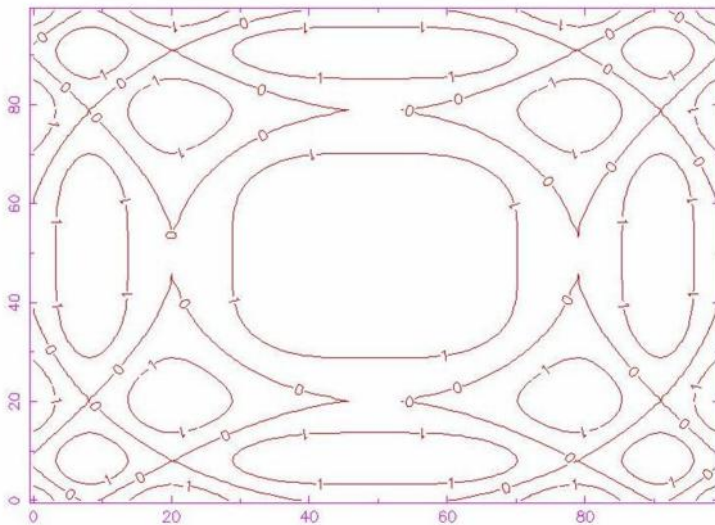
$y=ylinvals(zeroes(100,100),-3,3);

$x= xlinvals(zeroes(100,100),-3,3);

$z=cos ($x**2)+cos($y**2);

cont $z,{contours=>pdl(-1,0,1),labels=>['-1','0','1']};

```



**მაგალითი 4:**

**3D გრაფიკული ობიექტები:**

ფერები თან ახლავს ზემოთ აგებული Gaussian იმიჯის თითოეულ დონეს და მათი შეცვლა შეიძლება ctab ბრძანებებით.

```
Perldl>$x=zeros(30,30)->xlinvals  
(-0,5,05);  
Perld>$y=(zeros(30,30)->ylinvals  
(-0,5,0,5);  
perldl>$z=exp(-($x**2)/0.05-($y**2)/0.02);
```

```
perldl>line3d[$x,$y,$z],[LineWidth=>2];  
perldl>$levels=zeros(11)->xlinvals(0,1)  
perl>p$levels  
[00.10.20.30.40.50.60.70.80.91]  
perldl>$red=zeros(11)->xlinvals(1,0)  
perldl>$gren=zeros(11)->xlinvals(1,0)  
perldl>$blue=zeros(11)->xlinvals(0,1)
```

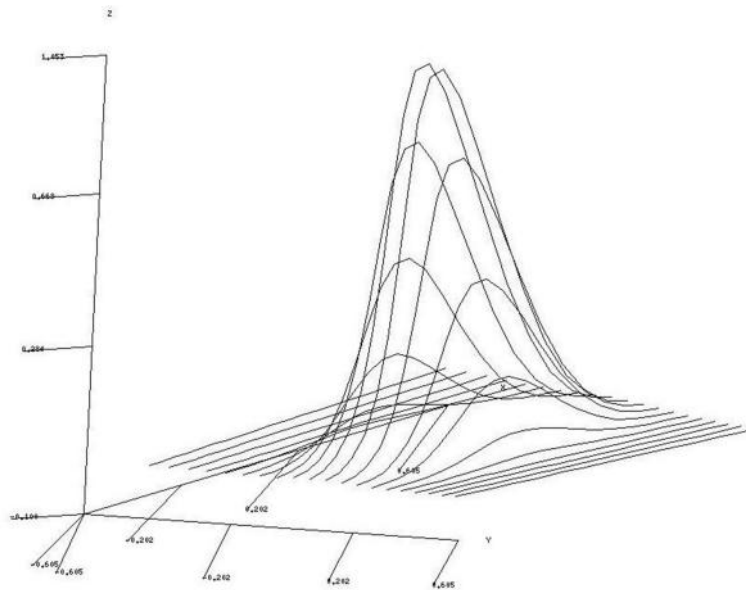
აღნიშნეთ უმკვეთრესი წითელი ფერი და უმკვეთრესი მწვანე ფერი პირველი მნიშვნელობით და ლურჯი ფერი კინკვერსიულად.

ფერის ცხრილის შესაცვლელად:

```
perldl>ctab($levels,$red,$green,$blue)
```

იმიჯის ეკრანზე ხელახლა გამოსატანად:

```
perldl>line3d[$x,$y,$z},{LineWidth=>2}
```





**ნაწილი 4** წარმოდგენილია PDL-შივექტორული ანალიზი და ამ საკითხის შედარება MAXIMA-სა და C პროგრამირების ენასთან.

ჩვენ განვსაზღვრავთ მის უპირატესობებსა და გამოტოვებებს (მისინგების) ასპექტს.

და ბოლოს, ჩვენ მოვამზადებთ ვექტორული ანალიზის ინსტრუმენტს PERL TOOL KiT-ის გამოყენებით.

### **ნაშრომის სამიზნე:**

თეზისები ინჟინრებსა და მკვლევარებს დაეხმარებათ მეცნიერებაში მათემატიკური გამოთვლების თვალსაზრისით ვრცელი მონაცემთა ბაზების მოსახერხებელი მართვის წარმოებაში. ანალიტიკური მეთოდები და ინტერპრეტაციის შედეგები მაღალი ეფექტურობით გამოირჩევა. შერჩეული მაგალითები ილუსტრირებულია პრაქტიკული შემთხვევების საფუძველზე.

### **დისერტაციის თემაზე გამოქვეყნებულ ნაშრომთა სია**

- Mustafa im ek, Basic of Perl IKECCO2009, 6 International Conference on Electronics and Computer in Kyrgyzstan-Kazakhstan, 12-13 October 2009 in Almaty, Kazakhstan, p.64-66
- Mustafa im ek, Perl Programming Input-Output Files, 3<sup>th</sup> IEEE International Conference on Electronics and Computer 2009, 14-16 October 2009 in Baku, Azerbaijan, p.338-341
- Mustafa im ek, PERL Programming Functions, IKECCO2009, 6 International Conference on Electronics and Computer in Kyrgyzstan-Kazakhstan, 12-13 October 2009 in Almaty, Kazakhstan, p.74-77
- Niyazi ARI, Prof.Dr.sc. tech., ETH, HSZ-T, Mustafa M EK, Int. Ataturk Alatau University, Bishkek, Kyrgyzstan, Scientific Programming with PERL, Jahresbericht aF&E 2009, p.47-50
- Mustafa im ek, Basic of PDL, Alatau Academic Studies, V4-N2-2009 – Kyrgyzstan, p.174-179

- Niyazi ARI, Prof.Dr.sc. tech. ,ETH,HSZ-T, Mustafa im ek, Introduction to Bioperl for Bioinformatics, IKECCO2010, 7 International Conference on Electronics and Computer in Kyrgyzstan-Kazakhstan, November 1, 2010 in Bishkek, Kyrgyzstan, p.6-10
- Mustafa im ek, Perl&Operating System,International Scientifically-Practical Conference, KGUSTA, 1-2 June 2011, Kyrgyzstan, p.96-102
- Mustafa im ek,Perl Object Oriented Programming, AICT2011 V International Conference, 12-14 October 2011, Baku-Azerbaijan, p.599-602
- Mustafa im ek, Perl Graphic Applications with Image Processing, IBSU Scientific Journal, 2012, 1(2) (would be published)