

ბ. ჯანელიძე, ლ. პეტრიაშვილი, ბ. მეფარიშვილი

ობიექტზე ორიენტირებული დაპროგრამება

(VBA ენის ბაზაზე)

**თბილისი
2012**

უაკ 681.3.06

წიგნში წარმოდგენილია ობიექტზე ორიენტირებული დაპროგრამება VBA ენის ბაზაზე. წიგნი გათვალისწინებულია როგორც დამწყებთათვის, ვინც არ ფლობს პროგრამირების სხვა ენებს, ასევე მათთვის ვინც ფლობს რომელიმე დაპროგრამების ენის საფუძვლებს.

წიგნი განკუთვნილია ინფორმატიკის სფეროს სტუდენტებისათვის.

რეცენზენტი ასოცირებული პროფესორი ნ. ოთხოზორია

ISBN 978-9941-0-4129-7

ყველა უფლება დაცულია. ამ წიგნის არც ერთი ნაწილი (იქნება ეს ტექსტი, ფოტო, ილუსტრაცია თუ სხვა) არანაირი ფორმით და საშუალებით (იქნება ეს ელექტრონული თუ მექანიკური), არ შეიძლება გამოყენებულ იქნას გამომცემლის წერილობითი ნებართვის გარეშე.

საავტორო უფლებების დარღვევა ისჯება კანონით.

შინაარსი

შესავალი.....	5
თემა №1	
ობიექტზე ორიენტირებული დაპროგრამების VBA ენის გაცნობა.....	7
თემა №2	
პროგრამული კოდის განხილვა მაკროსის მაგალითზე	22
თემა №3	
პროგრამული კოდში ფუნქციების გამოყენება	35
თემა №4	
სამომხმარებლო ფორმებთან მუშაობა	49
თემა №5	
Image ობიექტთან მუშაობა	70
თემა №6	
MsgBox და InputBox ობიექტებთან მუშაობა	89
თემა №7	
ენის ალგორითმული კონსტრუქციები.....	108
თემა №8	
Excel – ის მუშა ფურცელზე ინფორმაციის დამუშავების ალგორითმები.....	120
თემა №9	
ამოცანების ამოხსნა ალგორითმული სტრუქტურების გამოყენებით.....	134
თემა №10	
მატრიცებთან მუშაობა	145

თემა №11	
მატრიცებთან მუშაობა-2.....	152
თემა №12	
VBA გამოყენება Word დანართისათვის.....	157
თემა №13	
ტექსტებთან მუშაობა	166
თემა №14	
VBA ენა PowerPoint დანართისათვის	170
თემა №15	
ტესტირების სისტემის დაპროგრამება	181
ლიტერატურა	187

შესავალი

კომპიუტერული ტექნოლოგიების სამყაროში არსებობს მრავალი განსხვავებული სტრუქტურის და შესაძლებლობის პროგრამირების ენა, რომელიც შეიძლება კლასიფიცირებულ იქნას სხვადასხვა ნიშნის მიხედვით. ენის და შესაბამისად პროგრამირების სისტემის მთავარ კვალიფიციურ ნიშნად ითვლება პროგრამირების თანამედროვე სტილის უზრუნველყოფა.

პროგრამირების ენები შეიძლება დაიყოს: პროცედურულ, ფუნქციონალურ, ლოგიკურ, ობიექტზე ორიენტირებულ ენებად. პროგრამირების სისტემების წამყვანი დამმუშავებელია ფირმები: Microsoft და Borland International.

დაპროგრამების პროცედურულ ენაზე პროგრამა შედგება ოპერატორების მიმდევრობისაგან, ეს ინსტრუქციებია, რომლებშიც მოცემულია ესა თუ ის მოქმედება. პროცედურული ენის საკვალიფიკაციო ნიშნად ითვლება მანქანურ ენასთან მიახლოება. Basic ენა (Beginner's All-purpose Symbolic Instruction Code – სიმბოლიკური ინსტრუქციების მრავალმიზნობრივი ენა დამწყებთათვის) დამუშავებულ იქნა 1964 წელს დამწყებთათვის გამოსაყენებლად. თავდაპირველად მუშაობა მიმდინარეობდა ინტერაქტიული ინტერპრეტაციის რეჟიმში. დაპროგრამების Basic ენა პერსონალურ კომპიუტერებზე ძალიან პოპულარულია. არსებობს მისი მრავალი დიალექტი. Basic ენის თანამედროვე დიალექტები საკმაოდ განვითარებულია და ნაკლებად ჰგავს თავის წინაპრებს.

ობიექტზე ორიენტირებული ენისათვის პირველადი არის არა დამოკიდებულება, არამედ ობიექტი. პროგრამირების ეს სტილი ხასიათდება განვითარებული მოდულური სტრუქტურით. სწორედ მოდული ამარტივებს რთული პროგრამული პროდუქტების დამუშავებას.

ობიექტზე ორიენტირებული დაპროგრამების მაგალითად შეიძლება დასახელებულ იქნას **Visual Basic**.

კომპიუტერების სიმკლავრის გაზრდასთან ერთად შეიქმნა გაცილებით სრულყოფილი პროგრამული უზრუნველყოფა. **Microsoft** ფირმამ **MS Office** პაკეტში ჩანერგა ჩაშენებული პროგრამული დანართის სახით **Visual Basic** სისტემა, რომელიც გაცილებით სრულყოფილი და ინტეგრირებულია მომხმარებელთა ფართო სპექტრისათვის და ამ პლატფორმას უწოდა **VBA (Visual Basic for Application)**. მას ახასიათებს გამოყენების სიმარტივე და ამოცანათა ფართო სპექტრის შესრულების უნარი.

დღევანდელ პერიოდში **VBA** წარმოადგენს მძლავრ პროგრამულ დანამატს ისეთ პაკეტებზე როგორცაა: Microsoft Office – Word, Excel, PowerPoint, Access, Outlook, FrontPage, Microsoft Project, Corel WordPerfect Office, AutoCAD, Autodesk Map, TurboCAD Professional და სხვა.

VBA სისტემის შესაძლებლობების ნუსხა ფანტასტიურია, რამდენადაც თვით მომხმარებელი ქმნის თავისთვის პროგრამას ან ცვლის უკვე არსებულ რესურსებს.

წიგნი შეიცავს სხვადასხვა სფეროდან აღებული მაგალითების დიდ რაოდენობას, რათა სტუდენტს ან სხვა მომხმარებელს მიეცეს უდაოდ საინტერესო და ცხოვრებისეული ამოცანების გადაწყვეტის შესაძლებლობა.

თემა №1. ობიექტზე ორიენტირებული დაპროგრამების VBA ენის გაცნობა

1. VBA Excel ენის ძირითადი ცნებები
მეცადინეობაზე განიხილება შემდეგი საკითხები:

- VBA ენის ზოგადი ცნება;
- VBA-ს ობიექტები;
- თვისებები, მეთოდები და ხდომილებები;
- VBA ენის ელემენტები;
- VBA რედაქტორის სტრუქტურა.

VBA მიეკუთვნება ობიექტზე ორიენტირებული დაპროგრამების ენებს, ამდენად მისი გაცნობა მიზანშეწონილია დავიწყოთ ობიექტის ცნებით.

ობიექტი

ობიექტი VBA Excel ენის ძირითადი ძირითადი ელემენტია. VBA-ს ობიექტებს მიეკუთვნება მუშა წიგნი, მუშა ფურცელი და მისი შემადგენლობა.

ობიექტების მაგალითებია:

- Sheet – Excel-ის ფურცელი;
- Cell – უჯრა;
- Range – უჯრების დიაპაზონი;
- Application – დანართი;
- UserForm – სამომხმარებლო ფორმა.

ობიექტებზე წვდომა ხორციელდება მეთოდებისა და თვისებების მეშვეობით.

მეთოდი

ობიექტებზე შეიძლება შესრულდეს სხვადასხვა მოქმედება. მოქმედებას, რომელიც სრულდება ობიექტზე ჰქვია მეთოდი. მაგალითად, უჯრა გასუფთავდეს -Clear,

დანართი დაიხუროს – Quit, გამოჩნდეს სამომხმარებლო ფორმა – Show, დაიმალოს სამომხმარებლო ფორმა – Hide.

მეთოდის დასახელება ობიექტის დასახელებიდან გამოიყოფა წერტილით: ობიექტი.მეთოდი.

განვიხილოთ მეთოდების გამოყენების მაგალითები:

Range("B2:E2").Select – ამორჩეულ იქნას უჯრების

დიაპაზონი B2:E2;

Range("C1:C5").Clear – გასუფთავდეს უჯრების

დიაპაზონი C1:C5;

UserForm2.Hide – დაიმალოს ფორმა № 2;

UserForm5.Show – გამოჩნდეს ფორმა № 5;

Application.Quit – გამოსვლა დანართიდან.

თვისება

თვისებებით აღიწერება ობიექტების მახასიათებლები. მაგალითად, შრიფტის ზომა ან ფერი, ფორმის განთავსება ეკრანზე ან ობიექტის მდგომარეობა, როგორცაა წვდომადობა, ხილვადობა. ობიექტის მახასიათებლის შესაცვლელად უნდა შეიცვალოს თვისების მნიშვნელობა, ანუ მას მიენიჭოს განსაზღვრული მნიშვნელობა.

თვისების მნიშვნელობის მინიჭების სინტაქსი:

ობიექტი.თვისება = თვისების მნიშვნელობა;

სადაც ობიექტი არის ობიექტის სახელი, თვისება – თვისების სახელი, რომელსაც ენიჭება მნიშვნელობა.

ობიექტის სახელი თვისების სახელიდან გამოიყოფა წერტილით.

განვიხილოთ თვისებების მაგალითები:

Range("D1").Value=200 - D1 უჯრაში განთავსდეს ნიშნელობა 200.

Range("C1:C10").Text = "ინფორმატიკა" - C1:C10 უჯრების დიაპაზონში განთავსდეს ტექსტი "ინფორმატიკა".

Range("B2").Font.Size = 14 - B2 უჯრაში შრიფტის ზომა იყოს 14.

VBA ენის ზოგიერთი ობიექტის მეთოდები და თვისებები

ობიექტი: Application (დანართი).

მეთოდი: Quit (დაიხუროს).

თვისება: Caption (მთავარი ფანჯრის სახელი).

მაგალითები:

Application.Quit – დანართი დაიხუროს;

Application.Caption = "უწყისი" – დანართის ფანჯარას დაერქვას "უწყისი".

ობიექტი: Sheet (ფურცელი), ActiveSheet (აქტიური ანუ ამორჩეული ფურცელი).

ოჯახი: Sheets (ფურცლები).

მეთოდები: Select (ამორჩევა); ShowDataForm (გამოხსნის ჩადგმული ფორმა).

მაგალითები:

Sheets("მენიუ").Select – ამორჩეულ იქნას ფურცელი "მენიუ";

ActiveSheet.ShowDataForm – მოცემულ მომენტში აქტიურ ფურცელზე ნახვენები იქნას ჩაშენებული ფორმა.

ობიექტი: Range(დიაპაზონი).

მეთოდი: Select (გამოვეყოთ/მოვნიშნოთ); Clear

(გასუფთავდეს).

თვისება: Name(სახელი).

მაგალითები:

Sheets("უწყისი").Range("B4:B10").Name = "კლასი" -

"უწყისი" ფურცელზე განთავსებულ B4:B10 დიაპაზონს მიენიჭოს სახელი "კლასი";

Sheets("უწყისი").Range("B4:B10").Select - "უწყისი"
ფურცელზე გამოვეყენოთ B4:B10 დიაპაზონი.

ობიექტი, ოჯახი: UserForm(სამომხმარებლო ფორმა).

მეთოდი: Show(გამოჩენა); Hide(დამალვა).

თვისება: Caption(ტექსტი სათაურის სტრიქონში).

მაგალითები:

UserForm1.Show - გამოჩნდეს სამომხმარებლო ფორმა
ნომრით პირველი;

UserForm1.Hide - დაიმალოს სამომხმარებლო ფორმა
ნომრით პირველი;

UserForm1.Caption="ინფორმატიკა"-სათაურის
სტრიქონში გამოჩნდეს ბრჭყალებში მოცემული სიტყვა.

ობიექტი, ოჯახი: TextBox(შეტანის ველი).

თვისება: Text(შინაარსი).

მაგალითები:

UserForm1.TextBox1.Text = Date - პირველ
სამომხმარებლო ფორმის ნომრით პირველ შეტანის
ველში ჩაიწეროს მიმდინარე თარიღი;

UserForm1.TextBox2.Text = " " - პირველ სამომხმარებლო
ფორმის ნომრით პირველი შეტანის ველი გასუფთავდეს.

ობიექტი, ოჯახი: ComboBox(ველი სიით).

მეთოდი: AddItem(ელემენტის დამატება სიაში).

თვისება: Text (ჩაწერილი ტექსტი); Rowsource
(სტრიქონების წყარო სიისათვის).

მაგალითები:

UserForm1.ComboBox2.Text = " " - პირველ სამომხმარებლო
ფორმის მეორე შეტანის ველის გასუფთავება;

UserForm2.ComboBox1.Rowsource="B2:B10" - მეორე
სამომხმარებლო ფორმის პირველი სიებიანი შეტანის
ველისათვის სტრიქონების წყაროდ ჩაითვალოს მონაცემები
B2:B10 დიაპაზონიდან;

UserForm1.ComboBox1.AddItem ("ფაქსი") - ComboBox1 –ის სიაში დაემატოს ელემენტი სახელით “ფაქსი“.

ობიექტი, ოჯახი: OptionButton.

თვისება: Value(მნიშვნელობა); Caption(წარწერა).

მაგალითები:

UserForm3.OptionButton1.Value = True – მესამე სამომხმარებლო ფორმაზე ავირჩიოთ გადამრთველი ნომერი პირველი;

UserForm3.OptionButton1.Capture = "მოსწრება" – მესამე სამომხმარებლო ფორმის გადამრთველთან დავაყენოთ წარწერა "მოსწრება".

ობიექტი, ოჯახი: CheckBox.

თვისება: Value(მნიშვნელობა); Caption(წარწერა).

მაგალითები:

UserForm2.CheckBox1.Value = True – მეორე სამომხმარებლო ფორმაზე აღემაართოთ ალამი ნომრით პირველი;

UserForm3.CheckBox1.Value = False – მესამე სამომხმარებლო ფორმაზე ჩამოვაგდოთ ალამი ნომრით პირველი;

UserForm4.CheckBox1.Capture="კლასი" - მესამე სამომხმარებლო ფორმის დროშასთან გავაკეთოთ წარწერა “კლასი“.

ხდომილებები

ხდომილება წარმოადგენს ქმედებას, რომელიც უნდა ამოიცნოს ობიექტმა. VBA ენაში ცალკეული ობიექტისათვის განსაზღვრულია სტანდარტული ხდომილებების ნაკრები. ობიექტისათვის „დილაკი“ (CommandButton) სტანდარტული ხდომილება არის თავის დაწკაპუნება (**Click**). მომხმარებლის მიერ დილაკზე დაჭერა უკვე ხდომილებაა. ხდომილებას უნდა ჰქონდეს გამოძახილი, რომელიც ნიშნავს

რაიმე პროგრამის შესრულებას. ასეთ პროგრამას ჰქვია ხდომილების დამუშავების პროცედურა და აქვს სტანდარტული სახელი. თუ ასეთი გამოძახილი არ არის შექმნილი, ანუ არ არის დაწერილი შესაბამისი პროგრამა, სისტემა ხდომილებაზე რეაგირებას არ გააკეთებს.

VBA ენის ელემენტები

ობიექტი VBA ენის ძირითადი ელემენტია, მაგრამ არა ერთადერთი. სხვა ელემენტებს მიეკუთვნება კონსტანტები, ცვლადები, მასივები, გამოსახულებები, ჩადგმული ფუნქციები, ჩადგმული დიალოგური ფანჯრები, ოპერატორები.

კონსტანტა – მონაცემია, რომელიც ამოცანის ამოხსნის პროცესში არ იცვლის მნიშვნელობას. კონსტანტა შეიძლება იყოს რიცხვითი და სიმბოლური.

რიცხვითი კონსტანტა – მთელი ან ნამდვილი რიცხვია.

სიმბოლური კონსტანტა – ტექსტია, ჩასმული ბრჭყალებში.

ცვლადი – მონაცემია, რომლის მნიშვნელობა იცვლება პროგრამის მსვლელობის პროცესში. ცვლადისათვის საჭიროა სახელის და ტიპის მიცემა.

ცვლადის სახელი უნდა იწყებოდეს ასოთი და შეიძლება შეიცავდეს ციფრებსა და ზოგიერთ სიმბოლოს. ცვლადის სახელი არ უნდა შეიცავდეს წერტილს, პრობელს და შემდეგ სიმბოლოებს: “!”, “%”, “&”, “\$”, “#”, “@”. სახელის სიგრძე არ უნდა აღარბებდეს 255 სიმბოლოს. სახელი არ უნდა ემთხვეოდეს ენის გასაღებურ სიტყვებს, ჩაშენებულ ფუნქციებს და პროცედურებს.

ცხრილში წარმოდგენილია ცვლადის ძირითადი ტიპები, მათი ზომები და მნიშვნელობათა დიაპაზონები.

ტიპი	ზომა(ბაიტებში)	მნიშვნელობათა დიაპაზონი
Byte (ბაიტი)	1	0 - 255
Boolean (ლოგიკური)	2	True/False
Integer (მთელი რიცხვი)	2	-32768 - 32767
Long (გრძელი მთელი რიცხვი)	4	-2 147 483 648 – 2 147 483 647
Single(რიცხვი მცოცავი მძიმით)	4	-3,4E38 –დან -1,4E-45-მდე 1,4E-45-დან 3,4E38-მდე
Double(რიცხვი მცოცავი მძიმით ორმაგი სიზუსტის)	8	-1,79 E308–დან -4,94E-342 4,94E-324–დან 1,79 E308-მდე
Currency (ფულადი)	8	922 337 203 685 477,5808-დან 922 337 203 685 477,5807-მდე
Date (თარიღი და დრო)	8	01.01.100-დან 31.12.9999-მდე
String(სტრიქონი)	10+სტრიქონის სიგრძე	0-დან 2×10^9
Variant(ვარიანტი)		დამოკიდებულია ცვლადის შემადგენლობაზე

ვარიანტის ტიპის ცვლადი ინახავს იმდენ ცვლადს, რამდენსაც მასში მოათავსებენ.

მასივი – ერთნაირი ტიპის ცვლადების მოწესრიგებული ერთობლიობაა. მასივს აქვს ზომა და სახელი. მასივის სახელი ითვალისწინებს იგივე წესებს, რასაც ზოგადად ცვლადის სახელი. მასივის ზომა მასივის შემადგენელი ელემენტების რაოდენობაა.

გამოსახულება შედგება კონსტანტების, ცვლადების, ჩადგმული ფუნქციებისა და არითმეტიკული ოპერაციის ნიშნებისაგან. კერძო შემთხვევაში გამოსახულება შეიძლება იყოს მხოლოდ ერთი ელემენტი, ანუ კონსტანტა, ცვლადი ან მიმართვა ჩადგმულ ფუნქციაზე.

გამოსახულების მაგალითები:

1. Z
2. $(a+b)^2$
3. 45
4. $\sin(y)$

VBA ენაში არის სხვადასხვა კატეგორიის ჩაშენებული ფუნქციები, ესენია:

- მათემატიკური ფუნქციები;
- ფორმატის გარდაქმნის ფუნქციები;
- ლოგიკური ფუნქციები;
- თარიღისა და დროის ფუნქციები;
- ტიპების შემოწმების ფუნქციები;

ზოგიერთი ფუნქცია მოყვანილია ცხრილში

ფუნქცია	გამოსახულება
	მათემატიკური ფუნქციები
Abs(x)	მოდული
Cos(x)	კოსინუსი
Exp(x)	ექსპონენტა
Log(x)	ნატურალური ლოგარითმი
Sin(x)	სინუსი
Sqr(x)	კვადრატული ფესვი
	ლოგიკური ფუნქციები
x1 and x2	ლოგიკური გამრავლება
x1 or x2	ლოგიკური შეკრება
	ტიპების შემოწმების ფუნქციები
IsNumeric(ცვლადი)	True, თუ ცვლადი რიცხვია False, თუ ცვლადი არ არის რიცხვი
IsNull(ცვლადი)	True, თუ ცვლადი არის ცარიელი მნიშვნელობის False, თუ ცვლადი არ არის ცარიელი მნიშვნელ.
	ფორმატების გარდაქმნის ფუნქციები
CInt(x)	x გარადაიქმნება მთელად
CSng(x)	x გარადაიქმნება ჩვეულებრივი სიზუსტის მცოცავწერტილიან რიცხვად
	დროისა და თარიღის ფუნქციები
Date	მიმდინარე თარიღი
Now	მიმდინარე თარიღი და დრო

ჩამოთვლილი ფუნქციების გარდა Applications ობიექტი მუშა ფურცლის 400-ზე მეტი ჩაშენებული ფუნქციის გამოძახების საშუალებას იძლევა შემდეგი კონსტრუქციის მეშვეობით:

Application.მუშა ფურცლის ფუნქცია(არგუმენტი).

მაგალითები:

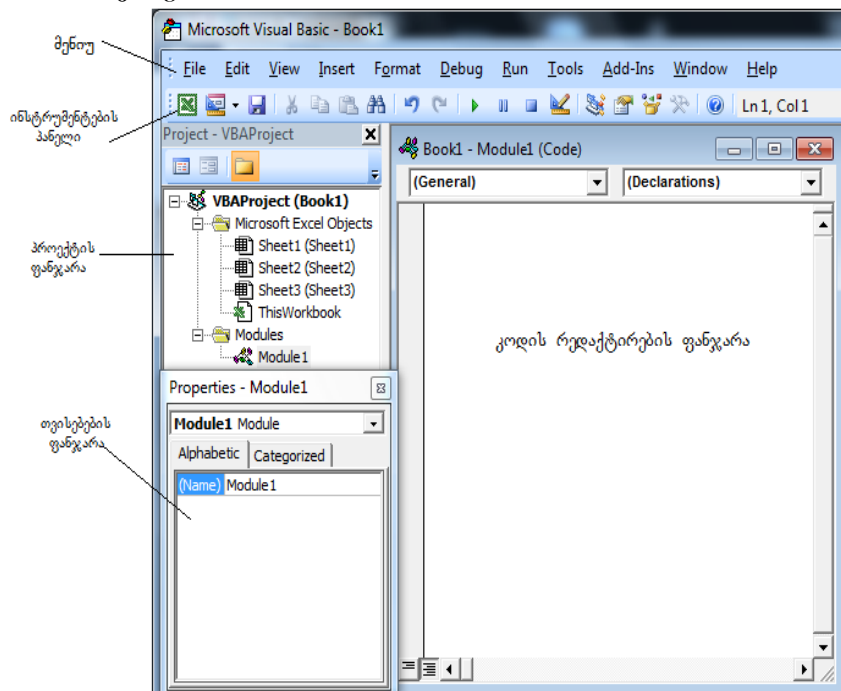
Application.Sum(Sheets("შემოწმება").Range("A1:B20")) – ჯამდება "შემოწმება" ფურცელზე განთავსებული A1:B20 უჯრების მნიშვნელობები.

Application.CountA(Sheets("სტუდენტები").Range("A:A")) – დაითვლება „სტუდენტები“ ფურცელზე, A სვეტში განთავსებული არაცარიელი უჯრების რაოდენობა.

VBA რედაქტორის სტრუქტურა

VBA რედაქტორი აქტიურდება ბრძანებით Developer/Visual Basic ანუ ბრძანების შესრულების შემდეგ ჩვენ აღმოვჩნდებით VBA რედაქტორში. მუშა წიგნში დაბრუნება შეიძლება View მენიუში Microsoft Excel ღილაკით. განვიხილოთ რედაქტორის ფანჯრის ძირითადი კომპონენტები. რედაქტორის ფანჯარა შედგება შემდეგი კომპონენტებისაგან:

- ინსტრუმენტების პანელი;
- პროექტის ფანჯარა;
- თვისებების ფანჯარა;
- კოდის რედაქტირების ფანჯარა;
- მენიუ.



ინსტრუმენტების პანელი

Visual Basic რედაქტორის ინსტრუმენტების სტანდარტული პანელი შეიცავს დილაკებს, რომელთა დანიშნულებაა ხშირად გამოსაყენებელი ბრძანებების შესრულება. პანელი დაყოფილია ცალკეულ სეგმენტებად შესასრულებელი ბრძანების ტიპის მიხედვით.



პირველი სეგმენტის დილაკები:



დილაკი Excel-ში დასაბრუნებლად, პროექტში ელემენტების ჩასმა (სამომხმარებლო ფორმა, მოდული, პროცედურა), მუშა წიგნის შენახვა.

მეორე სეგმენტის დილაკები:



დილაკები კოდის ფრაგმენტის ამოჭრის, კოპირების, ჩასმის და ძეზისათვის.

მესამე სეგმენტის დილაკები:



მოქმედებების გაუქმება, გაუქმებული მოქმედებების განმეორება.

მეოთხე სეგმენტის დილაკები:



პროცედურის შესრულების, შეჩერების, შეწყვეტის, ასევე, სამომხმარებლო ფორმის ასახვის რეჟიმის შეცვლის დილაკები.

მეხუთე სეგმენტის დილაკები:




პროექტის, თვისებების და ობიექტების დათვალიერების ფანჯრების, ასევე, ინსტრუმენტების პანელის ასახვის დილაკები.



საცნობარო სისტემის გამოძახების დილაკი.


პროექტის ფანჯარა

პროექტის ფანჯარა აქტიურდება View -> Project

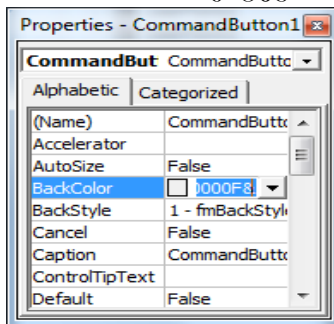
Explorer ბრძანებით ან  დილაკზე დაჭერით. პროექტის ფანჯარაში წარმოდგენილია მიმდინარე პროექტის ფაილების, ფორმების და მოდულების იერარქიული სტრუქტურა.

თვისებების ფანჯარა


თვისებების ფანჯარაში ჩამოთვლილია არჩეული ობიექტის თვისებების ძირითადი მნიშვნელობები. აღნიშნული ფანჯრის გამოყენებით შეიძლება თვისებების დათვალიერება და მათი მნიშვნელობების შეცვლა. არჩეული ობიექტის თვისებების დათვალიერებისათვის უნდა შევასრულოთ ბრძანება (View -> Properties Window ან

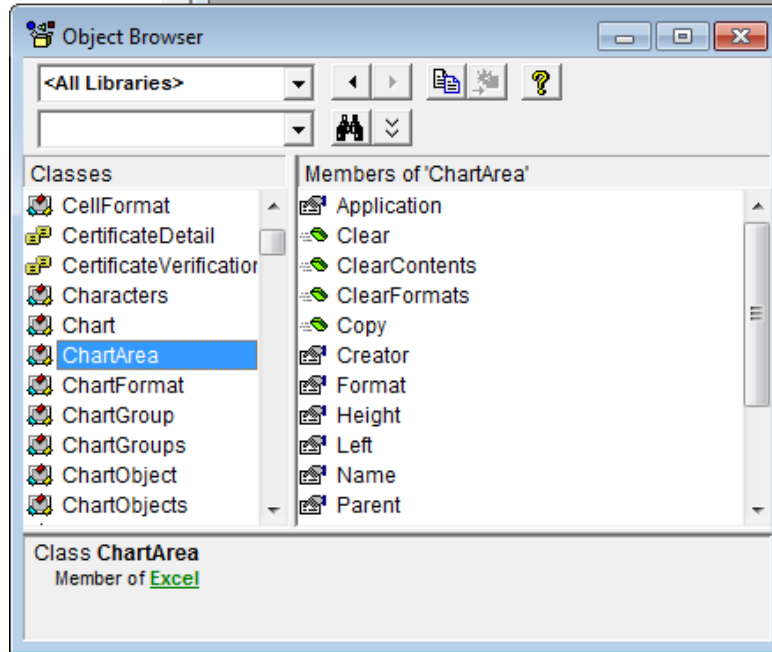
დავაჭიროთ  დილაკზე.

თვისებების ფანჯარა შედგება ორი შედგენილი ნაწილისაგან, ესენია: ზედა და მუშა ნაწილები. ფანჯრის ზედა ნაწილში განთავსდება სია, საიდანაც შეიძლება ავირჩიოთ მიმდინარე ფორმის ნებისმიერი მართვის ელემენტი ან თვით ფორმა. მუშა ნაწილი შედგება ორი ჩანართისაგან: Alphabetic და Categorized, რომლებშიც თვისებები განთავსდება ანბანის ან კატეგორიის მიხედვით.



ობიექტების დათვალიერების ფანჯარა (Object Browser)

აღნიშნული ფანჯრის გამოძახება შეიძლება View-> Object Browser ბრძანებით, ან  ღილაკზე დაჭერით. გამოსულ ფანჯარაში შეიძლება დავათვალიეროთ პროექტის ყველა ობიექტი. აქ იპოვით ყველა თვისებას, მეთოდს და ხდომილებას, რომელიც დაკავშირებულია ნებისმიერ ობიექტთან.



შეასრულეთ მოქმედებათა თანმიმდევრობა:

1. გახსენით ახალი ფაილი.
2. გადადით VBA რედაქტორში.
3. დააჭირეთ F2 ღილაკს Object Browser ფანჯრის გახსნისათვის.

4. Classes სიაში იპოვეთ Range ობიექტი და მონიშნეთ იგი.
5. გადაახვიეთ მეზობელი სია Members of 'Range' , Range ობიექტის მეთოდების და თვისებების დასათვალიერებლად.
6. აირჩიეთ Activate მეთოდი.
7. დააჭირეთ F1 ღილაკს, გამოვა საცნობარო სისტემა Activate მეთოდის აღწერით.
8. გახსენით Range ობიექტის მიმართ Activate მეთოდის გამოყენების მაგალითი.
9. დაწერეთ მეთოდი, რომელიც გამოიყენება უჯრების დიაპაზონის შემადგენლობის წასაშლელად.
10. დაწერეთ მეთოდი, რომელიც გამოიყენება წიგნში მუშა ფურცლის წასაშლელად.
11. პროექტის ფანჯარაში აირჩიეთ ფურცელი1. თვისებების ფანჯარაში შეუცვალეთ სახელი და დაარქვით „სავარჯიშო“.
12. პროექტის ფანჯარაში ფურცელი2. აირჩიეთ თვისება, რომელიც ფურცელს უხილავს გახდის.
13. აირჩიეთ ფურცელი3. აირჩიეთ თვისება, რომელიც რომელიც შეცვლის სვეტის სიგანეს. მიუთითეთ ზომა-15.

თემა №2. პროგრამული კოდის განხილვა მაკროსის მაგალითზე

სამუშაოს დანიშნულებაა შემდეგი საკითხების შესწავლა:

1. მაკროსის ცნება
2. მარტივი მაკროსის ჩაწერა
3. მაკროსის გაშვება შესრულებაზე
4. მაკროსის რედაქტირება
5. მაკროსის წარმოდგენა გრაფიკული გამოსახულებით
6. მაკროსის ამოშლა ჩამონათვალიდან
7. მაკროსის შესაძლებლობათა შეზღუდულობა

მაკროსის დანიშნულება

მაკრორეკორდერი – არის სტანდარტული ფუნქციური ლილაკი რათა **Excel** პროგრამაში მოხდეს მაკროსის ჩაწერა. მაკროსის ჩაწერის საფუძველზე შესაძლებელია მომხმარებლის მიერ შესრულებულ მოქმედებათა თანამიმდევრობით ჩაწერა და შესაბამისი VBA პროგრამული კოდის მიღება. პროგრამა, რომელიც მიიღება მაკრორეკორდერის საშუალებით ეწოდება მაკროსი.

მაკროსის ძირითადი დანიშნულებაა მომხმარებლის მიერ შესრულებული სამუშაოს ავტომატიზაცია. გარდა ამისა შესაძლებელია არსებული მაკროსის კოდის შემგდგომი დამუშავება და სრულყოფა. როგორც აღვნიშნეთ მაკროსის ჩაწერის დროს ხდება მომხმარებლის მიერ შესრულებული ყველა მოქმედების ეტაპობრივი ჩაწერა, რომლის გასააქტიურებლად საკმარისია კლავიატურიდან შესაბამისი კლავიშების გამოყენება ან მენიუდან შესაბამისი ბრძანების გააქტიურება, რის შედეგადაც პროგრამული კოდი ავტომატურად გარდაიქმნება **VBA** ენაზე.

ყველა მაკროსს აქვს თავისი ინდივიდუალური სახელი, მაკროსის სწრაფი გაშვებისათვის აუცილებელია მას მივანიჭოთ გამშვები ლილაკის ფუნქცია. მაკროსის

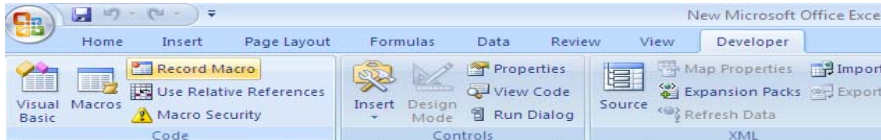
გაშვების შემდეგ ავტომატურად შესრულება ის დანართი, რომელიც შეიქმნა მაკროსის ჩაწერისას.

მაკროსი არის მომხმარებლის მიერ წარმოდგენილი ბრძანებათა თანამიმდევრული შესრულება, რომელიც ინახება პროგრამული სახით VBA ენაზე

მაკროსის ჩაწერის პროცედურა Excel დანართში

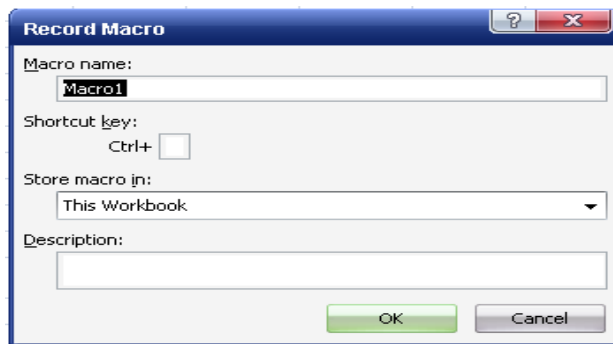
Excel დანართში მაკროსთან სამუშაოდ მთავარი მენიუდან საჭიროა შემდეგ მოქმედებათა თანამიმდევრული შესრულება:

1. მენიუდან **Developer**→**Record Macro** ბრძანების გამოძახება.



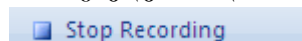
2. ჩანართის **Record Macro** გააქტიურება.

3. მიღებულ დიალოგურ ფანჯარას „მაკროსის ჩაწერა“ ჩანართში Macro name ენიჭება სახელი (პირველი სიმბოლო აუცილებელია იყოს ანბანიდან)



4. სრუდება მაკროსის ჩასაწერი მოქმედება;

5. მოქმედების დასასრულებლად აქტიურდება ლილაკი



მაგალითი

განვიხილოთ მოქმედებათა თანამიმდევრულად ჩაწერის პროცედურა შემდეგ მაგალითზე:

განსაზღვროთ უჯრათა გამოყოფილ დიაპაზონში ჩაწერილ მონაცემთა მაქსიმალური მნიშვნელობა. ცხრილში მოცემულია ინფორმაცია მსოფლიოში ყველაზე ძვირფასი ალმასების შესახებ. საჭიროა შეიქმნას მაკროსი, რომელიც განსაზღვრავს ალმასის მაქსიმალურ წონას გრამებში.

თუმცა, აღნიშნული ამოცანის გადაჭრა პროგრამა Excel-ში შესაძლებელია სტანდარტული ფუნქციის **MAX** გამოყენებითაც, მაგრამ ამ მაგალითის საშუალებით ვნახავთ, როგორ ხდება მომხმარებლის მიერ შესრულებულ მოქმედებათ თანამიმდევრულად შენახვა და ამის საფუძველზე მაკროსის შექმნა.

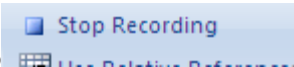
შეასრულეთ შემდეგი მოქმედებები:

1. გახსენით ახალი წიგნი;
2. შექმენით ცხრილი მოცემული მონაცემების საფუძველზე

	A	B	C	D	E
1	დასახელება	ძვეყანა (წარმოშობა)	რომის იქნა აღმოჩენილი	წონა კარატში	წონა გრამებში
2	დიდიგული მობოლი	ინდოეთი	XVII საუკუნე	787.0	157.40
3	ჰონკერი	სამხ. აფრიკა	1934	726.0	145.20
4	ლეონი	დასავ. აფრიკა	1972	968.9	193.78
5	კულინანი	სამხ. აფრიკა	1905	3106.0	621.20
6	პრუნიფენტი ვარბასი	ბრაზილია	1938	726.6	145.32
7	მდინარე შოიე	დასავ. აფრიკა	1945	770.0	154.00
8	ექსცელსიორი	სამხ. აფრიკა	1893	971.5	193.3

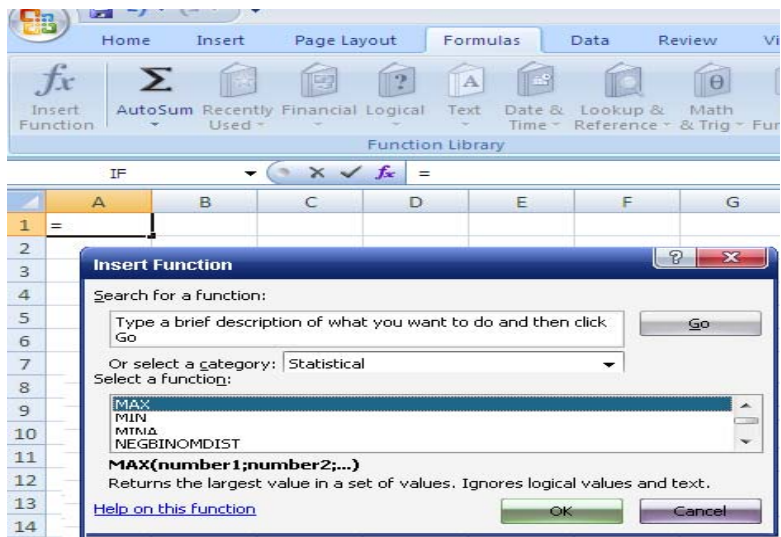
3. უჯრაში **D10** შეიტანეთ ჩანაწერი „მაქსიმალური მნიშვნელობა გრამებში“. ტექსტის შეტანის შემდეგ, კურსორი შესაძლებელია იყოს ნებისმიერ ადგილას გარდა იმ უჯრისა სადაც გამოტანილი უნდა იქნას

შედეგი. როგორც ზემოთ იყო წარმოდგენილი ვააქტიურებთ ჩანართს **Record Macro** ინსტრუმენტების პანელიდან და ვღებულობთ მაკროსის ჩაწერის დიალოგურ ფანჯარას. ჩანართში **Macro name** ვანიჭებთ **Macro 1** ის ნაცვლად სახელს „აღმასი“ და ვააქტიურებთ ღილაკს „OK“. ეკრანზე ინსტრუმენტების პანელზე

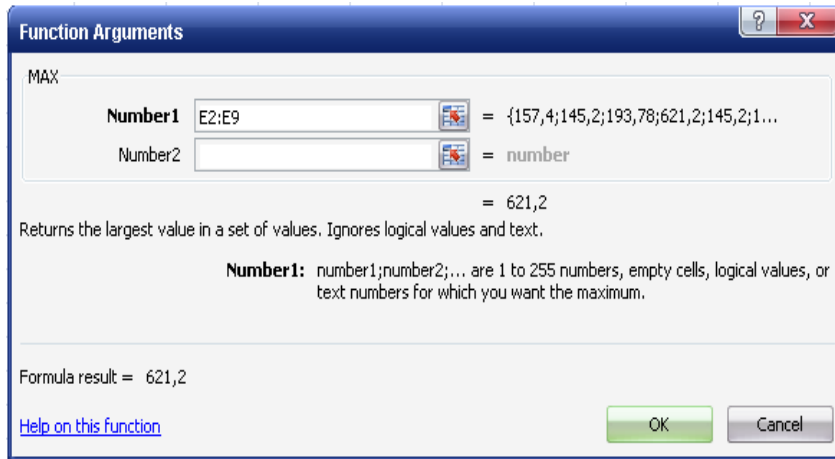


გამოჩნდება ღილაკი **Use Relative References** (ჩანაწერის გაჩერება). ახლა უკვე მოხდება ყველა შესრულებულ მოქმედებათა ჩაწერა, მანამ სანამ არ მოხდება აღნიშნული ღილაკის ვააქტიურება. (ამ დროს არ შეიძლება სხვა რაიმე მოქმედების შესრულება, რადგან მომხმარებლის მიერ შესრულებული ყველა მოქმედება ფიქსირდება და იწერება). კურსორი დააყენეთ უჯრაში **E10**, ამ უჯრაში უნდა მოხდეს შედეგის გამოტანა.

4. მთავარი მენიუდან ვააქტიურებთ ბრძანება **Formulas**→**Insert Function** საიდანაც მიიღება ფუნქციათა არჩევანის ფანჯარა, რომელიც წარმოდგენილია სურათზე:



ჩანართში ფუნქციათა კატეგორია (**Or select a category**) უნდა შეირჩეს სტატისტიკური, ხოლო ფუნქციათა ჩამონათვალში ფუნქცია - **MAX**. მიიღება დიალოგური ფანჯარა, რომლის პირველ სტრიქონში უნდა მიეთითოს იმ უჯრათა დიაპაზონი, საიდან იძებნება აღმასისი მაქსიმალურ წონას, **E2:E8**. ვაჭერთ ღილაკს **OK**.



5. ვააქტიურებთ ღილაკს „მაკროსის გაჩერება“ რის შედეგადაც მთავრდება მაკროსის ჩაწერის პროცედურა.

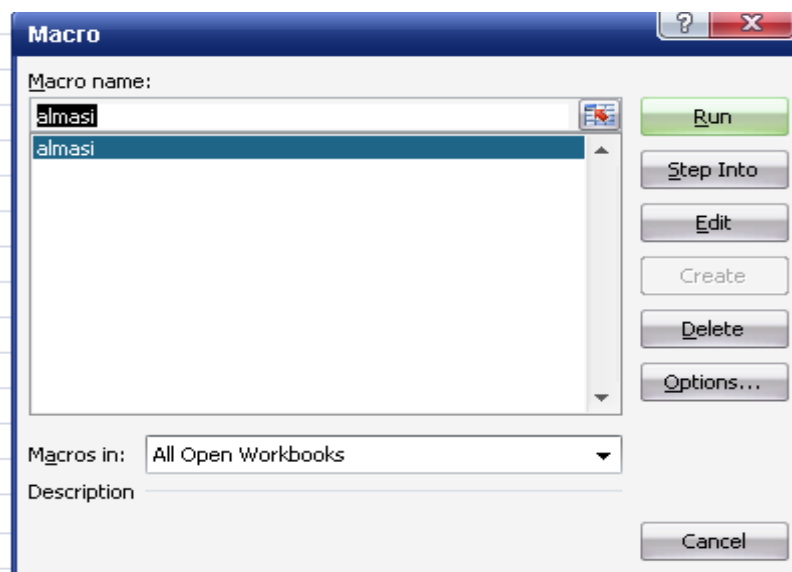
მაკროსის გაშვება შესრულებაზე

Excel პროგრამაში მაკროსის გაშვების წინ მეორდება იგივე მოქმედებები, რაც მაკროსის ჩაწერის დროს. იმისათვის რომ შემოწმდეს როგორ მუშაობს მოცემული მაკროსი, პირველ რიგში **E10** უჯრიდან უნდა წაიშალოს მიღებული შედეგი. შემდეგ კურსორი დადგეს ფურცლის ნებისმიერ



აღგილას და ვააქტიურდეს ბრძანება (**Macros**), რის შედეგადაც მიიღება დიალოგური ფანჯარა.

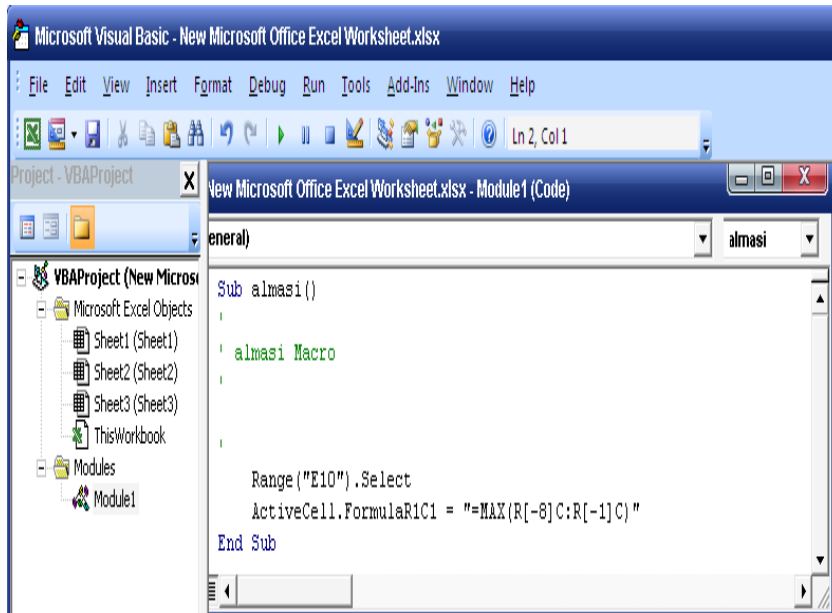
მონიშნეთ მაკროსი „ალმასი“ და გააქტიურეთ დილაკი



Run. E10 უჯრაში გამოჩნდება შედეგი.

მაკროსის რედაქტირება

მაკროსის ჩაწერისას პროგრამა **Excel** იმასსოვრებს მომხმარებლის მიერ შესრულებულ მოქმედებებს და მას გარდაქმნის **VBA** კოდში. მომხმარებელს აქვს შესაძლებლობა დაათვალიეროს მიღებული კოდი და მოახდინოს მისი რედაქტირება. კოდის დასათვალიერებლად ძირითადი მენიუს **Developer** ჩანართიდან აქტიურდება **Macros** და მიიღება წარმოდგენილი დიალოგურ ფანჯარა, სადაც ჯერ მონიშნება მაკროსი რომლის რედაქტირებაც უნდა მოხდეს, ხოლო შემდეგ აქტიურდება ჩანართი **Edit**. მიიღება სურათზე წარმოდგენილი რედაქტორის ფანჯარა:



მაკროსის ჩაწერისას გამოყენებული იყო მხოლოდ ორი მოქმედება, თავიდან კურსორი დაფიქსირდა **E10** უჯრაში, რასაც **VBA** ენაში შეესაბამება სტრიქონი: **Range("E10").Select** ხოლო შემდეგ გამოძახებული იქნა ფუნქცია **MAX** რომლის არგუმენტები მოთავსებული იყო უჯრათა დიაპაზონში **E2:E8**. **VBA** ენაში ამ მოქმედებას კი შეესაბამება სტრიქონი:

ActiveCell.FormulaR1C1 = "=MAX(R[-8]C:R[-1]C)"

მიღებული კოდი **Visual Basic** რედაქტორის დახმარებით რედაქტირდება, რაც შესაძლოა აისახოს სიმბოლოთა ზომის (**Range("E10").Font.Size=16**)

ფერის (**Range("E10").Font.ColorIndex = 3**)

და ა.შ. ცვლილებაში. მნიშვნელობა, რომელსაც ღებულობს სიმბოლოთა ფერი (**Font.ColorIndex**) განისაზღვრება 1 დან 56-მდე.

ცვლილებების შეტანის შემდეგ მაკროსის ტექსტი მიიღებს შემდეგ სახეს:

```
Sub Алмаз()  
' Almasi Макрос  
Range("E10").Select  
ActiveCell.FormulaR1C1 = "=MAX(R[-8]C:R[-1]C)"  
Range("E10").Font.Size = 16  
Range("E10").Font.ColorIndex = 3  
End Sub
```

კოდის რედაქტირების შემდეგ იხურება რედაქტორის ფანჯარა, ბრუნდებით Excel - ის ფანჯარაში, წაშლით E10 უჯრაში ჩაწერილ შედეგს და მოახდენთ მაკროსის თავიდან გაშვებას.

მაკროსის მიმაგრება გრაფიკულ გამოსახულებაზე

დანართის დამუშავებისას წინასწარ უნდა უნდა განისაზღვროს ინტერფეისის ფორმა, რათა კომფორტულად მოხდეს დასმული ამოცანის ავტომატიზაცია.

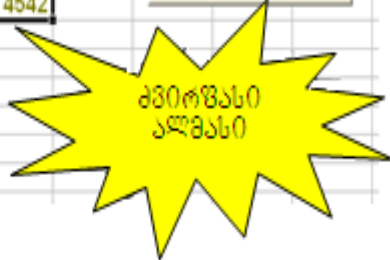
მაკროსის გასაშვებად შესაძლებელია გამოყენებული იქნას სამუშაო ფურცელზე წარმოდგენილი ნებისმიერი ელემენტი. მაგალითად გამშვებ ღილაკად შეიძლება გამოყენებული იქნას ნებისმიერი გრაფიკული გამოსახულება.

ხატვის/ხაზვის (Shapes) ფუნქციის გამოყენებით სამუშაო ფურცელზე გადაგაქვთ ნებისმიერი ავტოფიგურა. მონიშნეთ თქვენს მიერ შერჩეული გამოსახულება დააწკაპუნეთ თავის მარჯვენა ღილაკზე და კონტექსტური მენიუდან, შეარჩიეთ ჩანართი **Assign Macro** (მაკროსის მინიჭება). შემდეგ აუცილებელია მაკროსის გამშვები ფანჯრიდან შეირჩეს მაკროსი „ალმასი“ და მოხდეს მისი გარაფიკული ობიექტისათვის მინიჭება. გარაფიკულ გამოსახულებას უკვე აქვს მინიჭებული მაკროსის ფუნქცია

და მისი გააქტიურების შედეგად მაკროსი შეასრულებს თავის დანიშნულებას.

მაკროსის გაშვება ასევე შესაძლებელია ღილაკის გამოყენებითაც. სამუშაო ფურცელზე განათავსეთ ღილაკი და იმავე წესით, როგორც ხდებოდა გრაფიკულ გამოსახულებაზე მაკროსის მიბმა, მიაბით ღილაკს გამშვები მაკროსის ფუნქცია.

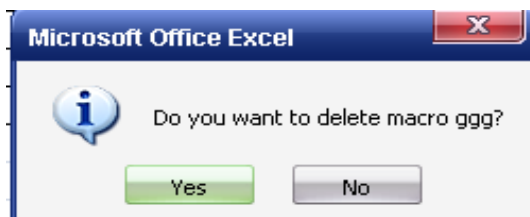
	A	B	C	D	E	F	G	H	I
1	დასახელება	მუარმოვის რიგ ნომერი	მუარმოვის თარიღი	წონა კარატში	წონა მარმუშო				
2	დიდიულო მოლოლი	ინდოეთი	XVII	787	34				
3	ჯონკერი	სამხრეთ აფრიკა	1934	726	145,2				
4	ვარსკლავი-რუხი ღუნო	დასავლეთ აფრიკა	1972	968,9	193,78				
5	მადამ სი	სამხრეთ აფრიკა	1905	3106	621,2				
6	პრეზიდენტი ბეკასი	ბრაზილია	1945	726,6	2222				
7	მდინარე უოი	დასავლეთ აფრიკა	1938	770,7	945				
8	კუსკულ სიორი	სამხრეთ აფრიკა	1893	971,5	4542				
9									
10				მაქსიმალური წონა გრამებში	4542		სამხრეთი აფრიკა		
11									
12									
13									
14									
15									
16									



მაკროსის წაშლა ჩამონათვალიდან

მაკროსთა ჩამონათვალიდან მაკროსის წასაშლელად საჭიროა შემდეგ მოქმედებათა თანამიმდევრული შესრულება:

1. ძირითადი მენიუდან გააქტიურეთ შემდეგი ბრძანებები: **Developer**→**Macros**, მიიღება უკვე ნაცნობი მაკროსის ფანჯარა.
2. მონიშნეთ არასასურველი მაკროსი და გააქტიურეთ აღნიშნულ ფანჯარაში მარჯვენა მხარეს მოთავსებული ჩანართებიდან ღილაკი **Delete**.
3. ბოლოს უნდა მოხდეს შესრულებელი ოპერაციის დადასტურება (**Yes**).



მაკროსის შესაძლებლობების შეზღუდულობა

Excel პროგრამაში მაკროსის გამოყენების საფუძველზე შესაძლებელია ოპერაციათა უმრავლესობის ავტომატიზაცია. მაგრამ მაკროსებს აქვთ გარკვეული შეზღუდულობა. მაკროსის დახმარებით არ შეიძლება შესრულდეს განსხვავებული მოქმედებები ერთსა და იმავე დიაპაზონში მოთავსებულ მნიშვნელობებზე, არ შეიძლება გამოძახებული იქნას **Excel** - ის რამდენიმე დიალოგური ფანჯარა, მაგ. დოკუმენტის შენახვის, არ შეიძლება აისახოს და გამოყენებული იქნას მონაცემთა შეტანის სამომხმარებლო ფორმები.

ეს ნაკლოვანებები გათვალისწინებულია **VBA** პროგრამასთან მუშაობის დროს.

**დავლება: საშუალო წლიური ტემპერატურის
განსაზღვრა**

სამუშაოს შესრულება:

1. გახსენით პროგრამა **Excel**.
2. გამუშაო ფურცელს დაარქვით სახელი „ტემპერატურა“.
3. შეადგინეთ ცხრილი „წლიური ტემპერატურა“

თვის დასახელება	წლიური ტემპერატურა
იანვარი	-5
თებერვალი	-6
მარტი	0
აპრილი	2
მაისი	10
ივნისი	15
ივლისი	18
აგვისტო	20
სექტემბერი	18
ოქტომბერი	6
ნოემბერი	3
დეკემბერი	-9

4. **E3** უჯრაში შეიტანეთ დასახელება საშუალო ტემპერატურა და გამოთვალეთ ფორმულით **AVERAGE(B3:D3)**.
5. მაკროსს დაარქვით სახელი “clis_sash_temp“.

6. მაკროსი გამოსახეთ გრაფიკულ გამოსახულებაზე, მაგ. დილაკზე და ზემოდან გაუკეთეთ წარწერა “წლის საშუალო ტემპერატურა“.
7. ტემპერატურათა მანვენებლის მარჯვენა მხარეს დახატეთ გრაფიკული გამოსახულება ან დილაკი. მიაბით მას მაკროსი, რომელიც შედგეს გამოიტანს **E3** უჯრაში.
8. შეამოწმეთ დილაკის მუშაობა და „წაშაღეთ“.
9. დამოუკიდებლად შეასრულეთ დილაკისა და მაკროსის შექმნის პროცედურა, რომელიც შეასრულებს გამოთვლისა და წაშლის ფუნქციებს, ასევე გამოითვლის:
 - წლიურ მაქსიმალურ ტემპერატურას
 - წლიურ მინიმალურ ტემპერატურას
 - საშუალო ტემპერატურას, გაზაფხულის, ზაფხულის, შემოდგომისა და ზამთრისათვის
 - შექმენით დილაკი, ყველა შედეგის წასაშლელად.

შესრულებული საქმის შედეგი წარმოდგენილია სურათებზე

	A	B	C	D	E
1	წლიური ტემპერატურა				
	თვის	საშუალო			
2	დასახელება	ტემპერატურა			
3	იანვარი	-5			6
4	თებერვალი	-6			
5	მარტი	0			
6	აპრილი	2			
7	მაისი	10			
8	ივნისი	15			
9	ივლისი	18			
10	აგვისტო	20			
11	სექტემბერი	18			
12	ოქტომბერი	6			
13	ნოემბერი	3			
14	დეკემბერი	-9			

საშუალო ტემპერატურა

	A	B	C	D	E	F	G	H
1	წლიური ტემპერატურა							
	თვის	საშუალო						
2	დასახელება	ტემპერატურა						
3	იანვარი	-5			6			
4	თებერვალი	-6						
5	მარტი	0	საშუალო ტემპერატურა		შედეგის წაშლა			
6	აპრილი	2						
7	მაისი	10						
8	ივნისი	15						
9	ივლისი	18						
10	აგვისტო	20						
11	სექტემბერი	18						
12	ოქტომბერი	6						
13	ნოემბერი	3						
14	დეკემბერი	-9						

	A	B	C	D	E	F	G	H	I
1	წლიური ტემპერატურა								
	თვის	საშუალო							
2	დასახელება	ტემპერატურა							
3	იანვარი	-5	წლის საშუალო ტემპერატურა		6	წაშლა		შედეგის წაშლა	
4	თებერვალი	-6	მარტი			წაშლა			
5	მარტი	0	აპრილი		20	წაშლა			
6	აპრილი	2	მაისი			წაშლა			
7	მაისი	10	ივნისი		-9	წაშლა			
8	ივნისი	15	ივლისი			წაშლა			
9	ივლისი	18	აგვისტო		-6,7	წაშლა			
10	აგვისტო	20	სექტემბერი			წაშლა			
11	სექტემბერი	18	ოქტომბერი		4	წაშლა			
12	ოქტომბერი	6	ნოემბერი			წაშლა			
13	ნოემბერი	3	დეკემბერი		17,67	წაშლა			
14	დეკემბერი	-9	წლიური ტემპერატურა		9	წაშლა			
15									
16									

თემა №3. პროგრამულ კოდში ფუნქციების გამოყენება

დავალება: საზღვრის გავლება

სამუშაოს შესრულება:

1. გახსენით პროგრამა **Excel**.
2. სამუშაო ფურცელს **Sheet2** დაარქვით სახელი „საზღვარი“.
3. შეადგინეთ ცხრილი „სახელმწიფო საზღვრის გავლება“

	A	B	C	D
1	სასაზღვრო უბანი	სახმელეთო	მდინარეები	საზღვაო
2	ნორვეგიასთან	43	152,8	23,3
3	ფინეთთან	1091,7	180,1	54
4	ესტონეთთან	89,5	235,3	142
5	ლატვიასთან	137,2	133,3	
6	ლიტვასთან	29,9	236,1	22,4
7	პოლონეთთან	203,3	0,8	32,2
8	ბელორუსიასთან	857,7	381,3	
9	უკრაინასთან	1500,2	425,6	320
10	საქართველოსთან	819,4	56,1	22,4
11	აზერბაიჯანთან	272,4	55,2	22,4
12	ყაზახეთთან	5936,1	1576,7	85,8
13	მონღოლეთთან	2878,6	606,4	
14	ჩინეთთან	650,3	3559	
15	კორეა		17,3	22,1
16	იაპონიასთან			194,3
17	ამერიკასთან			49

4. დაიწყო **F3** უჯრიდან, შეადგინეთ კითხვა პასუხის ცხრილი:

	A	B
1	კითხვა	პასუხი
2	საზღვაო საზღვრის ჯამური სიგრძე	
3	სასმელეთო საზღვრის ჯამური სიგრძე	
4	მდინარეებისა და ტბების ჯამური სასაზღვრო ზოლი	
5	საზღვრის მაქსიმალური სიგრძე	
6	საზღვრის მინიმალური სიგრძე	
7	ყველაზე გრძელი საზღვარი	
8	ყველაზე პატარა საზღვარი	

5. ცხრილის მარჯვენა მხარეს ჩასვით გრაფიკული ობიექტი, რომელზეც მიმაგრებული იქნება მაკროსი.

M20				fx											
	A	B	C	D	E	F	G	H	I	J	K	L	M		
1	სახელმწიფო საზღვრის სიგრძე														
	სასაზღვრო უბანი	სასმელეთო	მდინარე/ტბა	საზღვაო											
2															
3	ნორვეგიასთან	43	152,8	23,3											
4	ფინეთთან	1092	280,2	54											
5	ესტონეთთან	89,5	235,3	142											
6	ლატვიასთან	137,2	133,3												

	A	B	C
1	კითხვა	პასუხი	
2	საზღვაო საზღვრის ჯამური სიგრძე	989,9	
3	სასმელეთო საზღვრის ჯამური სიგრძე	14509,3	
4	მდინარეებისა და ტბების ჯამური სასაზღვრო ზოლი	7616	
5	საზღვრის მაქსიმალური სიგრძე	59361	

6. გამოსათვლელად გამოიყენეთ ფორმულები:

უჯრისათვის **H3: =SUM (D3:D18);**

უჯრისათვის **H4: = SUM (B3:B18);**

უჯრისათვის **H5: = SUM (C3:C18);**
 უჯრისათვის **H6: = SUM (B3:D18);**
 უჯრისათვის **H7: =MIN(B3:D18);**
 უჯრისათვის **H8: =IF**
(H6=MAX(H6:H7);B2;IF(H6=MAX(C3:C18);C2;D2));
 უჯრისათვის **H9:**
=IF(H6=MIN(H6:H7);B2;IF(H6=MIN(C3:C18);C2;D2)).


გრაფიკულ ობიექტზე მაკროსის მიმაგრების და გაშვების შემდეგ მოცემული ფორმულების გამოყენების შედეგად ეკრანზე მიიღება შედეგი:

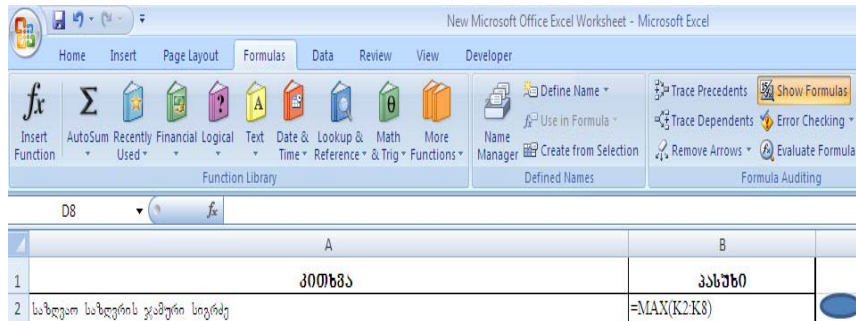
The screenshot shows the Microsoft Excel interface with a spreadsheet containing data. The main spreadsheet has columns A, B, C, and D. The data is as follows:

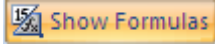
	A	B	C	D
1	სასაზღვრო უბანი	სახშირეუტი	მშენებელი	საზღვარი
2	ნორვეგია	43	152,8	23,3
3	ფინეთი	1091,7	180,1	54
4	ესტონეთი	89,5	235,3	142
5	ლატვია	137,2	133,3	
6	ლიტვა	29,9	236,1	22,4
7	პოლონეთი	203,3	0,8	32,2
8	ბელარუსია	857,7	381,3	
9	უკრაინა	1500,2	425,6	320
10	საქართველო	819,4	56,1	22,4
11	აზერბაიჯანი	272,4	55,2	22,4
12	ყაზახეთი	5936,1	1576,7	85,8
13	მონღოლეთი	2878,6	606,4	
14	ჩინეთი	650,3	3559	
15	კორეა		17,3	22,1
16	იაპონია			194,3
17	ამერიკა			49

Below the main spreadsheet, there is a smaller table with columns A, B, and C:

	A	B	C
1	კითხვა	პასუხი	
2	საზღვრო საზღვრის ჯამური სიგრძე	989,9	
3	საშუალო საზღვრის ჯამური სიგრძე	14509,3	
4	მინარეებისა და ტბების ჯამური სასაზღვრო ზოლი	7616	
5	საზღვრის მაქსიმალური სიგრძე	59361	
6	საზღვრის მინიმალური სიგრძე	0,8	
7	ყველაზე უბრალო საზღვარი	სახმელეთო	
8	ყველაზე ატარა საზღვარი	საზღვრო	

7. იმისათვის რომ ეკრანზე გამოჩნდეს თუ რომელი ფორმულების გამოყენებით მოხდა შედეგის მიღება, ძირითადი მენიუდან გაააქტიურეთ Formulas ჩანართში  ბრძანება. თქვენ დაინახავთ ფორმულებს, რომელიც აქამდე შესაძლებელი იყო დაგენახათ მხოლოდ ფორმულების ზოლში



8. იმისათვის, რომ სამუშაო დოკუმენტი დაუბრუნდეს ჩვეულ მდგომარეობას და პასუხების სვეტში გამოჩნდეს ისევ შედეგები, საჭიროა ფორმულების მენიუდან გამორთოთ  ბრძანება.

9. შექმენით მაკროსი, რომელიც წაშლის **H3:H9** დიაპაზონში მოთავსებულ მონაცემებს და მიამაგრეთ გრაფიკულ ობიექტს.

10. კიდევ ერთხელ შეამოწმეთ ყველა მაკროსის ფუნქციონირების პროცესი გრაფიკულ ობიექტებთან მიმართებით.

დავალება: საქართველოს დემოგრაფია (%)

სამუშაოს შესრულების მიმდევრობა:

1. გახსენით **Excel** პროგრამა.
2. სამუშაო ფურცელს **Sheet 3** დაარქვით სახელი „შობადობა“.
3. შეადგინეთ ცხრილი „მოსახლეობის განახლება“

	A	B	C	D
1	პერიოდი	შობადობა	ბარდაცვალება	ნამატი
2	1970	44,7	21,3	23,4
3	1982	33	20,6	12,4
4	1990	23,2	7,4	15,8
5	1995	15,9	10,6	5,3
6	2000	14,6	10,6	4
7	2003	12,1	11,4	0,7
8	2006	9,4	14,5	-5,1
9	2008	9,3	15	-5,7
10	2011	8,6	13,8	-5,2

4. შეადგინეთ კითხვა პასუხის ცხრილი და ჩაწერეთ კითხვები. პასუხი მიიღეთ გრაფიკული ობიექტზე მიმაგრებული მაკროსის გააქტიურებით.

	A	B
1	კითხვა	პასუხი
2	საშუალო შობადობა (1970-1982)	
3	საშუალო შობადობა (1990-1995)	
4	საშუალო სიკვდილიანობა (2000-2003)	
5	საშუალო მდგომარეობა (2006-2008)	
6	მინიმალური შობადობა	
7	მინიმალური გარდაცვალება	
8	მინიმალური ნამატი	
9	მაქსიმალური შობადობა	
10	მაქსიმალური გარდაცვალება	
11	მაქსიმალური ნამატი	

5. მაკროსის ჩაწერის შემდეგ **პითხვა/პასუხი** მაკროსის რედაქტორების ფანჯრიდან დაათვალიერეთ პროგრამული კოდი, რომელსაც უნდა ჰქონდეს შემდეგი სახე:

```
Sub KITXVA()  
" KITXVA Макрос  
Range("F15").Select  
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-12]C[-4]:R[-11]C[-4])"  
Range("F16").Select  
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-11]C[-4]:R[-5]C[-4])"  
Range("F17").Select  
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-14]C[-3]:R[-13]C[-3])"  
Range("F18").Select  
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-13]C[-3]:R[-7]C[-3])"  
Range("F19").Select  
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-16]C[-2]:R[-15]C[-2])"  
Range("F20").Select  
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-15]C[-2]:R[-9]C[-2])"  
Range("F21").Select  
ActiveCell.FormulaR1C1 = "=MIN(R[-18]C[-4]:R[-10]C[-4])"  
Range("F22").Select  
ActiveCell.FormulaR1C1 = "=MIN(R[-19]C[-3]:R[-11]C[-3])"  
Range("F23").Select  
ActiveCell.FormulaR1C1 = "=MIN(R[-20]C[-2]:R[-12]C[-2])"  
Range("F24").Select  
ActiveCell.FormulaR1C1 = "=MAX(R[-21]C[-4]:R[-13]C[-4])"  
Range("F25").Select  
ActiveCell.FormulaR1C1 = "=MAX(R[-22]C[-3]:R[-14]C[-3])"  
Range("F26").Select  
ActiveCell.FormulaR1C1 = "=MAX(R[-23]C[-2]:R[-15]C[-2])"  
End Sub  
Sub GASUFT()  
' Range("F15:F26").Select  
Selection.ClearContents  
End Sub  
საბოლოო დოკუმენტს ექნება შემდეგი სახე:
```


	A	B	C	D
1	პერიოდი	შობადობა	ბარდაცვალება	ნამატი
2	1970	44,7	21,3	23,4
3	1982	33	20,6	12,4
4	1990	23,2	7,4	15,8
5	1995	15,9	10,6	5,3
6	2000	14,6	10,6	4
7	2003	12,1	11,4	0,7
8	2006	9,4	14,5	-5,1
9	2008	9,3	15	-5,7
10	2011	8,6	13,8	-5,2
14	კითხვა			პასუხი
15	საშუალო შობადობა (1970-1982)			38,85
16	საშუალო შობადობა (2000-2003)			13,3
17	საშუალო სიკვდილიანობა (1970-1982)			20,95
18	საშუალო მდგომარეობა (2006-2008)			11,9
19	საშუალო მდგომარეობა (1990-1995)			17,9
20	საშუალო მატება (2006-2008)			1,44
21	მინიმალური შობადობა			8,8
22	მინიმალური გარდაცვალება			7,4
23	მინიმალური ნამატი			-5,7
24	მაქსიმალური შობადობა			44,7
25	მაქსიმალური გარდაცვალება			21,3
26	მაქსიმალური ნამატი			23,4

უკანსებით
კითხვებს

შედეგების
წასვლა

- ა. შექმენით მაკროსი, რომელიც შეცვლის ცხრილის ფერს (მწვანე ფერი) და ფონტის სტილს (გამუქებული დახრილი სიმბოლოები).
- ბ. ეს მაკროსები მიამაგრეთ გრაფიკულ ობიექტს.

დავალება: ნალექების რაოდენობა

სამუშაოს შესრულება:

1. გახსენით **Excel** პროგრამა.
2. სამუშაო ფურცელს **Sheet4** დაარქვით სახელი “ნალექი”.
3. შეადგინეთ ცხრილი „ნალექების რაოდენობა“

	A	B	C	D
1	თვე	2003	2006	2009
2	იანვარი	37,2	34,5	8
3	თებერვალი	11,4	51,3	12
4	მარტი	16,5	20,5	3,8
5	აპრილი	19,5	26,9	11,9
6	მაისი	11,7	45,5	66,3
7	ივნისი	129,1	71,5	60
8	ივლისი	57,1	152,9	50,6
9	ოქტომბერი	86	14,5	74,9
10	ნოემბერ	12,5	21	56,6
11	დეკემბერი	21,2	22,3	9,4
12	სულ ნალექი			

4. ცხრილში მოცემული მონაცემებისათვის განსაზღვრეთ:

- ნალექის მაქსიმალური რაოდენობა 3 წელში;
- ნალექის მინიმალური რაოდენობა 3 წელში;
- ნალექის ჯამური რაოდენობა 3 წელში;
- 3 წელში ნალექის თვიური საშუალო რაოდენობა;
- გვაღვა 3 წელში;
- თვეში მოსული უხვი ნალექი.

5. შედეგები მაკროსის საშუალებით მიიღეთ დამატებით ცხრილში „2003-2009 წლის შედეგები“ .
6. შექმენით მაკროსი უჯრებში შეტანილი მონაცემების წასაშლელად. მიაბით მაკროგი გრაფიკულ ობიექტს.

2003-2009 წლის შედეგები	
ნალექის მაქს. რაოდენობა 3 წელში	
ნალექის მინ. რაოდენობა 3 წელში	
ნალექის ჯამური რაოდენობა 3 წელში	
3 წელში ნალექის თვიური საშ. რაოდ.	
გვაღვა 3 წელში	
უხვი ნალექი თვეში	

7. ძირითადი ცხრილი მოთავსებული უნდა იყოს **B5–E18** დიაპაზონში, ხოლო დამატებითი კითხვა პასუხის ცხრილი შეადგინეთ **G5** უჯრიდან.

უჯრა **C18** შეიცავს ფორმულას =SUM(C6:C17);

უჯრა **C18** შეიცავს ფორმულას =SUM(D6:D17);

უჯრა **C18** შეიცავს ფორმულას =SUM(E6:E17);

შექმენით მაქსიმალური და მინიმალური მნიშვნელობის გამოსათვლელი მაკროსი და მიაბით იგი გრაფიკულ ობიექტს.

უჯრა **H6** შეიცავს ფორმულას =MAX(C6:E17);

უჯრა **H7** შეიცავს ფორმულას =MIN(C6:E17);

შექმენით ჯამური და საშუალო მნიშვნელობის გამოსათვლელი მაკროსი და მიაბით იგი გრაფიკულ ობიექტს.

უჯრა **H8** შეიცავს ფორმულას =SUM(C6:E17);

უჯრა **H9** შეიცავს ფორმულას =AVERAGE(C6:E17);

შექმენით გვაღვისა და უხვნალექის მნიშვნელობის გამოსათვლელი მაკროსი და მიაბით იგი გრაფიკულ ობიექტს.

უჯრა **H10** შეიცავს ფორმულას =SUM(C6:E17);

უჯრა **H11** შეიცავს ფორმულას = AVERAGE(C6:E17);

შექმენით მაკროსი, რომელიც უზრუნველყოფს უჯრათა დიაპაზონის **H5–H11** გასუფთვებას და მიაბით გარაფიკულ ობიექტს.

დავალება: ავტოფორმებთან მუშაობა

სამუშაოს შესრულება:

1. გახსენით **Excel** პროგრამა.
2. სამუშაო ფურცელს **Sheet5** დაარქვით სახელი “კლასის სია”.
3. შეადგინეთ ცხრილი და შეაყვეთ პირველი სტრიქონი:

	A	B	C	D	E
1	გვარი	სახელი	მამის სახ	დაბ.თარიღი	ტელეფონი
2	ბერიძე	ნინო	გიორგის-ას	01,01,98	4564564

4. შეადგინეთ ცხრილის შესავსები ავტოფორმა, შეიტანეთ მონაცემები

The screenshot shows a form window titled "Sheet9". On the left side, there are five input fields with labels: "rig.nom.:", "qvari:", "saxeli:", "dab.TariRi:", and "telefoni:". On the right side, there is a "New Record" section containing several buttons: "New", "Delete", "Restore", "Find Prev", "Find Next", "Criteria", and "Close".

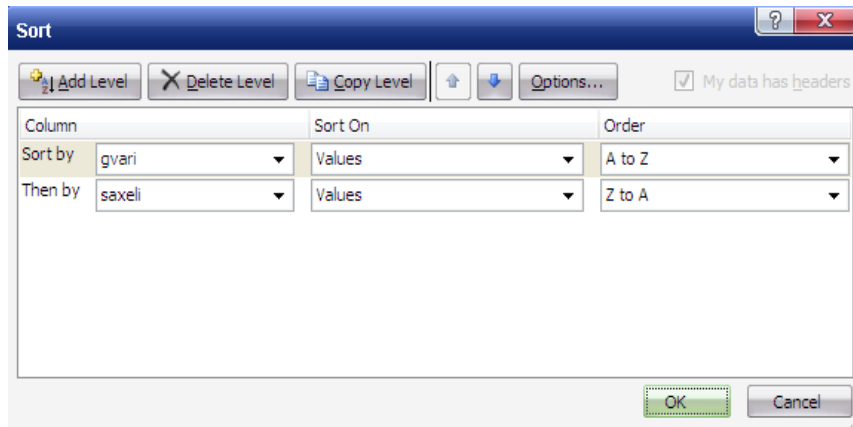
5. შექმენით ავტოფორმის გამოსაძახებელი მაკროსი და მიაბით გრაფიკულ ობიექტს.

6. შეავსეთ ცხრილი შემდეგი მონაცემებით:

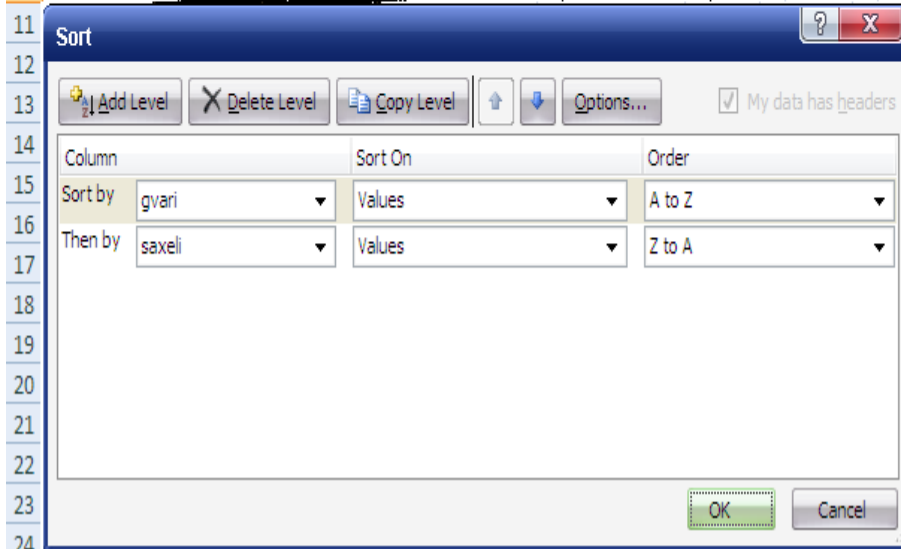
	A	B	C	D	E
1	რიგ.ნომ.	გვარი	სახელი	დაბ.თარიღი	ტელეფონი
2	1	გერიძე	ნონო	05.09.98	45-64-56
3	2	დოლიძე	იკა	01.01.98	45-61-20
4	3	ნონიძე	შოთა	03.21.97	48-45-66
5	4	ლაბაძე	ბირგი	05.03.91	45-48-72
6	5	კაკაბაძე	ლია	23.12.67	56-45-46
7	6	პიპინაძე	ლადო	28.11.79	21-84-85
8	7	ჯეღია	თეა	22.06.77	48-45-46
9	8	შიდაძე	ელისო	25.02.72	56-56-21
10	9	ლოლაძე	თორნიკე	24.04.97	96-55-56

7. შექმენით ცხრილის სორტირების მაკროსი (ველებისათვის, გვარი/სახელი) **B1–F16** უჯრათა დიაპაზონისათვის. მიაბით იგი გეომეტრიულ ობიექტს.

სორტირების ფანჯარა გამოიტანეთ DataSort ბრძანებით.



	A	B	C	D	E	F	G
1	რიგ.ნომ.	გვარი	სახელი	დაბ.თარიღი	ტელეფონი		
2	1	გურიძე	ნინო	05.09.98	45-64-56		
3	2	დოლიძე	იკა	01.01.98	45-61-20		
4	3	ჭელია	თბა	03.21.97	48-45-66		
5	4	კახაბაძე	ლია	05.03.91	45-48-72		
6	5	პინაძე	ლადო	23.12.67	56-45-46		
7	6	წაბაძე	გიორგი	28.11.79	21-84-85		
8	7	ლოლაძე	თორნიკე	22.06.77	48-45-46		
9	8	შიქაძე	ელისო	25.02.72	56-56-21		
10	9	ნინიძე	შოთა	24.04.97	96-55-56		



8. შექმენით თარიღის სორტირების მაკროსი და მიაბით გეომეტრიულ ობიექტს.

9. შექმენით სიმბოლოთა ფერის შეცვლის მაკროსი (მიაბით გრაფიკულ ობიექტს) შემდეგ დიაპაზონში:

A1–F1 — გამუქებული სიმბოლოები, ზომა 16, ფონი მწვანე, სიმბოლოები თეთრი;

A2–A16 — ვარდისფერი ფონ;

- B2-B16** — ლურჯი ფონი;
- C2-C16** — ნაცრისფერი ფონი;
- D2-D16** — ვარდისფერი ფონი;
- E2-E16** — ლურჯი ფონი;
- F2-F16** — ნაცრისფერი ფონი.

10. შექმენით მაკროსი (მიაბით გარფიკულ ობიექტს) დააბრუნეთ საწყის მდგომარეობაში.
11. შეამოწმეთ მაკროსის მუშაობა.

დავალება: დანაშაული მაკროსი ტექნოლოგიების სამყაროში

სამუშაოს შესრულება:

1. გახსენით **Excel** პროგრამა.
2. სამუშაო ფურცელს **Sheet6** დაარქვით სახელი “დანაშაული”.
3. შეადგინეთ ცხრილი:

	A	B	C	D	E
1	დანაშაულის სახე	თარიღი			
2		1998	1999	2000	2001
3	უცხო კომპიუტერულ ქსელში შეღწევა	22145	21568	18900	17895
4	მავნე პროგრამების (ვირუსების) გაერცელება	45475	46879	51057	60458
5	კომპიუტერის მუშაობაში ხელის შეშლა	45879	45487	44568	45757
6	მარალი ტექნოლოგიების სფეროში ჩადენილი დანაშაულები				

4. მოცემული ცხრილისათვის განსაზღვრეთ:

- დანაშაულთა ჯამური რაოდენობა 1998, 1999, 2000 და 2001 წლებში (შეავსეთ ბოლო სტრიქონი)
 - ჩადენილი დანაშაულთა მაქსიმალური რაოდენობა 4 წლის განმავლობაში;
 - ჩადენილი დანაშაულთა მინიმალური რაოდენობა 4 წლის განმავლობაში;
 - ოთხი წლის განმავლობაში ჩადენილი დანაშაულის „უცხო კომპიუტერულ ქსელში შეღწევა“ ჯამური რაოდენობა;
 - მანე პროგრამების გავრცელებით ჩადენილი დანაშაულის რაოდენობა ოთხი წლის განმავლობაში;
 - კომპიუტერის მუშაობაში ხელის შეშლით გამოწვეული დანაშაულის რაოდენობა ოთხი წლის განმავლობაში;
 - ყველაზე ნაკლებად ჩადენილ დანაშაულთა რაოდენობა;
 - ყველაზე მეტად ჩადენილ დანაშაულთა რაოდენობა;
 - ყველაზე მეტად გავრცელებული დანაშაულთა სახეობა;
- შედეგები გამოიტანეთ დამატებით ცხრილში.

თემა №4. სამომხმარებლო ფორმებთან მუშაობა

შესასწავლი საკითხები:

- ეკრანული ფორმების შექმნა;
- მართვის ელემენტების პანელი;
- UserForm ობიექტის თვისებები, მეთოდები და ხდომილებები.

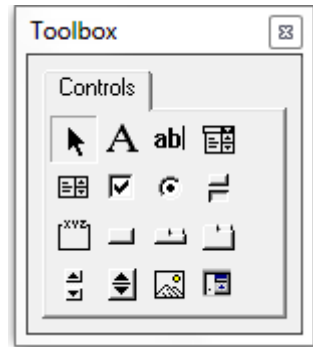
სამომხმარებლო ფორმა UserForm


ობიექტ-ორიენტირებული დაპროგრამება დასამუშავებელი დანართის გრაფიკული ინტერფეისის შექმნის საშუალებას იძლევა მმართველი ელემენტების გამოყენების საშუალებას იძლევა. მართვის ელემენტები წარმოადგენენ ობიექტებს. ამიტომ, როგორც ნებისმიერი ობიექტს, მათაც გააჩნიათ თვისებები, მეთოდები და რეაგირებენ ხდომილებებზე. მართვის ელემენტები შეიძლება განთავსდეს როგორც მუშა ფურცელზე, ასევე ეკრანულ ფორმაზე. ეკრანული ფორმა, ანუ სამომხმარებლო ფორმა – *UserForm* არის ფანჯარა, რომელიც წარმოადგენს დანართის ინტერფეისის ნაწილს.

სამომხმარებლო ფორმასთან მუშაობა მოიცავს რამდენიმე ეტაპს:









- სამუშაო ფაილის გახსნა ან შექმნა;
- Visual Basic რედაქტორში გადასვლა;
- სამომხმარებლო ფორმის შექმნა;
- შექმნილი ფორმისათვის თვისებების გამოყენება;
- ფორმაზე მართვის ელემენტების შექმნა;
- დაიწეროს ხდომილებების დამუშავების პროცედურები.








მართვის ელემენტების პანელი



სამომხმარებლო ფორმის შექმნისას ავტომატურად აისახება ინსტრუმენტების პანელი, რომელიც შეიცავს მართვის ელემენტებს. თუ *UserForm* გამოტანის დროს ინსტრუმენტების პანელი არ გამოვა, იგი შეიძლება გამოვიძახოთ შემდეგი ბრძანებით: **View | ToolBox** ან  ღილაკზე დაჭერით.

ინსტრუმენტების პანელის ანუ მართვის ელემენტების პანელის დანართის ობიექტების შექმნა და რედაქტირება. ცხრილში მოცემულია მართვის ელემენტები და მათი განმარტებები:

ელემენტის შესაქმნელი ღილაკი	სახელი	მართვის ელემენტი
	Label	წარწერა
	TextBox	ველი
	CommandButton	მართვის ღილაკი
	ListBox	სია
	ComboBox	ველი სიით
	ScrolBar	გადახვევის ზოლი
	SpinButton	მთვლედი
	OptionButton	გადამრთველი

	CheckBox	დროშა
	ToggleButton	გამომრთველი
	Frame	ჩარჩო
	Image	ნახატი
	MultiPage	გვერდების ნაკრები
	TabStrip	ჩანართების ნაკრები
	RefEdit	ველი გადახვევით

Label – ტექსტის სტატიკური ველია, რომელიც შეიცავს განმარტებით ინფორმაციას, ხშირად იგი გამოიყენება შედეგების გამოსატანად.

TextBox – მომხმარებლის მიერ ტექსტის შესატანი ველია, რომელიც შემდგომში გამოიყენება პროგრამაში.

CommandButton – ელემენტი, რომლის დახმარებითაც სამომხმარებლო ფორმაზე შეიძლება განთავსდეს ბრძანების ღილაკი. აღნიშნულ ღილაკზე დაჭერისას შესრულდება დაპროგრამებული მოქმედება.

ListBox – გამოიყენება მნიშვნელობების სიის შესანახად. სიაში მომხმარებელმა შეიძლება აირჩიოს ერთი ან რამდენიმე მნიშვნელობა, რომელიც შემდგომ გამოიყენება პროგრამაში.

ComboBox - გამოიყენება მნიშვნელობების სიის შესანახად. ეს ელემენტი ითავსებს ListBox და TextBox ელემენტების მნიშვნელობებს.

ScrolBar – გამოიყენება ამორჩევის მექანიზმის სახით. მაგალითად გრაფიკულ გამოსახულებებში შეიძლება აირჩეს ფერი ან ნახატის ტონები შეიცვალოს.

SpinButton – გამოიყენება რიცხვითი მნიშვნელობის შესატანად ან შესაცვლელად.

OptionButton – რამდენიმე ურთიერთგამომრიცხავი პარამეტრის ან მოქმედებიდან ერთის არჩევის საშუალებას

იძლევა. გადამრთველი აისახება ჯგუფურად, უზრუნველყოფს ალტერნატიული არჩევის შესაძლებლობას.

CheckBox – მომხმარებელს სთავაზობს ამორჩევის შესაძლებლობას. დროშას აქვს ორი მდგომარეობა: აღმართული ან დაშვებული.

ToggleButton – ღილაკია, რომელიც მასზე დაჭერის შემდეგ დარჩება დაჭერილ მდგომარეობაში და დაუბრუნდება საწყის მდგომარეობას განმეორებითი დაწკაპუნების შემდეგ.

Frame – გამოიყენება აზრობრივად ერთმანეთთან დაკავშირებული ელემენტების ვიზუალურად ერთ ჯგუფში გაერთიანებისათვის.

Image – ეკრანულ ფორმაზე გრაფიკული ელემენტების განთავსების საშუალებას იძლევა. გამოსახულება შეიძლება ჩაისვას შემდეგი ტიპის გრაფიკული ფაილებიდან: *bmp; cur; gif; ico; jpg; wmf*.

MultiPage – ელემენტი შეიცავს ერთ ან რამოდენიმე გვერდს, გვერდები წარმოადგენს ფორმებს, რომლებიც შეიცავს საკუთარ მმართველ ელემენტებს.

TabStrip – შეიცავს ერთ ან რამდენიმე ჩანართს. გამოიყენება დაკავშირებული ინფორმაციის ჯგუფში ორგანიზებისათვის.

RefEdit - ელემენტი გავს ჩვეულებრივ შეტანის ველს, მაგრამ აქვს ღილაკი ველის მარჯვენა მხარეს, რომელიც *сворачивает* ეკრანულ ფორმას, რაც მუშა ფურცელზე უჯრების ნებისმიერი დიაპაზონის იოლად არჩევის საშუალებას იძლევა.

ჩვენი სამუშაოს შესასრულებლად ვისარგებლებთ ხშირად გამოყენებადი ღილაკებით, ესენია:

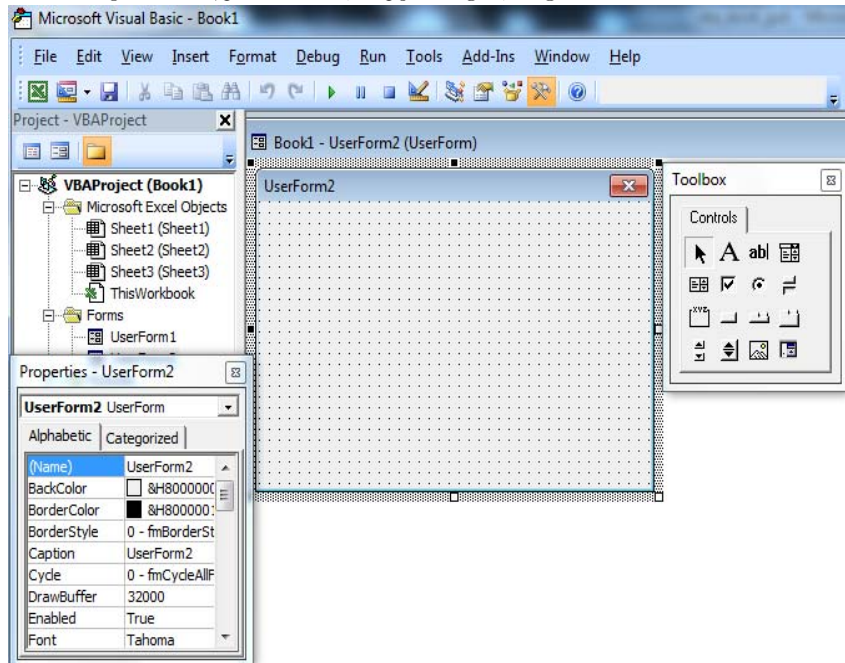
- **Label;**
- **TextBox;**
- **CommandButton;**

- Image;
- **OptionButton**;
- **CheckBox**.

მომხმარებლის ფორმის შექმნა.


მომხმარებლის ფორმის შესაქმნელად უნდა შესრულდეს შემდეგი მოქმედებები:

- შექმენით ახალი ფაილი ან გახსენით უკვე არსებული.
- გახსენით Visual Basic რედაქტორის ფანჯარა.
- **Insert** მენიუში აირჩიეთ **UserForm** ბრძანება. გამოჩნდება ახალი ეკრანული ფორმა.



შექმნილ ეკრანულ ფორმას დუმილით აქვს სათაური UserForm1. ფორმის შექმნის შემდეგ საჭიროა ამ ობიექტის თვისებების დათვალიერება და ცვლილებების შეტანა, თუ

ამას მოითხოვს ამოცანა. თვისებების ფანჯრის ასახვისათვის უნდა ავირჩიოთ ბრძანება: **View | Properties**

Window, ან დავაწკაპუნოთ ინსტრუმენტების პანელზე  დილაკზე.

UserForm ობიექტის ძირითადი თვისებები:

სახელი	სამომხმარებლო ფორმის სახელი
Caption	ფორმის სათაურის სტრიქონში გამოსული ტექსტი
BackColor	ფორმის ფონის ფერი
BorderStyle	საზღვრების ტიპის დაყენება
Picture	უზენებს ფორმის ფონად ასახულ ნახატს
Left, Top	უზენებს ფორმის მარცხენა და ზედა კუთხის ადგილმდებარეობას(წერტილებში)
Height, Width	იძლევა ფორმის სიმაღლეს და სიგანეს (წერტილებში)
CpecialEffect	აყენებს ფორმის მუშა ველის ასახვის ტიპს
StartUpPosition	აბრუნებს ფორმის საწყისი ასახვის მნიშვნელობებს. დასაშვები მნიშვნელობებია: <ul style="list-style-type: none"> - Manual (საწყისი მნიშვნელობა არ ყენდება) - CenterOwner (იმ ობიექტის ცენტრის მიმართ გასწორება, რომელსაც ეკუთვნის ფორმა) - CenterScreen (სწორება ეკრანის ცენტრზე) - WindowDefault (ეკრანის ზედა მარცხენა კუთხის მდგომარეობა)

UserForm ობიექტის ძირითადი მეთოდები

Show	ფორმის ასახვა ეკრანზე
Hide	ფორმის გაქრობა ეკრანიდან

მართვის ელემენტების პანელის გამოყენებით შეუვსებელი ფორმიდან შეიძლება მოთხოვნილი დიალოგური ფანჯრის კონსტრუირება. მართვის ელემენტების ფორმაზე განთავსება ხდება შემდეგი მოქმედებების მიმდევრობით:

- საჭიროა ფორმაზე განსათავსებელი ელემენტის ნიშანზე დაჭერა.
- თავის ისარი გააჩერეთ იმ ადგილას, სადაც უნდა განთავსდეს მართვის ელემენტი.
- თავის მარცხენა ღილაკზე დაჭერილი თითით მოხაზეთ ელემენტი.
- აიღეთ თავის ღილაკიდან თითი, ელემენტი შექმნილია.

ეკრანული ფორმის დამუშავების შემდეგ შეიძლება ვნახოთ მიღებული შედეგი შემდეგი ბრძანებით: Run | Run Sub/UserForm ან F5 ღილაკზე დაჭერით. ფორმა გამოისახება მუშა წიგნის ზედაპირზე. დახურვის ღილაკით დაიხურება ფორმა და დაბრუნდებით Visual Basic გარემოში.

დამუშავებულ დანართს ჰქვია პროექტი. იგი მოიცავს არა მარტო ფორმას მასზე განთავსებული მმართველი ელემენტებით, არამედ ხდომილების დამუშავების პროგრამულ მოდულებს, რომლებიც აღწერენ დანართის ობიექტების ქცევებს და ობიექტების ერთმანეთთან ურთიერთქმედებას.

დავალება: შევქმნათ სამომხმარებლო ფორმა, განვათავსოთ მასზე მართვის ელემენტები: **Label**, **CommandButton**. დანართის გაშვების შემდეგ დაგენერირდება შემთხვევითი რიცხვი, იგი აისახება **Label** ელემენტში. გამოვითვალთ ამ რიცხვიდან კვადრატული ფესვი, სინუსი, კოსინუსი. შედეგები გამოვიტანოთ სხვა **Label** ელემენტებში. გარდა ამისა ფორმაზე გამოვიტანოთ მიმდინარე თარიღი. „გამოსვლა“ ღილაკზე დაჭერისას ფორმა იმალება ეკრანიდან. Excel- ის ფურცელზე ფორმის გასაშვებად საჭიროა შევქმნათ ღილაკი წარწერით: „VBA-ს ფუნქციები“.

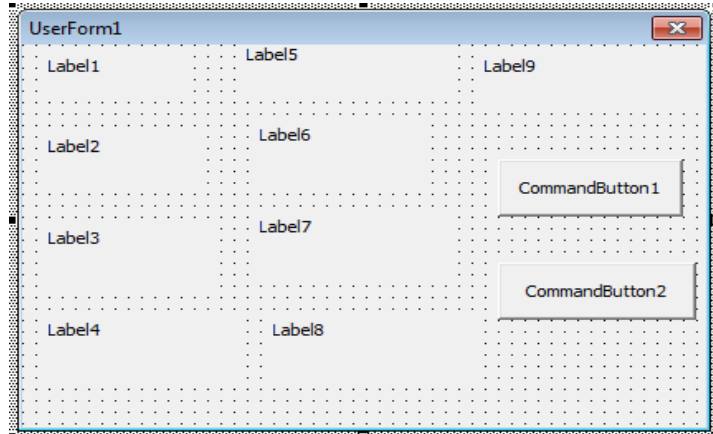
სამუშაოს მიმღევრობა:

- გაუშვით Excel.
- ჩატვირთეთ VBA.
- შეამოწმეთ, რომ ეკრანზე გამოვიდეს პროექტის ფანჯარა და თვისებების ფანჯარა. შეასრულეთ ბრძანება **Insert | UserForm**. ფორმას შეიძლება შევუცვალოთ ზომები სტანდარტული სახით.
- გაააქტიურეთ გამოსული ფორმა და დააყენეთ მოცემული ობიექტის თვისებების მნიშვნელობები:

თვისება	მნიშვნელობა
Caption	VBA-ს ფუნქციები
Height	350
Width	500
SpecialEffect	2(შეხედულებისამებრ 0-6)
StartupPosition	2-CenterScreen

- ფორმას დაამატეთ მართვის საშუალებები. დამუშავებისას საჭიროა ელემენტების პანელიდან მხოლოდ ორი ინსტრუმენტი: **Label** და **CommandButton**. ფორმა

მართვის ელემენტებით კონსტრუქტორის რეჟიმში მიიღებს შემდეგ სახეს.



- გამოთვლის შედეგების გამოსატანად მოამზადეთ ელემენტები: **Label1, Label2, Label3, Label4**. დააყენეთ **Label1- Label4** ობიექტების თვისებების მნიშვნელობები.

თვისება	მნიშვნელობა
Caption	წარწერა არ უნდა ჰქონდეს
Font	AcadNusx, მსხვილი, დახრილი, ზომა 12
Width	150
Height	45
SpecialEffect	Label1-1; Label2-2; Label3-3; Label4-6
TextAllign	2-fmTextAlignCenter

- განმარტებითი წარწერებისათვის მოამზადეთ ელემენტები: **Label5, Label6, Label7, Label8**. დააყენეთ თვისებების მნიშვნელობები **Label5 – Label8** ობიექტებისათვის:

თვისება	მნიშვნელობა
Caption	Label5- „შემთხვევითი რიცხვი“ Label6- „კვადრატული ფესვი“ Label7- „სინუსი“ Label8- „კოსინუსი“
Font	AcadNusx, მსხვილი, ზომა 12
Width	120
Height	35
SpecialEffect	1
TextAllign	1-fmTextAlignLeft

- მიმდინარე თარიღის გამოსატანად მოამზადეთ მართვის ელემენტი **Label9**.

დააყენეთ **Label9** ობიექტის თვისებების მნიშვნელობები:

თვისება	მნიშვნელობა
Caption	თარიღი
Font	AcadNusx, მსხვილი, დახრილი, ზომა 12
Width	100
Height	50
SpecialEffect	0
TextAllign	2-fmTextAlignCenter

- ფორმას დაამატეთ ბრძანების ღილაკი პროგრამის გასაშვებად, რისთვისაც გამოიყენეთ **CommandButton1** მართვის ელემენტი.

დააყენეთ **CommandButton1** ობიექტის თვისებების მნიშვნელობები:

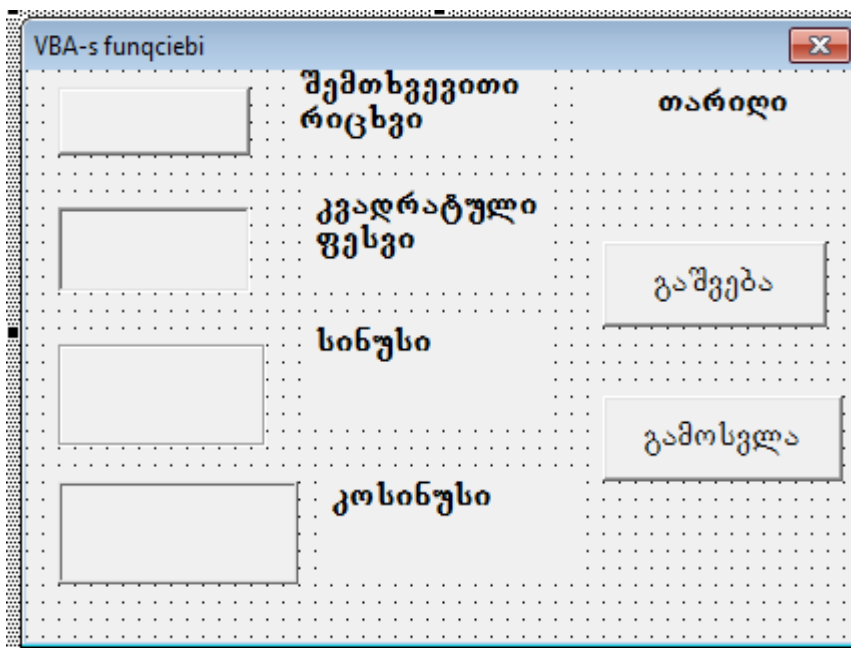
თვისება	მნიშვნელობა
Caption	გაშვება
Font	AcadNusx, ზომა 12

- დაამატეთ ფორმას ბრძანების ღილაკი ფორმის დახურვისათვის, რისთვისაც გამოიყენეთ **Command-Button2** მართვის ელემენტი.

დააყენეთ **CommandButton2** ობიექტის თვისებების მნიშვნელობები:

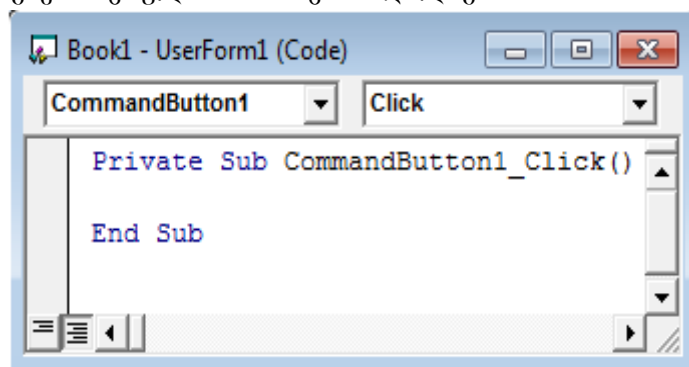
თვისება	მნიშვნელობა
Caption	გამოსვლა
Font	AcadNusx, ზომა 12

- ყველა პუნქტის შესრულების შემდეგ ფორმა კონსტრუქტორის რეჟიმში გამოიყურება შემდეგნაირად:



გადავიდეთ პროგრამის ტექსტის დაწერაზე.

- პროგრამის დაწერამდე შევამოწმოთ გამზადებული ფორმის მუშაობისუნარიანობა. ფორმის ეკრანზე გამოსატანად შევასრულოთ ბრძანება **Run | Run Sub/UserForm** ან დავაჭიროთ **F5** ღილაკს. Excel-ის ფურცლის ფონზე გამოჩნდება ჩვენ მიერ შექმნილი ფორმა. „გაშვება“ ან „გამოსვლა“ ღილაკებზე დაჭერისას არაფერი არ შესრულდება, რადგანაც ჩვენ ჯერ არ დავგიპროგრამებია გამოძახილი ხდომილებაზე. დავხუროთ ფორმა და დავბრუნდეთ VBA რედაქტორში.
- ფორმაზე ორჯერ დააწკაპუნეთ „გაშვება“ ღილაკზე, რადგანაც გამოვიდეს პროგრამის სათაური, რომელიც დაკავშირებულია ბრძანების ღილაკთან



პროგრამა იწვება სათაურით Private Sub (ლოკალური პროცედურა) და მთავრდება სიტყვებით End Sub (პროცედურის დასასრული). ამ წინადადებებს შორის იწერება პროგრამის ტექსტი.

- გამოჩნდება პროგრამის სათაური. დავწეროთ პროგრამის ტექსტი, შემდეგ კი გავარჩიოთ ცალკეული სტრიქონი.

```

Randomize
Label9.Caption = Date
Label1.Caption = Int(Rnd * 90)+1)
Label2.Caption = Sqr(Label1.Caption)
Label3.Caption = Cos(Label1.Caption * 3.14159 / 180)
Label4.Caption = Sin(Label1.Caption * 3.14159 / 180)

```


Label- ობიექტია. Caption – თვისებაა. Date – მიმდინარე თარიღი. Rnd * 90 – თვისების მნიშვნელობაა. Rnd აგენერირებს შემთხვევით რიცხვებს [0-1] შუალედში. Int ფუნქცია აბრუნებს არგუმენტის მთელ ნაწილს და უკუაგდებს არგუმენტის წილად ნაწილს. შედეგია რიცხვები 1-დან 90-მდე. Sqr – კვადრატული ფესვის ამოღების ფუნქციაა. Sin ფუნქციის არგუმენტი განიხილება რადიანებში, რისთვისაც გამოყენებულია გრადუსების რადიანებში გარდაქმნის ფუნქცია ანუ გარდაქმნით შემთხვევით რიცხვს რადიანში და მივიღებთ შედეგს.

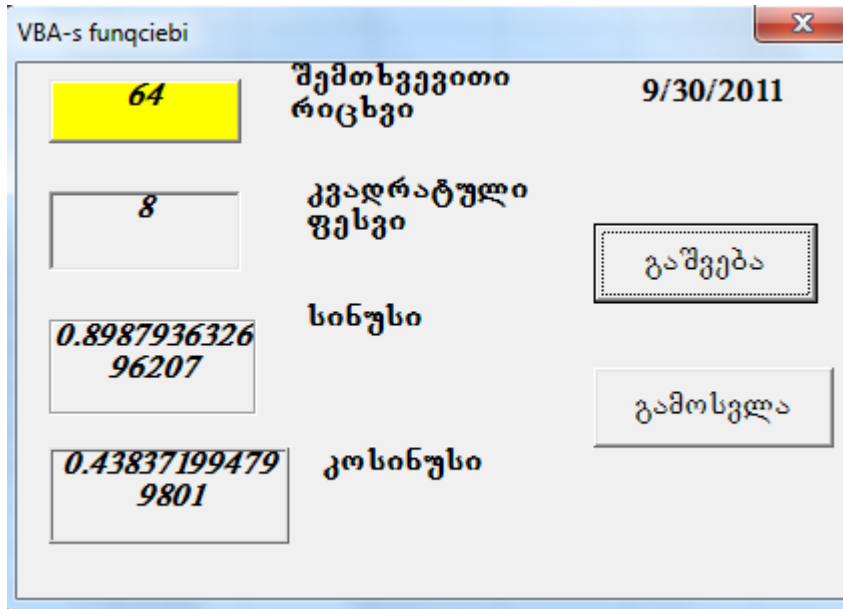
პროგრამის ტექსტი

```

Private Sub CommandButton1_Click()
Label9.Caption = Date
Label1.Caption = Int(Rnd * 90) + 1
Label1.BackColor = RGB(255, 255, 0)
Label2.Caption = Sqr(Label1.Caption)
Label3.Caption = Sin(Label1.Caption * 3.14159 / 180)
Label4.Caption = Cos(Label1.Caption * 3.14159 / 180)
End Sub

```

გამოვიტანოთ ფორმა ეკრანზე  ღილაკზე დაჭერით და შევამოწმოთ როგორ მუშაობს „გაშვება“ ღილაკი. პროგრამის შესრულების შედეგი ნახვენებია სურათზე:



– ესლა დავაპროგრამით „გამოსვლა“ ღილაკი. გადავიდეთ VBA რედაქტორში და ორჯერ დავაწკაპუნოთ „გამოსვლა“ ღილაკზე. გამოჩნდება პროგრამის სათაური. ჩავსვათ ერთი სტრიქონი UserForm1.Hide, სადაც UserForm1 ობიექტია, ხოლო Hide – მეთოდი.

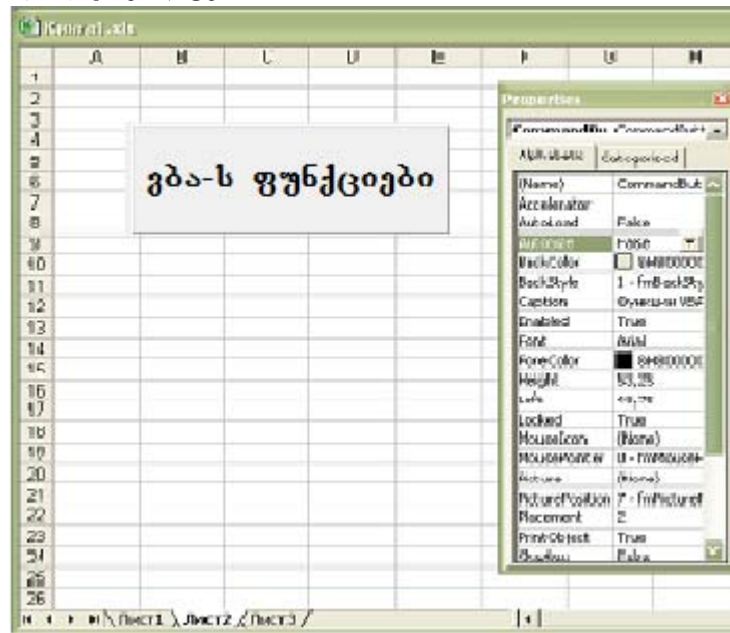
პროგრამის ტექსტი

```
Private Sub CommandButton2_Click()
    UserForm1.Hide
End Sub
```

პროგრამის გაშვება ხდება **F5** ღილაკზე დაჭერით, მაგრამ არსებობს ფორმის ეკრანზე გამოტანის კიდევ ერთი ხერხი. ამისათვის:

- გადავიდეთ Excel დანართის ნებისმიერ გვერდზე.
- მართვის ელემენტების პანელზე ავირჩიოთ **CommandButton** ინსტრუმენტი და მოვხაზოთ ღილაკი Excel-ის ფურცელზე.


- გამოვიდასოთ ამ ელემენტების თვისებები დილაკზე დაჭერით.




- შეარჩიეთ **CommandButton1** ობიექტის თვისებების მნიშვნელობები.

თვისება	მნიშვნელობა
Caption	ვბა-ს ფუნქციები
Font	AcadNusx, მსხვილი, ზომა 14
BackColor (დილაკის ფონის ფერი)	ნებისმიერი
ForeColor (დილაკზე სიმბოლოთა ფერი)	ნებისმიერი

– დააპროგრამეთ ღილაკი „ VBA-ს ფუნქციები“, ამისათვის მონიშნეთ CommandButton1 და ელემენტების

მართვის პანელზე „საწყისი ტექსტი“  ღილაკზე დაჭერით გამოიძახეთ პროგრამის სათაური.

– ჩასვით ერთი სტრიქონი - UserForm1.Show, რაც ნიშნავს UserForm1 ფორმის ეკრანზე გამოჩენას. UserForm1 არის ობიექტი, ხოლო Show – მეთოდი.

– გადადით Excel-ის დანართის ფურცელზე და გამორთეთ კონსტრუქტორის რეჟიმი ელემენტების მართვის პანელიდან  ღილაკით.

– გაუშვით პროგრამა შესრულებაზე.

– შეინახეთ თქვენი სამუშაო.

შესასრულებელი სამუშაო „გამოთვლები“

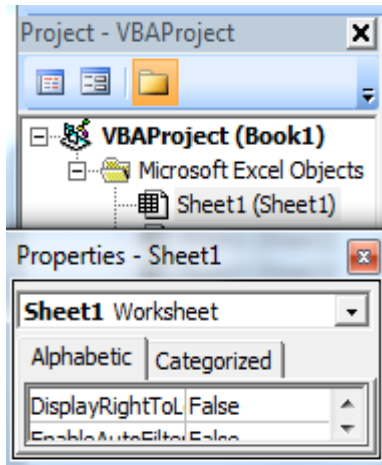
დავალება: დაგაპროექტოთ ფორმა შემთხვევითი რიცხვებისათვის ოთხი არითმეტიკული ოპერაციის: შეკრება, გამოკლება, გამრავლება, გაყოფა, შესასრულებლად.

სამუშაოს მიმდევრობა:

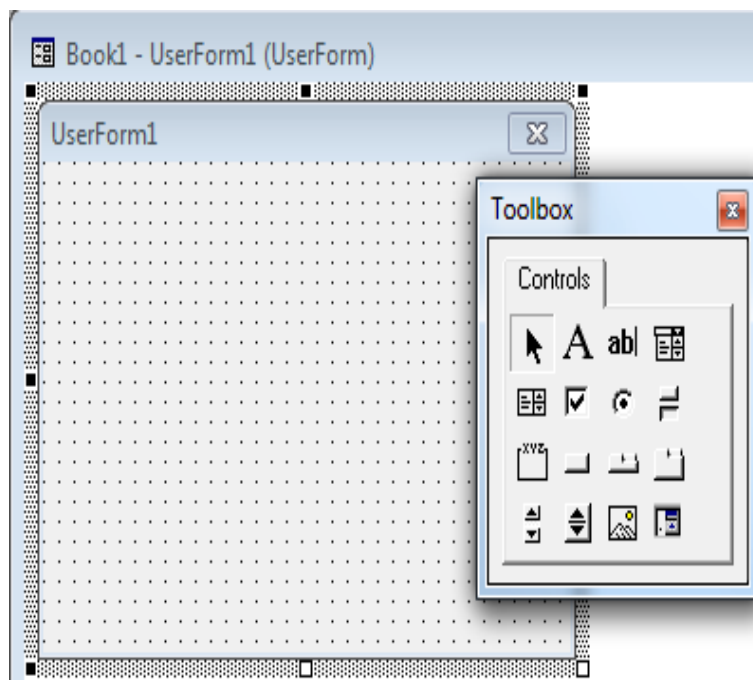
– გაუშვით შესრულებაზე Excel.

– ჩატვირთეთ VBA (Developer -> Visual Basic)

– შეამოწმეთ, რომ ეკრანზე გამოვიდეს პროექტისა და თვისებების ფანჯარა.



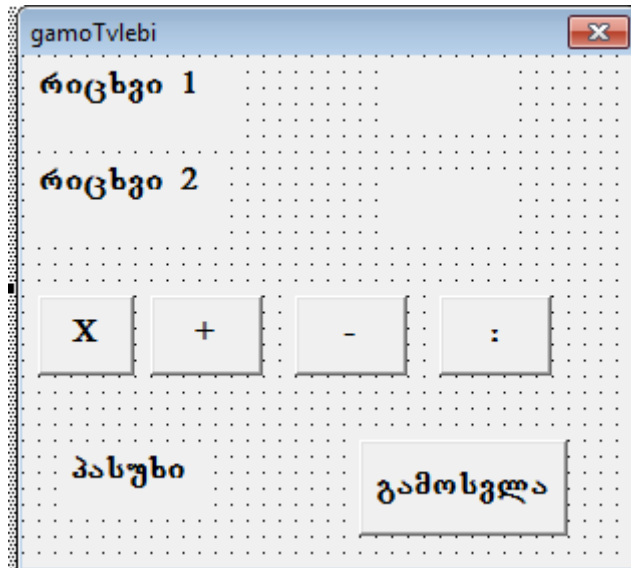
– შესრულეთ ბრძანება **Insert | UserForm**. დაარქვით ფორმას „გამოთვლები“ და გამოიძახეთ „Toolbox“ პანელი.



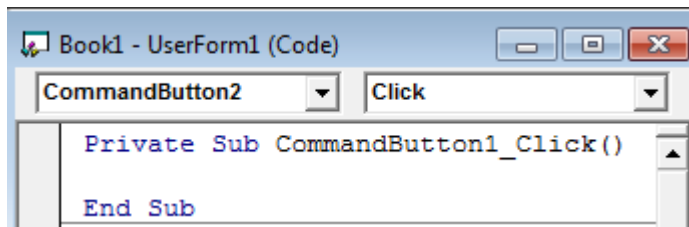
- ფორმაზე დააპროექტეთ 5 „CommandButton” და 5 “Label”.

ობიექტი	წარწერა(Caption)	დანიშნულება
Label1	პირველი რიცხვი	კომენტარი
Label2	მეორე რიცხვი	კომენტარი
Label3	პასუხი	შედეგის გამოსატანად
Label4	(ცარიელი)	პირველი რიცხვის მნიშვნელობა
Label5	(ცარიელი)	მეორე რიცხვის მნიშვნელობა
CommandButton1	X	რიცხვების გადამრავლება
CommandButton2	+	რიცხვების შეკრება
CommandButton3	-	რიცხვების გამოკლება
CommandButton4	:	რიცხვების გაყოფა
CommandButton5	გამოსვლა	ფორმის დახურვა

- ობიექტების თვისებების მნიშვნელობები შეარჩიეთ სურვილისამებრ.
თქვენ მიიღებთ ფორმას:



- ფორმაზე ორჯერ დააწკაპუნეთ „X“ ღილაკზე, პროგრამის სათაურის ეკრანზე გამოსატანად:



პროგრამის სათაური იწყება Private Sub და მთავრდება End Sub. ამ წინადადებებს შორის ჩაწერეთ პროგრამის ტექსტი:

```

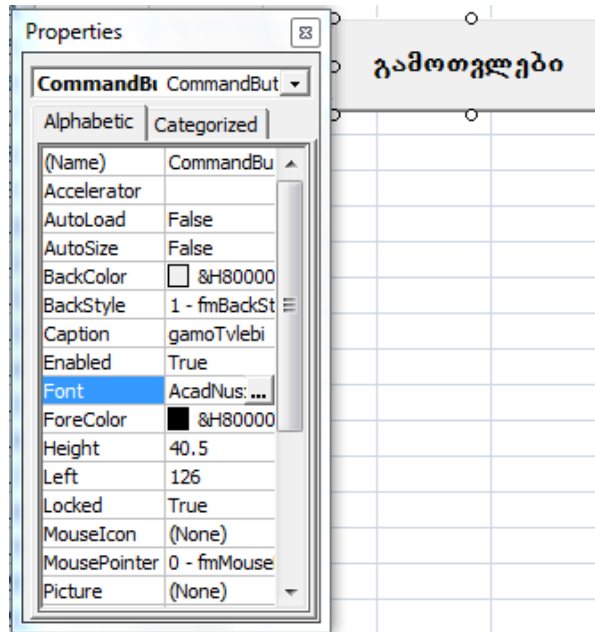
Randomize
Label4.Caption=Int(Rnd*90)+1
Label5.Caption=Int(Rnd*90)+1
Label3.Caption= Label4.Caption* Label5.Caption

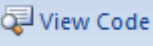

```

- ანალოგიურად „+“, „-“, „:“ და გამოსვლა დილაკებისათვის პროგრამა იქნება შემდეგი სახის:

+	Randomize Label4.Caption=Int(Rnd*90)+1 Label5.Caption=Int(Rnd*90)+1 Label3.Caption=Label4.Caption+Label5.Caption
-	Randomize Label4.Caption=Int(Rnd*90)+1 Label5.Caption=Int(Rnd*90)+1 Label3.Caption=Label4.Caption-Label5.Caption
:	Randomize Label4.Caption=Int(Rnd*90)+1 Label5.Caption=Int(Rnd*90)+1 Label3.Caption=Label4.Caption/Label5.Caption
გამოსვლა	UserForm1.Hide

- შეამოწმეთ ფორმის ქმედითუნარიანობა “Run | Run Sub/UserForm” ბრძანებით, ან F5 დილაკზე დაჭერით.
- გადადით Excel-ში. Sheet1 მუშა ფურცელს გადაარქვით სახელი და დაარქვით „გამოთვლები“.
- მუშა ფურცელზე შექმენით დილაკი ფორმის გამოსაძახებლად და დაარქვით „გამოთვლები“, რისთვისაც მონიშნეთ დილაკი, თავის მარჯვენა დილაკით გამოიძახეთ „თვისებები“ და დააფორმატეთ მონიშნული დილაკი.



- მონიშნეთ დილაკი „გამოთვლები“, დააჭირეთ ელემენტების მართვის პანელზე  დილაკს.
- დაწერეთ ფორმის გამოძახების პროგრამა: `UserForm1.Show`.
- მონიშნეთ დილაკი „გამოთვლები“ და გამორთეთ დიზაინის რეჟიმი  დილაკზე დაწკაპუნებით.
- შეინახეთ თქვენ მიერ შესრულებული სამუშაო.

თემა №5. Image ობიექტთან მუშაობა

განსახილველი საკითხები:

- **Image** ობიექტის თვისებები.
- პროგრამის შედგენა **Image** ობიექტის დასამუშავებლად.

Image(სურათი) ობიექტი. **Image** ობიექტი ემსახურება გრაფიკული ფაილიდან სამომხმარებლო ფორმაზე სურათის განთავსებას.

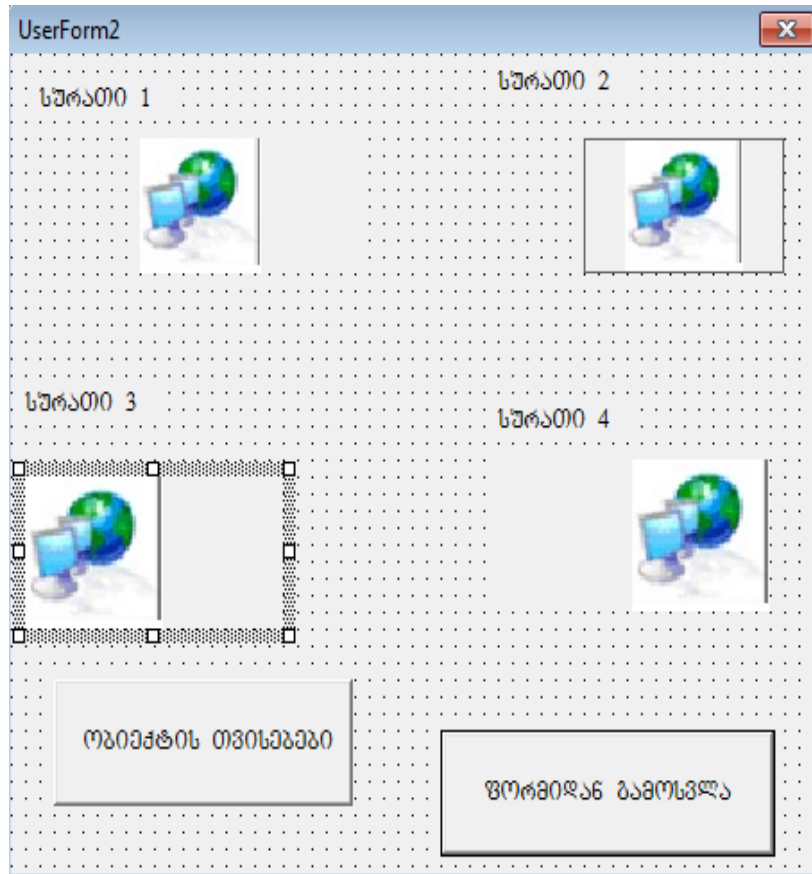
Image ობიექტის თვისებები. მართვის ელემენტი **Image** იქმნება ილუსტრაციების ჯგუფიდან **Picture** დილაკის დახმარებით. იგი გამოიყენება გრაფიკული ფაილების წარმოსადგენად. ქვემოთ მოყვანილ ცხრილში წარმოდგენილია რამდენიმე თვისება, რომელიც გამოიყენება გამოსახულებათა მართვისათვის. **Image** მართვის ელემენტში სურათის ჩასმა შესაძლებელია მხოლოდ პროგრამული გზით.

დავალება: Image ობიექტის თვისებები

მაგალითში განვიხილავთ **Image** ობიექტს და მასზე კონკრეტული თვისების მინიჭებას.

1. შექმნით ეკრანული ფორმა.
2. შექმნილ ფორმაზე გამოიტანეთ ოთხი **Image** ობიექტი. არ გაამახვილოთ ყურადღება **Image** ობიექტის საწყის მდებარეობასა და ზომებზე. (ამ ობიექტთა თვისებები მინიჭებული იქნება დუმილის პრინციპით).
3. ფორმაზე განათავსეთ ასევე, ოთხი Label კომპონენტი, რომელსაც შეუცვალეთ ტექსტი და შესბამისად დაარქვით: სურათი 1, სურათი 2, სურათი 3 და სურათი4.





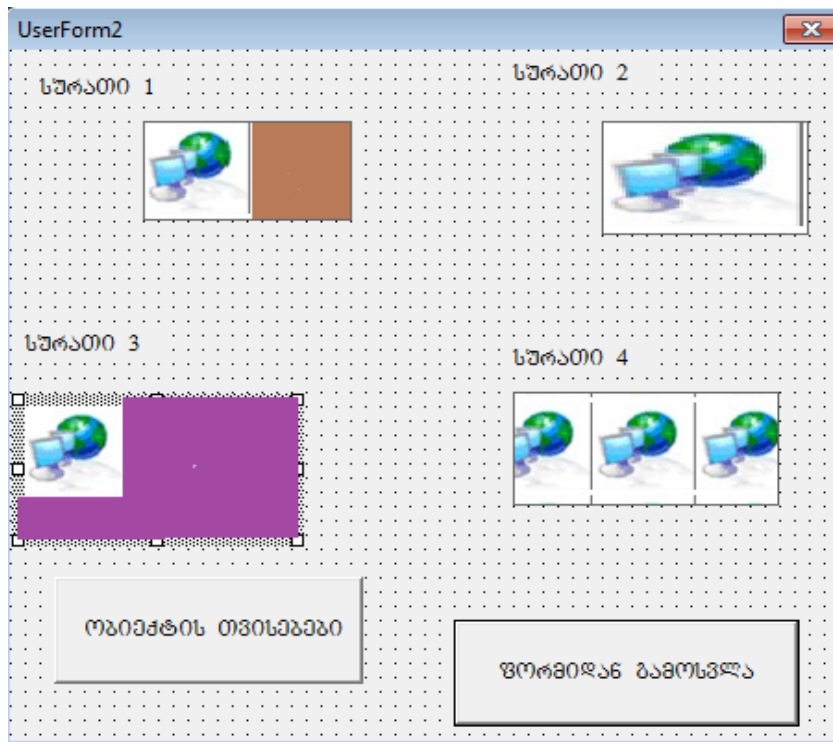
4. სამომხმარებლო ფორმაზე განათავსეთ ორი CommandButton ობიექტი, შესაბამისად ერტს დაარქვით ობიექტის თვისებები, ხოლო მეორეს – ფორმიდან გამოსვლა.
5. ობიექტის თვისებებს მიანიჭეთ მნიშვნელობები ქვემოთ მოყვანილი ცხრილის შესაბამისად.

ოპციონები	მნიშვნელობა
Auto Size (ზომა)	შესაძლო მნიშვნელობები: True (სურათი ავტომატურად იცვლის ზომას). False (მოთხოვნის შესაბამისად)
Back Color	უკანა ფონის ფერი
Border Color	გრაფიკული გამოსახულების ჩარჩოს ფერი
Picture (გრაფიკული ფაილის გამოსახ.)	მითითებული უნდა იყოს გრაფიკული გამოსახულების დასახელება
Picture Alignment (საზღვრების გასწორების რეჟიმი)	უნდა მიეთითოს საზღვრების გასწორების პოზიციები 0-fmPicture AlignmentTopLeft (ზედა მარცხენა მხარე) 1-fmPicture AlignmentTopRight (ზედა მარჯვენა მხარე) 2- fmPicture AlignmentCenter (ცენტრირება) 3- fmPicture AlignmentBottomLeft (ქვედა მარცხენა მხარე) 4- fmPicture AlignmentBottomRight (ქვედა მარჯვენა მხარე)
PictureSize Mode (სურათის წარმოდგენის /აღქმის რეჟიმები)	0-fmPictureSizeModeClip (სურათის ნაწილი მოჭრილია) 1-fmPictureSizeModeStretch (სურათის წარმოდგენა მთელ მოცულობაზე) 3- fmPictureSizeModeZoom (სურათის წარმოდგენა რეალური ზომით)
Picture Tiling (როდესაც სურათის ზომა მეტია მოც.ზე)	True (მოცულობის ზედა ფონი ივსება სურათით). False (ხეულებრივი მდგომარეობა, სურათი განთავსებულია მოცულობის ზედა ფონზე)
Specialeffect (მართვის ელემენტების გააქტიურება)	არსებული ეფექტების გააქტიურება

თვისება	მნიშვნელობა
ფორმა	
Width	455
Height	360
Caption	ობ.თვისებები
სურათი 1	
Left	55
Height	42
Width	85
Top	28
სურათი 2	
Left	240
Height	42
Width	85
Top	28
სურათი 3	
Left	55
Height	42
Width	85
Top	150
სურათი 4	
Left	55
Height	42
Width	85
Top	150
ფილაპი „ობ.თვისებები“	
Left	255
Height	45
Width	125
Top	210
ფილაპი „ზუსტა“	
Left	180
Height	45
Width	140
Top	240

ობიექტზე თვისებების მინიჭების პროცესი აისახება პროგრამაში.

სურათზე წარმოდგენილია ობიექტზე მინიჭებული თვისებების შესაბამისი მდგომარეობა:



- პირველ შემთხვევაში გამოსახულება მოცულობაში წარმოდგენილია პროპორციულად მასშტაბირებულად;
- მეორე შემთხვევაში გამოსახულება ასევე წარმოდგენილია სრულად მაგრამ აქ დარღვეულია პროპორციულობა;
- მესამე შემთხვევაში გამოსახულება მოცულობაში განთავსებულია შემთხვევითად;
- მეოთხე შემთხვევაში ობიექტის მოცულობა სურათით შეესებურება მოზაიკისებურად.

ასევე ნახაზე წარმოდგენილ პირველ და მესამე სურათზე გამოტანილია უკანა ფონი, ოთხივე შემთხვევაში

კი გამოსახულება გასწორებულია ზემოთა მარცხენა კუთხეში

მაგალითი

ქვემოთ მოცემულია პროგრამა, რომელიც უზრუნველყოფს „ობიექტის თვისებები“ დილაგის ფუნქციონირებას:

```
Private Sub CommandButton1_Click()
```

Rem ობიექტის განთავსება ფორმის მარცხენა საზღვრიდან

```
Image1.Left = 25
```

```
Image2.Left = 225
```

```
Image3.Left = 25
```

```
Image4.Left = 225
```

Rem ობიექტის განთავსება ფორმის ზედა საზღვრიდან

```
Image1.Top = 30
```

```
Image2.Top = 30
```

```
Image3.Top = 170
```

```
Image4.Top = 170
```

Rem ობიექტის სიგანე

```
Image1.Width = 160
```

```
Image2.Width = 160
```

```
Image3.Width = 160
```

```
Image4.Width = 160
```

Rem ობიექტის სიმაღლე

```
Image1.Height = 85
```

```
Image2.Height = 85
```

```
Image3.Height = 85
```

```
Image4.Height = 85
```

Rem ობიექტის ზომისა და სურათის ზომის თანხვედრა

```
Image1.PictureSizeMode = 3
```

```
Image2.PictureSizeMode = 1  
Image3.PictureSizeMode = 0  
Image4.PictureSizeMode = 0
```

```
Rem სურათის გამრავლება ობიექტის ზედა ნაწილში  
Image4.PictureTiling = True
```

```
Rem ორ ობიექტზე ფონის ფერის შეცვლა  
Image1.BackColor = RGB(100, 100, 100)  
Image3.BackColor = RGB(100, 50, 100)
```

```
Rem ობიექტებზე ეფექტების მინიჭება  
Image1.SpecialEffect = 1  
Image2.SpecialEffect = 2  
Image3.SpecialEffect = 3  
Image4.SpecialEffect = 6
```

```
End Sub
```

6. შექმენით დილაკი, **Excel**-ის მუშა ფურცელზე, რომელიც უზრუნველყოფს ეკრანზე ფორმის გამოტანას.
7. შეინახეთ სამუშაო..

პროგრამის ტესტირება და გამართვა.

პროგრამის გამართვა და შეცდომების აღმოფხვრა არის პროგრამაზე მუშაობის განუყოფელი ნაწილი. ყველა პროგრამულ ენას აქვს თავისი ინსტრუმენტი, რომელიც ამოწმებს პროგრამას გაშვების წინ, და მომხმარებელს აძლევს შეტყობინებას ამა თუ იმ შეცდომის შესახებ. VBA პროგრამასაც აქვს ინსტრუმენტული საშუალება, რომელიც ახდენს პროგრამირების პროცესში დაშვებულ შეცდომათა იდენტიფიცირებას და მომხმარებელს აძლევს შეტყობინებას ამ შეცდომათა შესახებ.

მაგალითად, სინტაქსური შეცდომები, რომელიც შეიძლება დაშვებული იყოს პროგრამის ტექსტში, ფუნქციისა და ცვლადების დასახელებაში, ასევე ლოგიკური შეცდომები, ფაილების გამოტოვება, ტექნიკური პრობლებით გამოწვეული შეცდომა და ა.შ.

პროგრამისტის მიზანია, რაც შეიძლება ნაკლები შეცდომა დაუშვას და პროგრამამ იმუშაოს გამართულად. **VBA** ენა მომხმარებელს სთავაზობს თანამედროვე საშუალებებს, რომელიც უზრუნველყოფს ტესტირების პროცესისი შერულებას და შეცდომათა დროულ აღმოფხვრას.

ტესტირება – ეს არის პროცედურათა შესრულების პროცესი და მისი მუშაობის ყველა ასპექტის გამოკვლევა. ტესტირების მიზანია პროცედურათა შერულებისას მიღებული შედეგების სისწორის და მომხმარებლის მიერ ამ შედეგების გამოყენების სრულფასოვნების შემოწმება.

ყველა შესაძლო დაშვებული შეცდომები ერთიანდება სამ ჯგუფად:

1. **კომპილაციის შეცდომა.** ასეთი სახის შეცდომები თავს იჩენს იმ შემთხვევაში, როდესაც ენა ვერ ახდენს შეტანილი ტექსტის იტერპრეტირებას, მაგალითად არასწორად გამოყენებულია სინტაქსის ინსტრუქცია ან არასწორადაა ჩაწერილი მეთოდის ან თვისების დასახელება. კომპილაციის რამდენიმე სახის შეცდომა თავს იჩენს ინსტრუქციის შეტანისთანავე, ხოლო ზოგიერთი კი პროგრამის გაშვების წინ. ასეთი სახის შეცდომები იდენტიფიცირდება და მომხმარებელს მიეწოდება შეტყობინება.

2. **პროგრამის გაშვების შეცდომა.** ასეთი სახის შეცდომები თავს იჩენს კომპილაციის შემდეგ პროგრამის შესრულებისას, რომელთა მიზეზიც შესაძლოა იყოს მონაცემთა გამოტოვება ან არასწორი ინფორმაცია (მაგ. მომხმარებლის მიერ შეტანილი მონაცემები). **VBA** ენაში

გაშვების შეცდომა ისევე, როგორც კომპილაციის შეცდომა იოლად იდენტიფიცირდება. მომხმარებელს მიეწოდება ინსტრუქცია, სადაც აღწერილია შეცდომის დაშვების სავარაუდო პირობები.

3. ლოგიკური შეცდომები. ყველაზე რთულად შესამჩნევი და აღმოსაფხვრელი არის დაშვებული ლოგიკური შეცდომები. ლოგიკური შეცდომები არ აისახება კომპილაციის ან პროგრამის გაშვების პროცესზე იგი თავს იჩენს, როდესაც პროგრამა მომხმარებელს არ აძლევს სასურველ შედეგს. ასეთი სახის შეცდომა მოითხოვს საფუძვლიან შემოწმებას.

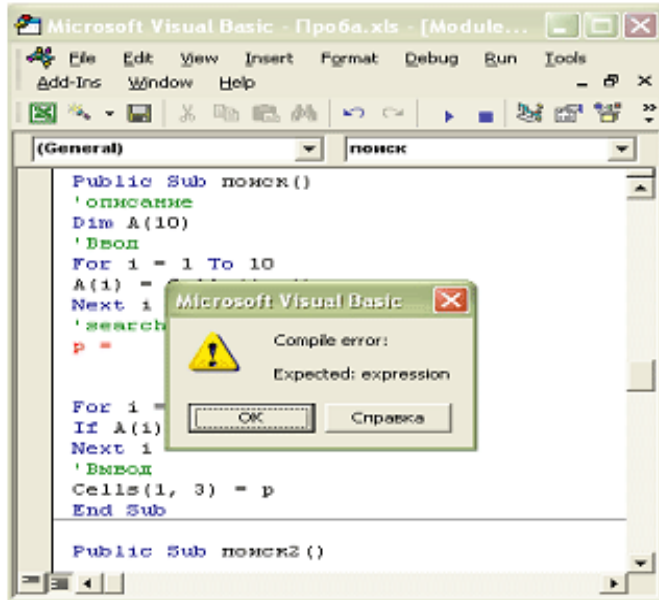
კომპილაცია – კომპილაციის დროს ხდება ალგორითმის ენაზე დაწერილი პროგრამის მანქანურ კოდებში გადაყვანა. როდესაც პროგრამაში დაშვებულია სინტაქსური შეცდომები კომპილაციის პროცესი წყდება და ის სტრიქონი, სადაც შეცდომაა დაშვებული შეფერადდება ყვითლად და მომხმარებელი დებულობს შესაბამის შეტყობინებას. პროგრამის გაშვების პროცესის გასაგრძელებლად აუცილებელია გასწორდეს შეცდომა და გააქტიურდეს დილაკი:

რომელიც განთავსებულია **VBA** რედაქტორის სტანდარტულ პანელზე. ასევე შეძლებელია პროგრამირების პროცესის შეწყვეტა და დილაკის გააქტიურება, შემდეგ სწორდება პროგრამა და თავიდან ხდება გაშვება.

შეცდომის აღმოჩენის შემთხვევაში კომპილატორი იძლევა შეტყობინებას იმ სტრიქონის შესახებ, რომელშიც შეცდომაა დაშვებული. ყველაზე ხელსაყრელი შეცდომის გასასწორებლად არის **VBA** რედაქტორის საცნობარო სისტემა, რომელიც ეხმარება მომხმარებელს, დროულად განისაზღვროს შეცდომის ხასიათი და გასწორდეს.


▶ Contin

Reset

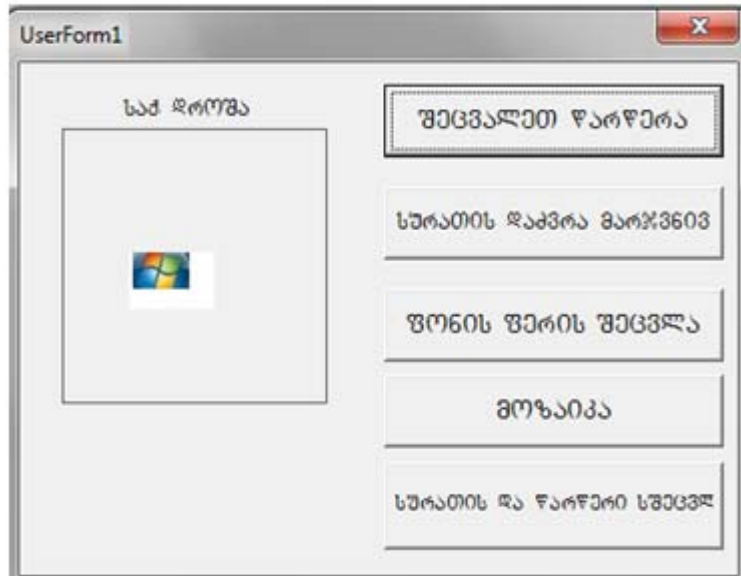


მომდევნო პრაქტიკულ დავალებაში განხილული იქნება პროგრამა, რომელშიც დაშვებულია შეცდომა.

დავალება: პროგრამის გასწორება

1. **Excel** პროგრამაში გახსენით ახალი სამუშაო წიგნი.
2. მოამზადეთ ეკრანული ფორმა. ფორმაზე განათავსეთ **Toolbox** –იდან **Image**  ობიექტი. (სასურველია პატარა ზომის სურათი).

ზუსტად განსაზღვრეთ, ფორმაზე მიმაგრებული გრაფიკული ფაილის აღწერის ეტაპები.



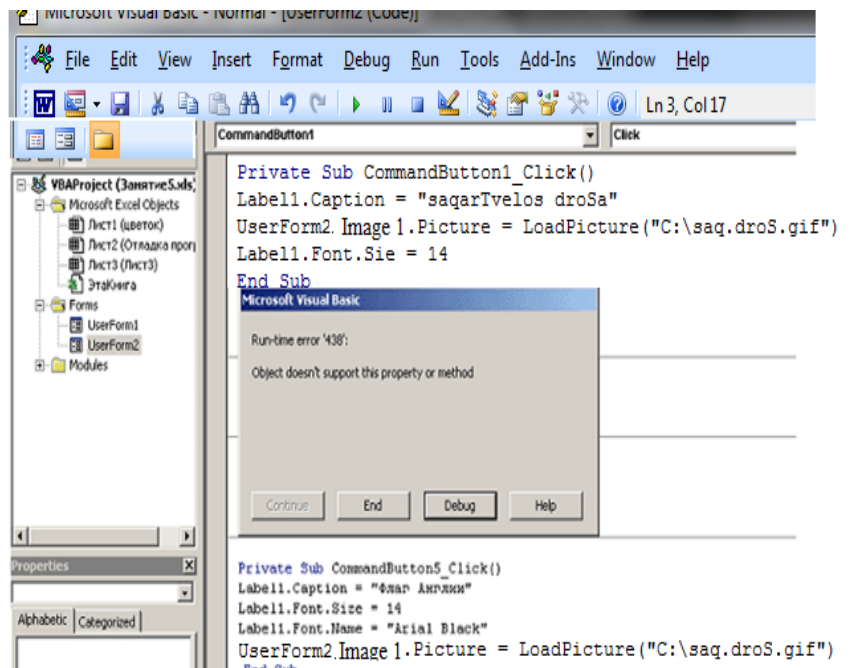
3. დილაკისათვის „შეცვალეთ წარწერა“ შექმენით ახალი პროცედურა.
4. შეიტანეთ პროცედურის ტექსტი, სადაც შეცდომა იქნება დაშვებული, მაგალითად **Size** (ფონტის ზომა) ნაცვლად ჩაწერეთ **Size**:

```
Private Sub CommandButton1_Click()  
Label1.Caption = "saqarTvelos droSa"
```

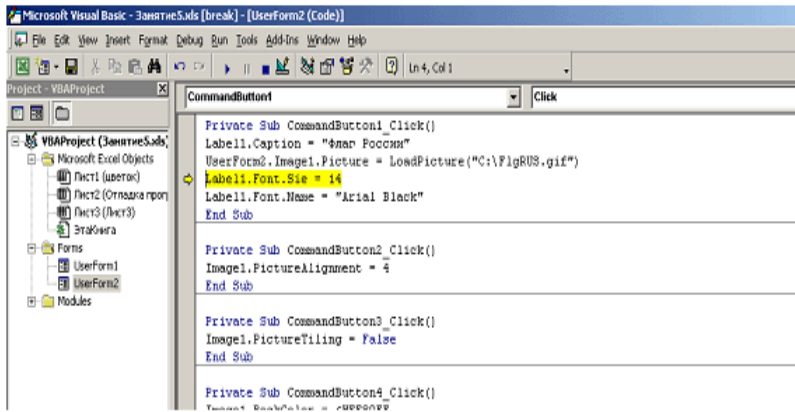


```
UserForm2.Image1.Picture = LoadPicture("C:\saq.droS.gif")
Label1.Font.Sie = 14
End Sub
```


5. ვბრუნდებით ფორმის შექმნის რედაქტორში და ვააქტიურებთ ღილაკს: .
6. ეკრანზე ფორმის გამოჩენის შემდეგ ვააქტიურებთ ღილაკს „შეცვალეთ წარწერა“. რადგან პროგრამა დაწერილი იყო შეცდომით (ნაცვლად **Size** გვეწერა **Sie**), ამიტომ გამოვა დაშვებულ შეცდომაზე, შეტყობინება და ისევ ვააქტიურდება VBA რედაქტორის ფანჯარა.



7. ვაააქტიურებთ ღილაკი “**Debug**” (გამართვა), რის შემდეგაც სტრიქონი, სადაც დაშვებულია შეცდომა შეფერადდება ყვითლად.



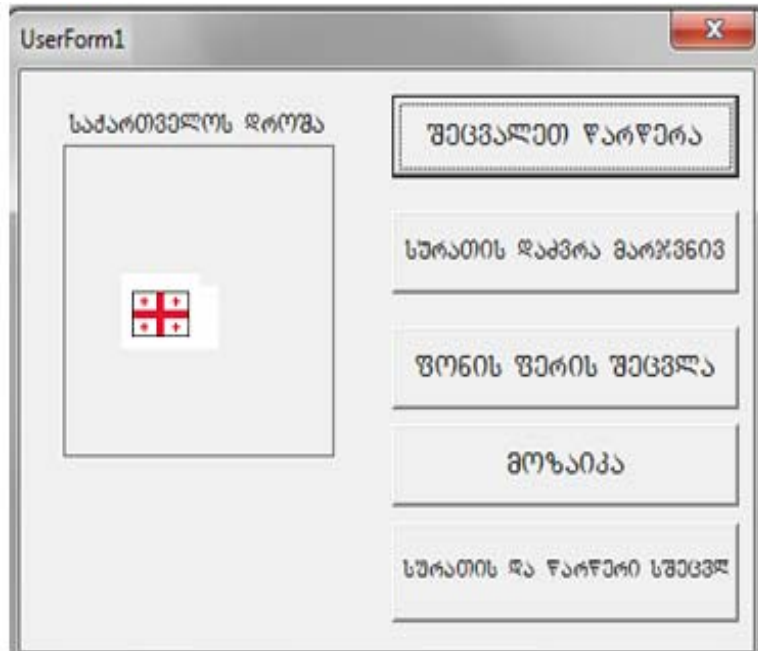
8. გაასწორეთ შეცდომა და გააქტიურეთ სტანდარული

ინსტრუმენტების პანელიდან ღილაკი  (“გაგრძელება”). ცხრილში წარმოდგენილია პროგრამის ტექსტი ღილაკებისათვის: CommandButton2, CommandButton3, CommandButton4, CommandButton5

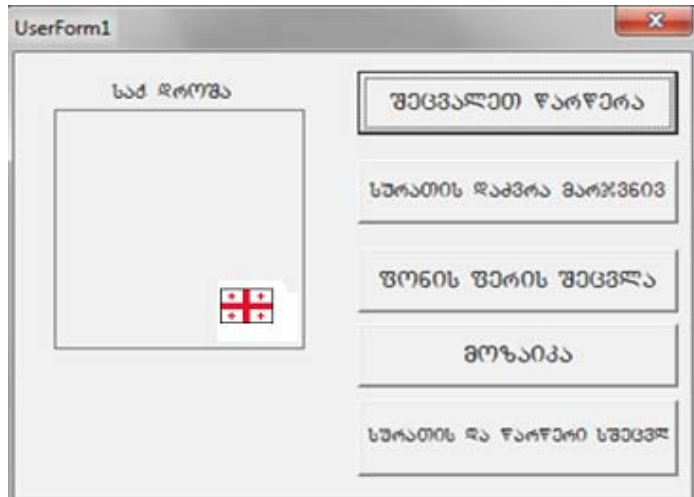
ობიექტი	პროგრამა
CommandButton2 (სურათის დაძვრა მარჯვნივ)	Private Sub CommandButton2_Click() Image1.PictureAlignment = 4 End Sub
CommandButton4 (შენიშნის შერის შეცვლა)	Private Sub CommandButton4_Click() Image1.BackColor = &HFF80FF UserForm2.BackColor = RGB(64, 0, 0) End Sub
CommandButton3 (მოხაიკა)	Private Sub CommandButton3_Click() Image1.PictureTiling = True End Sub
CommandButton5 (სურათის და წარწერის შეცვლა)	Private Sub CommandButton5_Click() Label1.Caption = "Флаг Англии" Label1.Font.Size = 14

```
Label1.Font.Name = "Arial Black"  
UserForm2.Image1.Picture =  
LoadPicture("C:\FlgEng.gif")  
End Sub
```

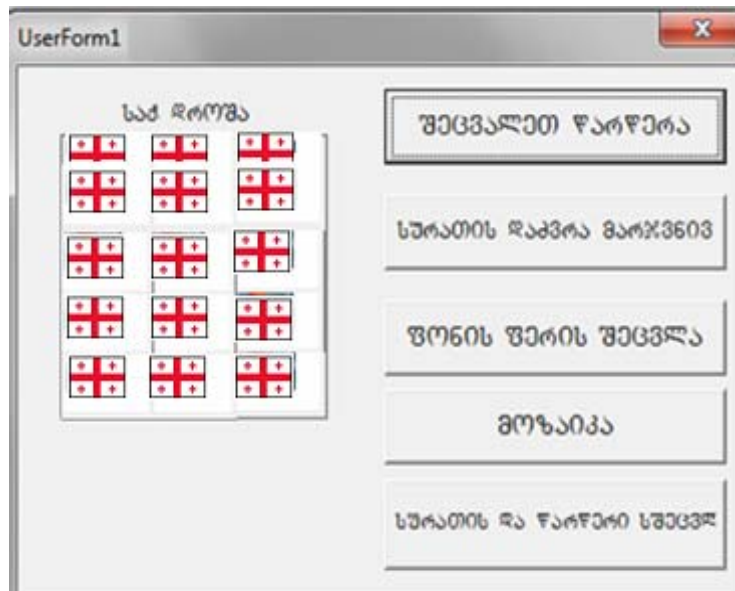
9. „შეცვალეთ წარწერა“ ღილაკის გააქტიურების შემდეგ ფორმა მიიღებს სახეს:



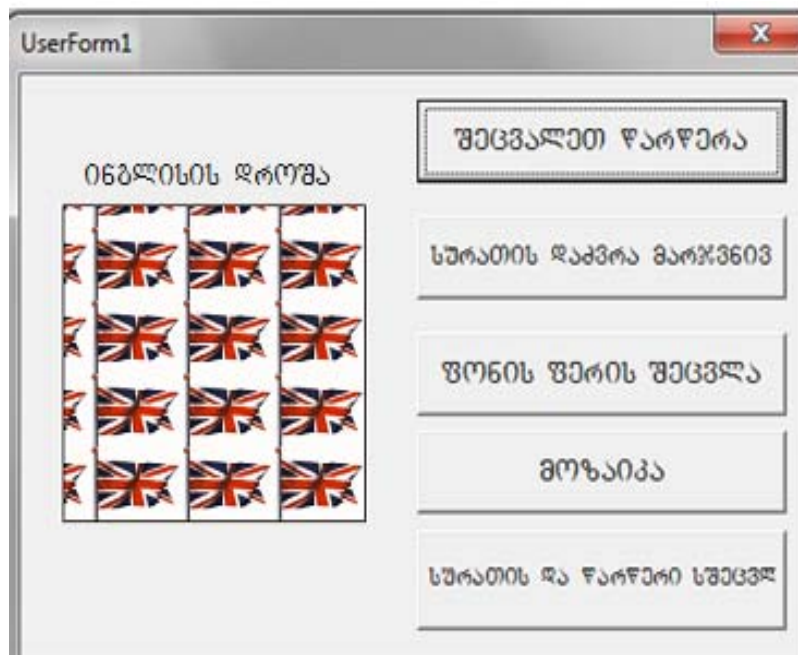
10. „სურათის დაძვრა მარჯვნივ“ ღილაკის გააქტიურების შემდეგ ფორმა მიიღებს სურათზე წარმოდგენილ სახეს:



11. „მოზაიკა“ ლილაკის გააქტიურების შემდეგ ფორმაზე წარმოდგენილი გრაფიკული გამოსახულება მიიღებს სახეს:



12. „სურათის და წარწერის შეცვლა“ დილაკზე დაჭერისას ფორმა მიიღებს სახეს:

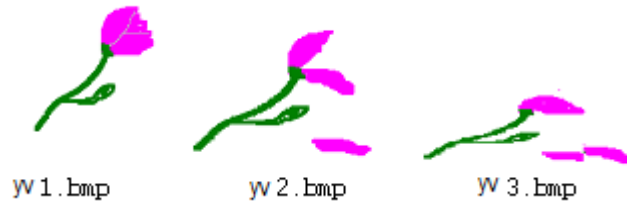


ასევე შესაძლებელია სურათებისა და წარწერების სხვადასხვა კომბინაციათა განხილვა.

13. შეინახეთ შესრულებული სამუშაო.

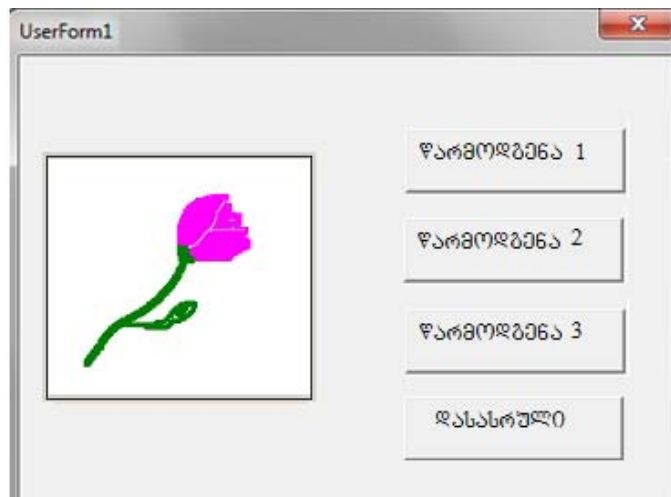
დავალება: სურათებთან მუშაობა

1. MS Paint რედაქტორში შექმენით ქვემოთ მოცემული სურათები:



2. შეინახეთ სურათები თქვენს საქაღალდეში ან C: დისკზე :
C:\yv1.bmp, C:\yv2.bmp, C:\yv3.bmp.

3. Excel – ში შექმენით ეკრანული ფორმა და დაარქვით სახელი „ქვაილი“. ფორმას მიამაგრეთ სურათი და ოთხი ღილაკი. ღილაკებს დააწერეთ სახელები: „წარმოდგენა1“, „წარმოდგენა2“, „წარმოდგენა3“, „დასასრული“. გამოიყენეთ ობიექტის თვისებები და დააფორმატეთ მოცემული ფორმა, რის შემდეგაც იგი მიიღებს სახეს:



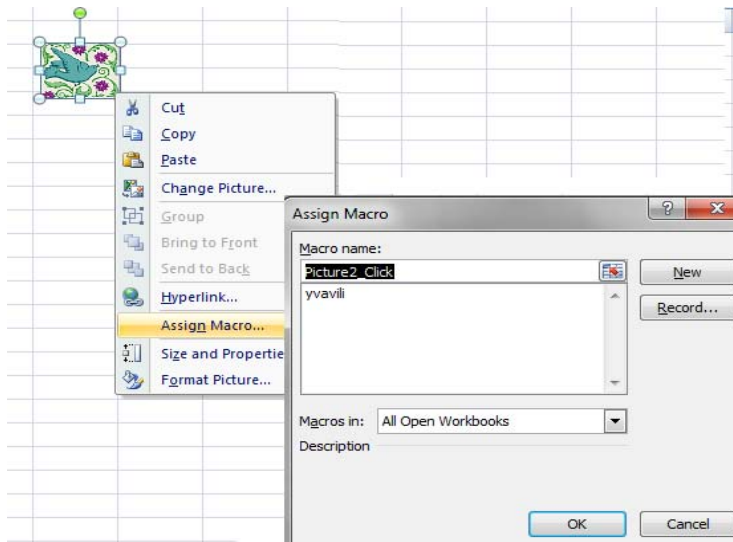
4. ცხრილში წარმოდგენილია პროგრამის ტექსტი შესაბამისი დილაკებისათვის:

ობიექტი	პროგრამა
CommandButton1 ჩასვით სურათი C:\yv1.bmp	Private Sub CommandButton1_Click() UserForm1.Image1.Picture = LoadPicture("C:\yv1.bmp") End Sub
CommandButton2 ჩასვით სურათი C:\yv2.bmp	Private Sub CommandButton2_Click() UserForm1.Image1.Picture = LoadPicture("C:\yv2.bmp") End Sub
CommandButton3 ჩასვით სურათი C:\yv3.bmp	Private Sub CommandButton3_Click() UserForm1.Image1.Picture = LoadPicture("C:\yv3.bmp") End Sub
CommandButton4 დასასრული	CommandButton4 Private Sub CommandButton4_Click() UserForm1.Hide End Sub

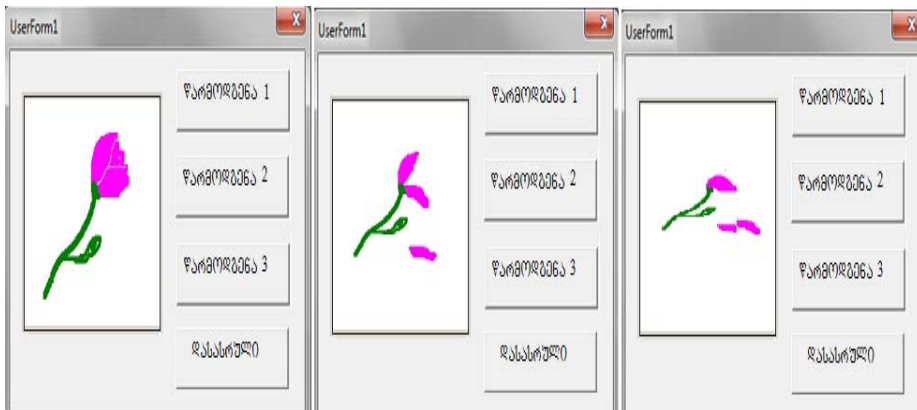
5. დავწეროთ მოცემული ფორმის გამოსაძახებელი პროცედურა. დავარქვათ სახელი „ყვავილი“:

```
Public Sub yvavili()  
UserForm1.Show  
End Sub
```

6. Excel –ის ფურცელზე ჩასვით სურათი და მიამაგრეთ მაკროსი „ყვავილი“. მიიღებთ დილაკს, რომელსაც აქვს სურათის ვიზუალური ფორმა.



7. შეამოწმეთ დილაკის მუშაობა. დილაკის გააქტიურების შემდეგ უნდა მიიღოთ ფორმა „ყვავილი“. ფორმაზე წარმოდგენილი დილაკების გააქტიურების შემდეგ, სურათი იცვლის სახეს და მიიღება ქვემოთ წარმოდგენილი სცენარები. ბოლო დილაკი კი უზრუნველყოფს ფორმის დახურვას.



თემა №6. MsgBox და InputBox ობიექტებთან მუშაობა

განსახილველი საკითხები:

- MsgBox შეტყობინების ფანჯარის გამოტანა
- InputBox მონაცემთა შეტანის ფანჯარა
- ტექსტური სტრიქონების გაერთიანება.
შეტყობინების ფანჯარის შექმნის ფუნქცია იწერება

შემდეგი სახით:

MsgBox (შეტყობინება [ატრიბუტი] [დასახელება]).

MsgBox გამოიყენება სპეციალურ ფანჯარაში შეტყობინების გამოსატანად. შეტყობინება არის დიალოგურ ფანჯარაში გამოსახული ტექსტი, რომელიც წარმოადგენს საჭირო არგუმენტს. ეს სტრიქონი აუცილებლად მოთავსებული უნდა იყოს ბრჭყალებში. ტექსტი შეიძლება შეიცავდეს 1024 სიმბოლოს. გამოყენებული მრავალი ფრჩხილები მიუთითებს, რომ **MsgBox** არის ფუნქცია.

ატრიბუტი განსაზღვრავს ფანჯარაში წარმოდგენილ სხვადასხვა სახის ღილაკებსა და ნიშნულებს. ატრიბუტის არგუმენტები უზრუნველყოფს შეტყობინების ფანჯარას მართოს შემდეგი პარამეტრები: ფანჯარაში ღილაკების რაოდენობა, ფანჯარაში გამოსახული პიქტოგრამები, ღილაკების ტიპები და მათი განთავსება ფანჯარაში.

არგუმენტი არის მთელი რიცხვი და სასურველი შემდეგის მისაღებად იგი შეიძლება წარმოვადგინოთ, როგორც ორი შესაკრები:

ატრიბუტი = პარამეტრი1 + პარამეტრი2.





პარამეტრი1- ის მნიშვნელობა შეტყობინების ფანჯარაში განსაზღვრავს ღილაკის (პიქტოგრამის) სახეს და რიცხვით კოდს. ქვემოთ მოცემულ ცხრილში მოცემულია პარამეტრის შესაძლო მნიშვნელობები:

პარამეტრი1	დილაკები
0	დილაკი "OK"
1	დილაკი "OK" და "Cancel"
2	დილაკი „Abort“(შეწყვეტა), „Retry“ (გამეორება) და „Ignore“ (უარყოფა)
3	დილაკი „Yes“ (დიახ), „No“ (არა), "Cancel" (გაუქმება)
4	დილაკი „Yes“ (დიახ), „No“ (არა)
5	დილაკი „Retry“ (გამეორება), "Cancel" (გაუქმება)

იმ შემთხვევაში, თუ არგუმენტის ატრიბუტები არ არის მოცემული, მაშინ **VBA** პროგრამა გვთავაზობს, რომ დიალოგურ ფანჯარაში შეტყობინებას თან ახლდეს მხოლოდ დილაკი "OK".

პარამეტრი 2 - ის მნიშვნელობა განსაზღვრავს შეტყობინების სახეობას და პიქტოგრამების კომპლექტის კოდს, რომელიც მოთავსებულია შეტყობინების ფანჯარაში.

სხვადასხვა ტიპის შეტყობინების ფანჯრების შესაბამის პიქტოგრამებს და რიცხვით კოდებს შემდეგი სახე აქვს:

პარამეტრი2	შეტყობინების სახეობა	პიქტოგრამა
16	ნიშანი გამოსახავს კრიტიკულ შეტყობინებას	
32	გამოსახება კითხვითი ნიშანი (შეკითხვა)	
48	გამოსახება ძახილის ნიშანი (გაფრთხილება)	
64	გამოსახება ნიშანი, რომელიც წარმოადგენს შეტყობინებას ინფორმაციაზე	

შეტყობინების ფანჯარაში პირველ სტრიქონზე წარმოდგენილია ფანჯრის სახელი . ეს არგუმენტი თუ გამოტოვებულია, მაშინ სათაურის სტრიქონი მიიღებს დასახელებას “Microsoft Excel”.

როგორც აღვნიშნეთ **MsgBox** არის ფუნქცია, ამიტომ მას ახასიათებს უკუქცევითობა (იბრუნებს მნიშვნელობებს). სინტაქსურად **MsgBox** ემთხვევა ოპერატორ **MsgBox** დასახელებას. დაბრუნებული მნიშვნელობა დამოკიდებულია მომხმარებლის მიერ გამოყენებულ ღილაკზე. ცხრილში მოცემულია MsgBox ფუნქციის მნიშვნელობათა ჩამონათვალი.

რიცხვითი კოდი	ღილაკები
1	“OK”
2	“Cancel”
3	„Abort”
4	„Retry“
5	„Ignore“
6	„Yes“
7	„No“

ინფორმაციის შეტანის ფანჯარა

მონაცემების შეტანის ფანჯარა, ისევე, როგორც შეტყობინებათა ფანჯარა შესაძლოა გამოსახული იყოს ოპერატორით ან ფუნქციით.

InputBox ფუნქცია გამოიყენება რიცხვითი ან ტექსტური მონაცემების შესატანად. ეს ფუნქცია ასახავს მონაცემთა შეტანის დიალოგურ ფანჯარას და შეიცავს მონაცემთა შეტანის ველს. იგი სინტაქსურად ჩაიწერება შემდეგი სახით:

InputBox(შეტყობინება[, სათაური] [,მნიშვნელობა]).

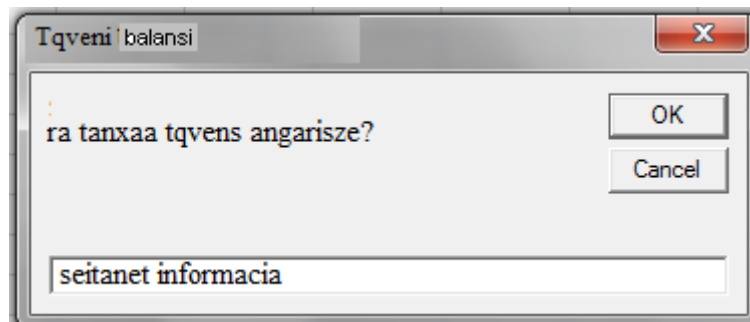
შეტყობინება - არის ერთადერთი აუცილებელი არგუმენტი; იგი ივსება მომხმარებლის მიერ და წარმოადგენს შესატანი მონაცემის მოკლე აღწერას.

სათაური - სათაური არის ტექსტი, რომელიც მოთავსდება დიალოგის ფანჯრის სათაურის სტრიქონში.

მნიშვნელობა - მონაცემთა შეტანის ფანჯარის ლურჯ ფონზე აისახება ჩუმათობის პრინციპით და დარჩება მანამდის სანამ მომხმარებელი არ ჩაანაცვლებს სხვა მნიშვნელობით. ან შესაძლებელია საერთოდ არ იყოს ეს არგუმენტი მინიჭებული და იყოს სიცარიელე ფანჯრის იმ ნაწილში.

MsgBox - ისგან განსხვავებით აქ ატრიბუტების პარამეტრი გამოტოვებულია, ამგვრამ არის ჩუმათობის პარამეტრი, რომელშიც მოცემული სტრიქონი ჩუმათობის პრინციპით აღმოჩნდება ტექსტურ ველში. "OK" ღილაკის გააქტიურების შემდეგ ველს მიენიჭება ცვლადი, ხოლო "Cancel" ღილაკის გააქტიურების შემდეგ სტრიქონი უბრუნდება ნულოვან მნიშვნელობას.

InputBox ფუნქცია აბრუნებს სტრიქონს, ხოლო **MsgBox** ფუნქცია აბრუნებს მთელი ტიპის მნიშვნელობებს.



ტექსტური სტრიქონების გაერთიანება

რამდენიმე ტექსტური სტრიქონის ერთ სტრიქონად წარმოსადგენად იყენებენ კონკატენაციის სიმბოლოს "&" (ამპერსანტი) ან "+" (პლიუსი). მისი გამოყენებით შესაძლებელია რიცხვითი ან სიმბოლური მნიშვნელობების გაერთიანება.

დავალება: შეტყობინების ფანჯრის შექმნა

დააპროექტეთ შეტყობინების ფანჯარა **MsgBox** დახმარებით.

1. გახსენით **Excel** პროგრამა, გადაარქვით **Sheet1** სახელი და დაარქვით „შეტყობინება“.
2. სამუშაო ფურცელზე შექმენით ღილაკი „შეტყობინება“.
3. ღილაკს მიაბით პროგრამა, რის შედეგადაც მიიღება სხვადასხვა შეტყობინებათა ფანჯარა. მაგალითად:

```
Private Sub CommandButton1_Click()
```

```
MsgBox "gaasufTavebT saqaRaldes?", 32 + 4, "saqaRaldis  
gasufTaveba"
```

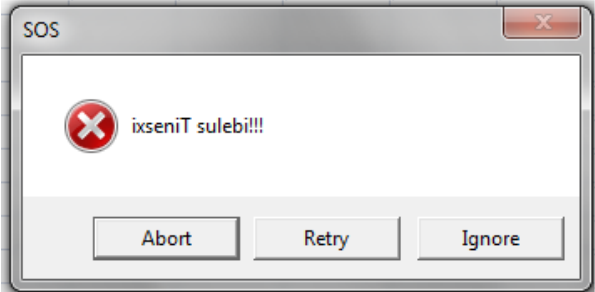
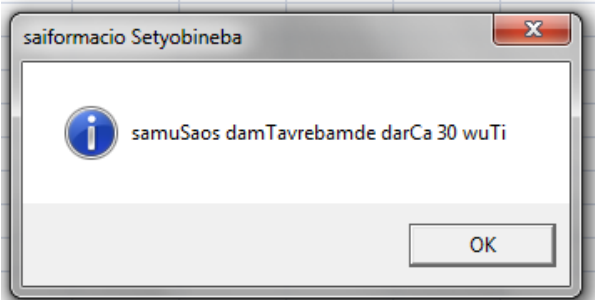
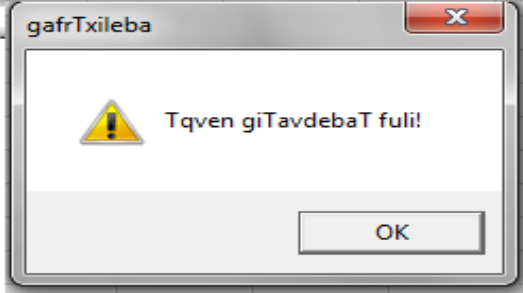
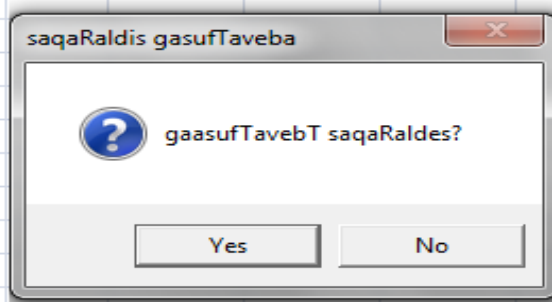
```
MsgBox "Tqven giTavdebaT fuli!",
```

```
48, "gafrTxileba"
```

```
MsgBox "samuSaos damTavrebamde darCa 30 wuTi", 64,  
"saiformacio Setyobineba"
```

```
MsgBox "ixseniT sulebi!!!", 18, "SOS"
```

```
End Sub
```



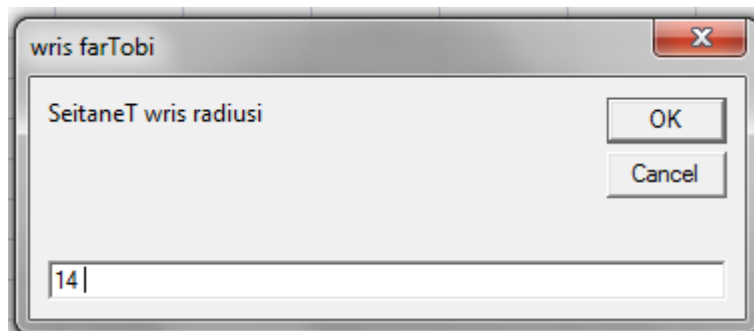
4. დააპროექტეთ რამდენიმე შეტყობინების ფანჯარა **MsgBox** – ის გამოყენებით.
5. შეინახეთ თქვენს მიერ შესრულებული სამუშაო.

დავალება: წრის ფართობის გამოთვლა

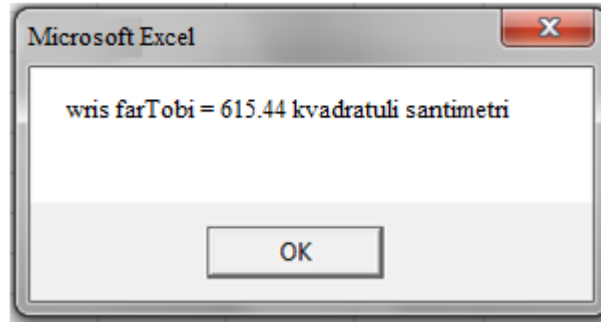
მნიშვნელობათა შეტანისა და გამოტანისათვის გამოიყენეთ **MsgBox** და **InputBox**, რადიუსის ცოდნის შემთხვევაში გამოთვალეთ წრის ფართობი.

1. გახსენით პროგრამა **Excel**, გადაარქვით **Sheet1**-ს სახელი და დაარქვით „წრე“.
2. **MsgBox** და **InputBox**-ის, გამოყენებით გამოთვალეთ წრის ფართობი, თუ ცნობილია მისი რადიუსი. ამისათვის საჭიროა:
 - **Excel**-ის სამუშაო ფურცელზე შექმენით ღილაკი.
 - ღილაკს მიაბით პროგრამა.

```
Private Sub CommandButton1_Click()
Dim S As Single, R As Single
R = InputBox("SeitaneT wris radiusi", "wris farTobi")
S = 3.14 * R ^ 2
MsgBox "wris farTobi = "&S&"kvadratuli santimetri"
End Sub
```



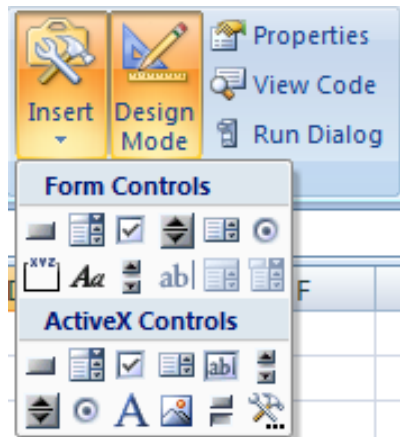
ღილაკზე “OK” გააქტიურების შემდეგ მიიღებთ შედეგს:



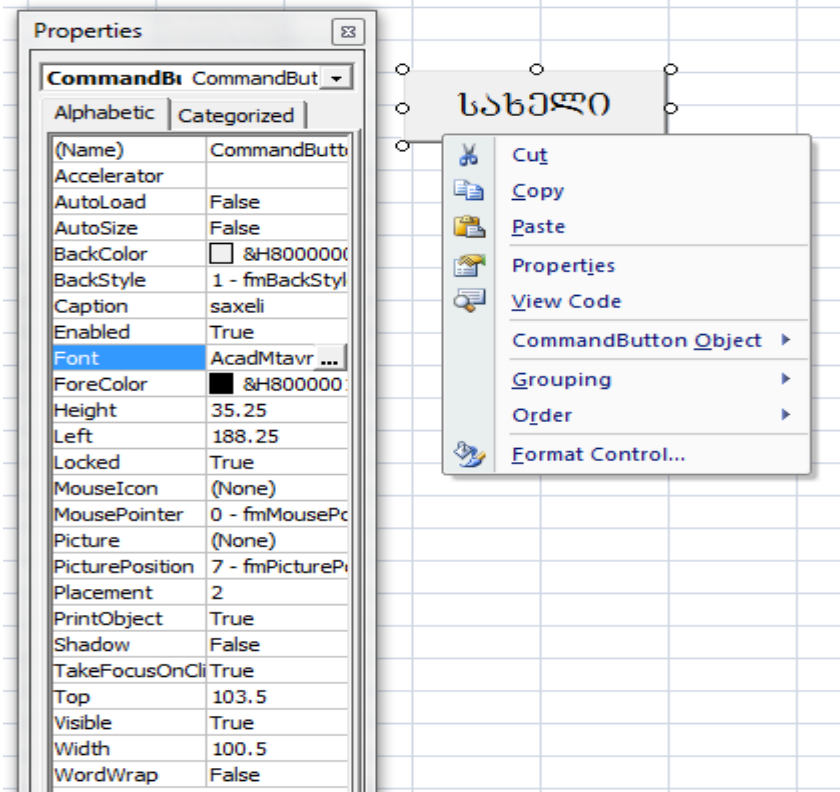
3. შეინახეთ შესრულებული სამუშაო.

დავალება: რა გქვიათ თქვენ?

1. პროგრამა **Excel** –ის სამუშაო ფურცელზე ელემენტების მართვის პანელის დახმარებით დააპროექტეთ ფორმის გამოსაძახებელი ღილაკი და დაარქვით მას „სახელი“, ასევე სამუშაო ფურცელს **Sheet1** დაარქვით სახელი „სახელი“.



2. მონიშნეთ ღილაკი, თავის მარჯვენა ღილაკის გააქტიურებით გამოიძახეთ თვისებების (Properties) ფანჯარა და დააფორმატეთ იგი.



3. მონიშნეთ ღილაკი “სახელი”, მართვის ელემენტების პანელიდან გააქტიურეთ „საწყისი ტექსტი“.

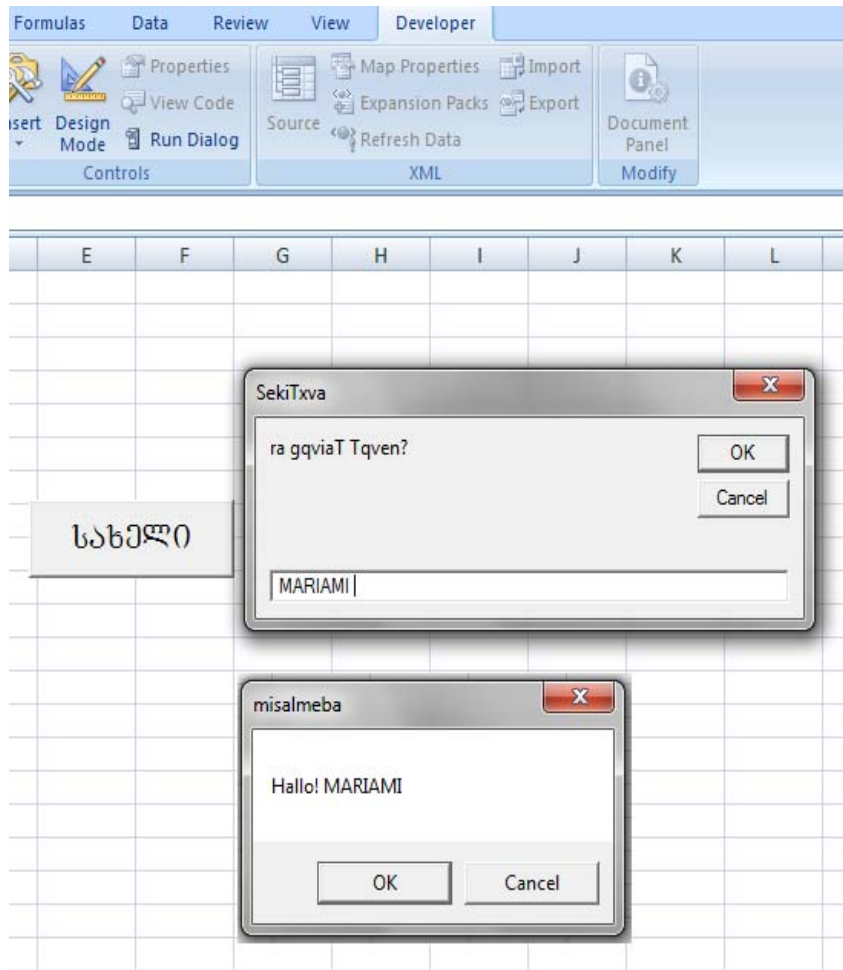
ჩაწერეთ პროგრამა:

```
Private Sub CommandButton1_Click()
Dim saxeli As String
saxeli = InputBox("რა გვიატ Tqven? ", "SekiTxva")
MsgBox "Hallo!" & saxeli, 1, "misalmeba"
End Sub
```

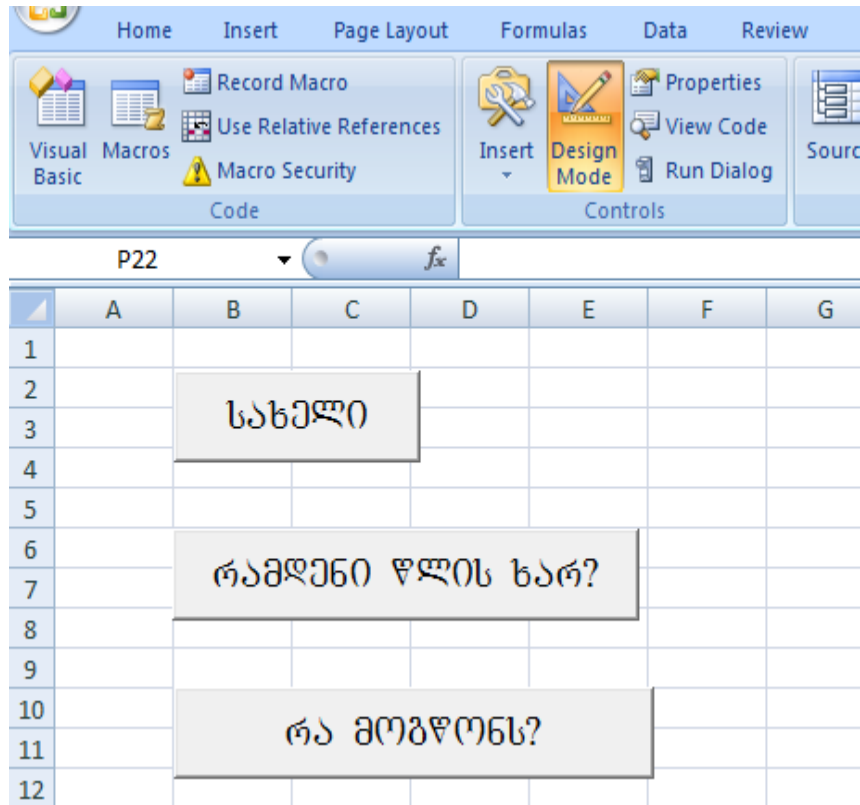
5. გამოიძახეთ სამუშაო ფურცელი „სახელი“.

6. გამორთეთ ოსტატის რეჟიმი  .

7. პროგრამის გაშვების წინ ეკრანს ექნება შემდეგი სახე:



8. დააპროექტეთ ანალოგიური დილაგები. მაგალითად, შემდეგი კითხვები: „ რამდენი წლის ხარ?“, „რა მოგწონთ?“ და ა.შ.




დავალება: ჩემი სამყარო

დააპროექტეთ კითხვარი “ჩემი სამყარო”.

მაგალითად:

- თქვენი საყვარელი საქმე ...*
- თქვენი საყვარელი ფერი...*
- თქვენი საყვარელი ცხოველი...*
- თქვენი საუკეთესო მეგობარი...*
- თქვენი საყვარელი ბგერა...*
- თქვენი საყვარელი სურნელი...*

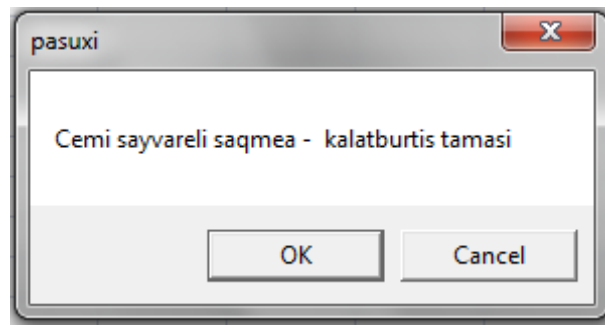
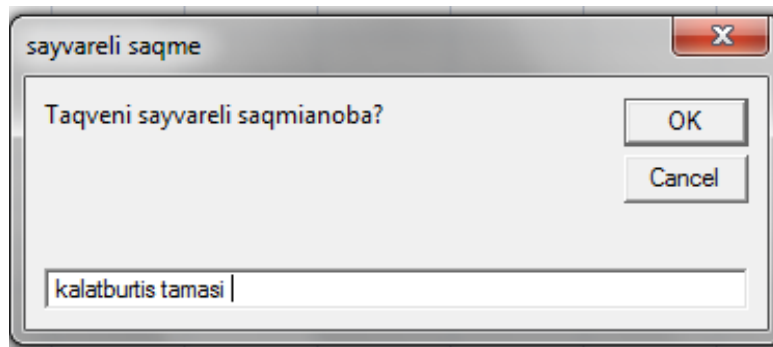
თქვენი საყვარელი თამაში..
 თქვენი საყვარელი სამოსი..
 თქვენი საყვარელი მუსიკა..
 თქვენი საყვარელი წელიწადის დრო..
 ყველაზე მეტად რის კეთებაც მიყვარს..
 ყველაზე მეტად საღაც მიყვარს ყოფნა..
 საყვარელი მომღერალი ან ჯგუფი..
 თქვენი საყვარელი გმირი..
 მე მაქვს უნარი..
 აღაძიანი, რომელსაც პატივს ვცემ..
 ყველაზე უკეთ შემიძლია..
 ვიცი, რომ შემიძლია..
 მე დარწმუნებული ვარ, რავღან..

1. პროგრამა Excel – ში შექმენით ღილაკი და დაარქვით სახელი „კითხვაზე პასუხი“.
2. გააქტიურეთ ღილაკი „კითხვაზე პასუხი“ და მართვის ელემენტების პანელზე გააქტიურეთ „საწყისი ტექსტი“ 
3. ჩაწერეთ პროგრამა ისეთნაირად, რომ პირველი შეკითხვის და პასუხის შემდეგ, გამოძახებული იქნას მეორე, მესამე და ა.შ.

```

Private Sub CommandButton1_Click()
Dim SekiTxva As String
SekiTxva = InputBox("Taqveni sayvareli saqmianoba? ",
"sayvareli saqme")
MsgBox "Cemi sayvareli saqmea — " & SekiTxva, 1, "pasuxi"
SekiTxva = InputBox("Tqveni sayvareli feri? ",
"sayvareli feri")
MsgBox "Cemi sayvareli feri — " & SekiTxva, 1, "pasuxi"
.....
End Sub
  
```

4. შეცვალეთ ფორმის დასახელება, კითხვის შესაბამისად. შედეგად მიიღებთ ქვემოთ მოცემულ გამოსახულებას:



5. შეინახეთ შესრულებული სამუშაო.

დავალება: მე მომწონს

დააპროექტეთ კითხვარი „მე მომწონს“:

მე მომწონს უნივერსიტეტში, რომ...

მე მომწონს უნივერსიტეტში ის, რომ...

მე ვამაყობ იმით, რომ...

ჩვენ არ მოგვწონს ლექციებზე ის, რომ ...

ძალიან კარგი იქნებოდა, ლექციებზე რომ შეიძლებოდა...

შეადგინეთ კითხვები და შესრულეთ დამოუკიდებელი


სამუშაო.

დავალება: მთვლელო

Excel სამუშაო ფურცელზე დააპროექტეთ ფორმის გამოსაძახებელი ღილაკი, რომლის დახმარებითაც შესაძლებელია ორ რიცხვზე არითმეტიკული მოქმედებების შესრულება.

1. **Excel** –ის სამუშაო ფურცელზე სამომხმარებლო მენიუს მართვის ელემენტების დახმარებით დააპროექტეთ ფორმის გამოსაძახებელი ღილაკი და დაარქვით „შეკრება“ ხოლო სამუშაო **Sheet 1** ფურცელს „მთვლელო“.

2. მონიშნეთ ღილაკი და მაუსის მარჯვენა ღილაკის დახმარებით თვისებების ფანჯრიდან დააფორმეთ იგი.

3. ღილაკიდან „საწყისი ტექსტი“  აწერეთ პროგრამა:

```
Private Sub CommandButton1_Click()
```

```
Dim ricxvi1 As String, ricxvi2 As String
```

```
Dim a As Integer, b As Integer
```

```
ricxvi1 = InputBox("SeitaneT pirveli ricxvi", "pirveli ricxvi")
```

```
ricxvi2 = InputBox("SeitaneT meore ricxvi ", "meore ricxvi")
```

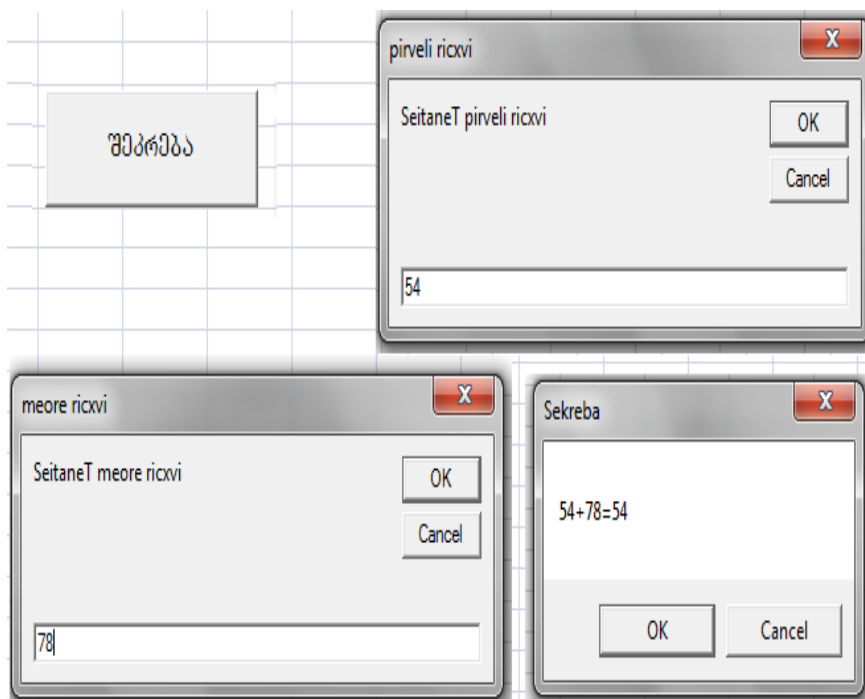
```
a = Val(ricxvi1)
```

```
b = Val(ricxv2)
```

```
MsgBox ricxvi1 & "+" & ricxvi2 & "=" & a + b, 1, "Sekreba"
```

```
End Sub
```


4. გააქტიურეთ ღილაკი „შეკრება“ და შეამოწმეთ ღილაკის ფუნქციონირება. შედეგად უნდა მიიღოთ ქვემოთ მოცემული სურათი:



- დააპროექტეთ შემდეგი დიალოგები: „გამოკლება“, „გამრავლება“, „გაყოფა“, „კვადრატული ფესვი“.
- შეინახეთ შესრულებული სამუშაო.

დავალება: ანდაზები

Excel სამუშაო ფურცელზე გამოიტანეთ ფორმის გამოსაძახებელი ღილაკი, რომლის დახმარებითაც შესაძლებელი იქნება ანდაზის „თაბჯმა თხარა თხარაო . . . ბამოთხარაო“ სრული სახით წარმოდგენა.

1. მართვის ელემენტების პანელის დახმარებით დააპროექტეთ ფორმის გამოსაძახებელი ღილაკი და დაარქვით სახელი „ანდაზები“ ასევე სამუშაო ფურცელს **Sheet1**-ს გადაარქვით სახელი და დაარქვით „ანდაზები“.
2. მონიშნეთ ფორმის გამოსაძახებელი ღილაკი (გააქტიურეთ მაუსის მარჯვენა ღილაკის დახმარებით კონტექსტური მენიუ) და დააფორმატეთ თვისებების (**Properties**) ფანჯრიდან.
3. ღილაკზე „საწყისი ტექსტი,“ მიაბით პროგრამა 

```
Private Sub CommandButton1_Click()
```

```
Dim pasuxi As String
```

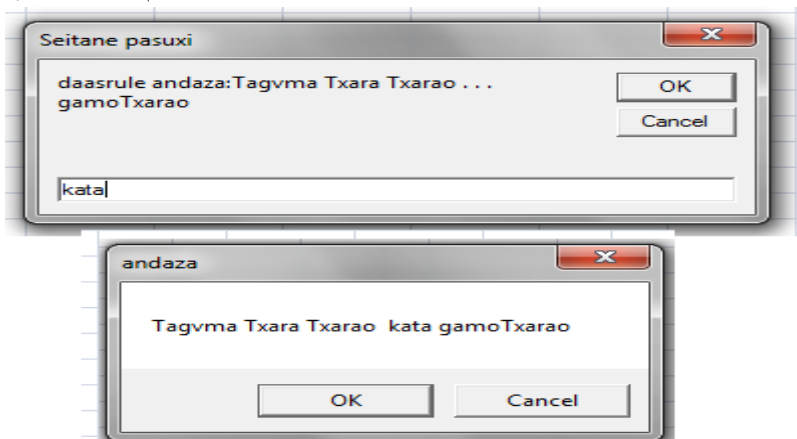
```
pasuxi = InputBox("daasrule andaza:Tagvma Txara Txarao . . .
```

```
gamoTxarao", "Seitane pasuxi ")
```

```
MsgBox " Tagvma Txara Txarao kata gamoTxarao", 1, "andaza"
```

```
End Sub
```

მიიღებთ შედეგს:



4. პროგრამა შეავსეთ რამდენიმე ანდაზით, ისე ისე, რომ პირველი შეკითხვის და პასუხის შემდეგ გამოვიდეს მეორე, მესამე და ა.შ.
5. შეინახეთ შესრულებული სამუშაო.

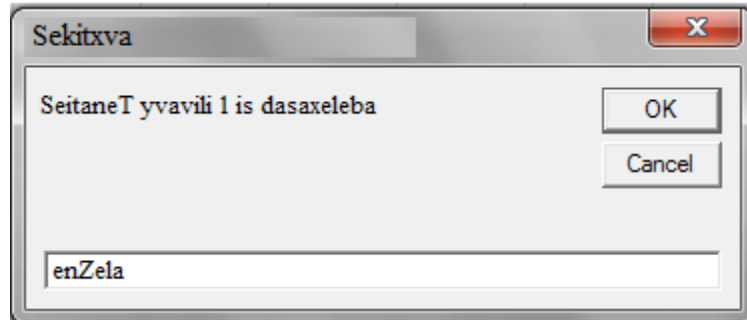
დავალება: მინდვრის ყვავილების თაიგული

ამ სამუშაოს შესრულება დაგეხმარებათ გაიგოთ, თუ როგორ ხდება ინფორმაციის გადატანა დიალოგური ფანჯრიდან სამომხმარებლო ფორმაზე.

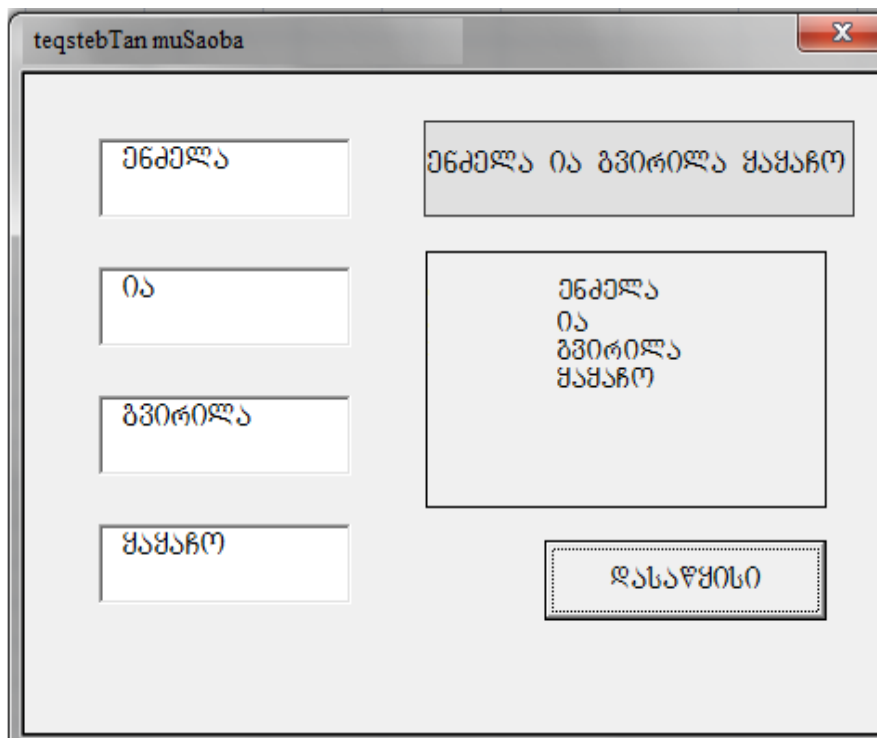
შევაგროვოთ ოთხი დასახელების მინდვრის ყვავილისაგან თაიგული.

1. ლილაკის „დასაწყისი“ გააქტიურების შემდეგ თანამიმდევრულად უნდა მივიღოთ დიალოგური ფანჯრები, ოთხი სახეობის მინდვრის ყვავილის დასახელების შესატანად (ბიჭებისათვის ავტომობილები).
2. ტექსტური სტრიქონი შეიტანეთ სამომხმარებლო ფორმის ჭდის ველში.
3. შეასრულეთ ტექსტური სტრიქონის გაერთიანება.
4. პირველ ეტაპზე მონიშნეთ გაერთიანებული ტექსტური სტრიქონი სამომხმარებლო ფორმაში და წარმოადგინეთ ერთ სტრიქონად.
5. მეორე ეტაპზე გაერთიანებული ტექსტური სტრიქონები განალაგეთ სამომხმარებლო მენიუზე „სვეტების“ სახით. ტექსტური სტრიქონის სვეტების სახით გამოსატანად საჭიროა გაერთიანებულ ტექსტურ სტრიქონებს შორის ჩაისვას უხილავი სიმბოლოები, რომელიც უზრუნველყოფს ჩანაწერის ახალ სტრიქონზე გადატანას. **ASCII** კოდებში მას შეესაბამება რიცხვი **13**. ფუნქცია **CHR**(კოდი) კოდს

გარდაქმნის ერთ სიმბოლოდ. CHR(13) არის ახალი სტრიქონის სიმბოლო.



ტექსტური სტრიქონის შესატანი ფანჯარა



პროგრამას დილაკისათვის „დასაწყისი“ აქვს შემდეგი სახე:

Private Sub CommandButton1_Click()

Rem ცვლადების გამოცხადება

Dim sityva1 As String

Dim sityva 2 As String

Dim sityva 3 As String

Dim sityva 4 As String

Dim sityva As String

Rem ტექსტური სტრიქონის შეტანა, შეტანის ფანჯრიდან

sityva 1 = InputBox("SeitaneT dasaxeleba yvavili 1 ", "SekiTxva")

Rem განათავსეთ ტექსტური სტრიქონი სამომხმარებლო

ფორმის ველში *Label1.Caption = sityva 1*

sityva 2 = InputBox("SeitaneT dasaxeleba2 yvavili 2", " SekiTxva ")

Label2.Caption = sityva 2

sityva 3 = InputBox("SeitaneT dasaxeleba yvavili 3", " SekiTxva ")

Label3.Caption = sityva 3

sityva 4 = InputBox("SeitaneT dasaxeleba yvavili 4", " SekiTxva ")

Label4.Caption = sityva 4

**Rem გააერთიანეთ ტექსტური სტრიქონები, ერთ სტრიქონად
გამოსატანად**

sityva = sityva 1 + " " + sityva 2 + " " + sityva 3 + " " + sityva 4

**Rem გაერთიანებული ტექსტური სტრიქონი განათავსეთ
სამომხმარებლო ფორმის ველში**

Label5.Caption = sityva

Rem ცვლადის sityva მოშადება ახალ სტრიქონთან

გასაერთიანებლად

sityva = ""

**Rem გაერთიანებული ტექსტური სტრიქონი უარმოადგინეთ
სვეტის სახით**

sityva = sityva 1 + " " + Chr(13) + sityva 2 + " " + Chr(13) + sityva 3

+ " " + Chr(13) + sityva 4

**Rem გაერთიანებული ტექსტური სტრიქონი განათავსეთ
სამომხმარებლო ფორმის ველში**

Label6.Caption = sityva

End Sub

6. პროგრამა **Excel** –ის სამუშაო ფურცელზემოათავსეთ ფორმის გამოსაძახებელი ღილაკი.

7. შეინახეთ თქვენს მიერ შესრულებული სამუშაო.

თემა №7 ალგორითმული კონსტრუქციები

პროცედურები და მოდულები VBA კოდისათვის.

VBA-ს კოდი განთავსებულია მოდულში. მოდული შედგება პროცედურებისაგან. მოდულში შემავალი ცალკეული პროცედურა არის ან პროცედურა-ფუნქცია ან პროცედურა ქვეპროგრამა.

მოდული არის გამოცხადებების და პროცედურების ერთობლიობა, რომელიც იწახება როგორც ერთი მთლიანი.

პროცედურა *VBA*-ს კოდის ნებისმიერი ერთობლიობაა, რომელიც განიხილება როგორც ერთი მთლიანი. როგორც წესი, პროცედურა შედგება ოპერატორებისაგან, რომელიც შეასრულებს ან ამოხსნის რაღაც ამოცანას. ცალკეულ პროცედურას აქვს თავისი უნიკალური სახელი. ხშირად პროცედურის შესრულება წარმოადგენს რეაქციას რაღაც ხდომილებაზე. ამ შემთხვევაში ამბობენ, რომ პროცედურა ამუშავებს ხდომილებას.

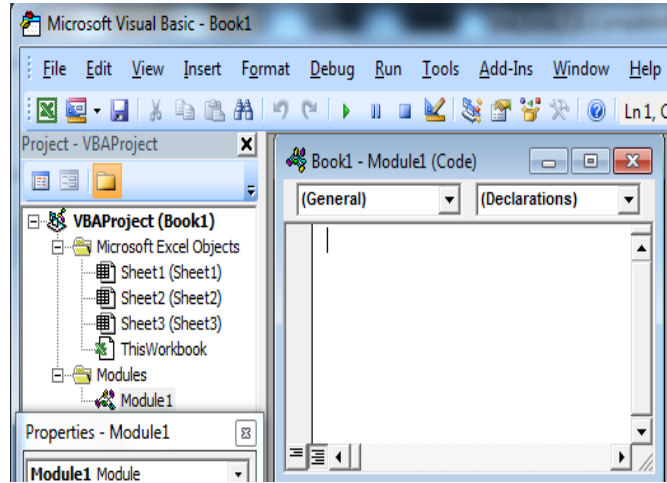
პროცედურის შექმნა

პირველი პროცედურის შექმნა მოითხოვს ორი ერთმანეთის მომდევნო ბიჯის შესრულებას:

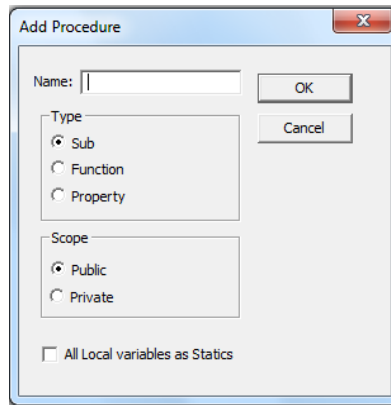
- თავდაპირველად მოდული უნდა ჩაიდგას მუშა წიგნში;
- შემდგომ ამ მოდულში შეიქმნას პროცედურა.

ცალკეული დანართისათვის უნდა შეიქმნას თავისი მოდული. დანართი შეიძლება შეიცავდეს რამოდენიმე მოდულს, თუმცა ეს არააუცილებელია.

1. გახსენით ახალი მუშა წიგნი.
2. გახსენით Visual Basic რედაქტორი.
3. აირჩიეთ ბრძანება **Insert|Module**. თქვენს დანართს დაემატება მოდული.

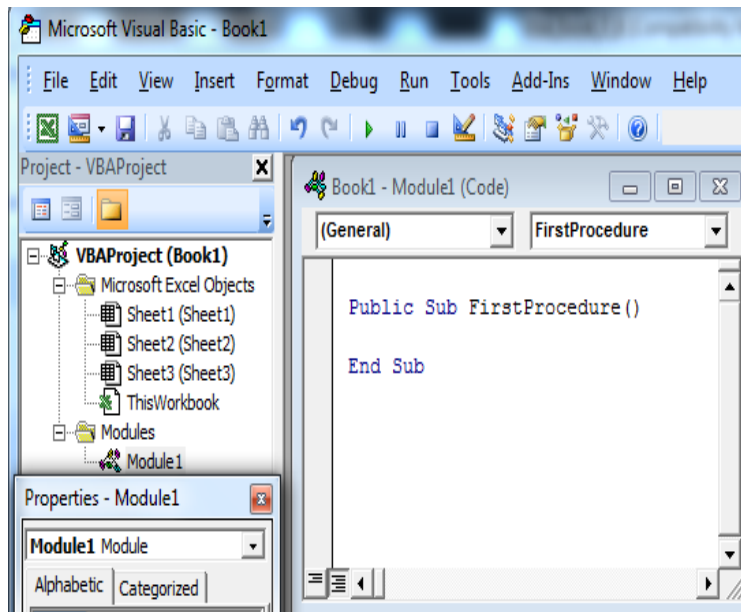


4. შესრულეთ ბრძანება **Insert | Procedure**. გაიხსნება დიალოგური ფანჯარა Add Procedure (პროცედურის დამატება).



5. შეიტანეთ პროცედურის

სახელი. Type (ტიპი) სექციაში დააყენეთ გადამროველი Sub (ქვეპროგრამა). დააწკაპუნეთ OK ღილაკზე. მოდულში დაემატება ახალი პროცედურა.



პროცედურის ტანის ცარიელ სტრიქონში განთავსებულია ტექსტური კურსორი, სადაც უნდა იქნას შეტანილი პროცედურის კოდი.

პროცედურის სახელის შექმნისას VBA მოითხოვს განსაზღვრული წესების დაცვას.

- სახელის პირველი-სამი სიმბოლო უნდა იყოს ასო.
- სახელი შეიძლება შეიცავდეს ასოს, ციფრს, ხაზგასმის ნიშანს.
- სახელი არ უნდა შეიცავდეს პრობელს, წერტილს, მძიმეს, ძახილის ნიშნებს, ასევე "@", "&", "\$", "#" სიმბოლოებს.
- სახელი არ უნდა შეიცავდეს 255 სიმბოლოზე მეტს.

პროცედურის შესრულება

პროცედურის შექმნის შემდეგ შეგიძლიათ იგი მაშინვე შეასრულოთ. ამისათვის არსებობს რამოდენიმე გზა:

- აირჩიეთ ბრძანება **Run Sub | UserForm;**
- დააჭირეთ **F5** ღილაკს;
- შექმენით Excel – ის მუშა ფურცელზე ავტოფიგურა და მიაბით მას გამზადებული პროცედურა.

ცვლადები

ცვლადი არის მნიშვნელობის შესანახი ადგილი. ცვლადი შეიცავს მონაცემებს, რომელიც შეიძლება შეიცვალოს პროგრამის შესრულების პროცესში.

მონაცემების აღწერა იწყება DIM ოპერატორით.

- ცვლადის სახელი შეიძლება იწყებოდეს ასოთი და შეიცავდეს ასოს, ციფრს და სხვა სიმბოლოს. სახელი არ უნდა არ უნდა შეიცავდეს პრობელს, წერტილს, მძიმეს, დახილის ნიშნებს, ასევე "@", "&", "\$", "#" სიმბოლოებს. სახელი არ უნდა შეიცავდეს 255 სიმბოლოზე მეტს.

Dim ოპერატორში ცვლადების ტიპი შეიძლება არ მიუთითოთ. ასეთ შემთხვევაში VBA ტიპს მიანიჭებს დუმილით (Variant). თუმცა მიზანშეწონილია ტიპი წინასწარ იქნას განსაზღვრული, პირველ რიგში მეხსიერების რესურსების ეკონომიის მიზნით. არც ერთი ტიპი არ მოითხოვს 16 ან 22 ბაიტს ცვლადის მნიშვნელობის შესანახად. გარდა ამისა VBA სხვადასხვანაირად ამუშავებს განსხვავებული ტიპის მონაცემებს. ამიტომ თუ ტიპი არ იქნება გამოცხადებული შეიძლება მიღებულ იქნას სასურველისაგან განსხვავებული შედეგი. ასევე საყურადღებოა რომ, მონაცემთა ტიპებზე დამოკიდებულია პროცედურის შესრულების დრო.

ცვლადების განსაზღვრის მაგალითები:

Dim A As Integer, B As Byte, C As String

A ცვლადი განსაზღვრულია როგორც მთელი რიცხვი (32767 -32768 დიაპაზონში); B ცვლადი განსაზღვრულია როგორც მთელი არაუარყოფითი რიცხვი (არაუმეტეს 255), ხოლო C ცვლადში შეიძლება ინახებოდეს ტექსტური ინფორმაცია.

მუდმივების გამოყენება

როგორც უკვე აღვნიშნეთ ცვლადები შეიძლება შეიცვალოს პროცედურის შესრულების პროცესში. თუ საჭიროა მუდმივი ინფორმაციის შენახვა ანუ ინფორმაციისა, რომელიც არ იცვლება პროცედურის შესრულებისას, მაშინ გამოიყენება კონსტანტა. მისი გამოცხადებისათვის გამოიყენება ოპერატორი **Const**, რომელსაც აქვს შემდეგი სინტაქსი:

Const კონსტანტის_სახელი **As** მონაცემთა_ტიპი = მნიშვნელობა

კონსტანტების გამოცხადებისას გამოიყენება იგივე მონაცემთა ტიპები, როგორც ცვლადების გამოცხადებისას.

კონსტანტის გამოცხადების მაგალითი:

Const სწავლის_წლები **As Byte** = 11

პროგრამა VBA ენაზე

პროგრამა რომ იყოს იოლი წასაკითხი იყენებენ კომენტარების ოპერატორს. VBA ენაში არსებობს კომენტარის შეტანის ორი ხერხი: აპოსტროფის გამოყენება, რომელიც შეიძლება ჩაისვას სტრიქონის ნებისმიერ ადგილას და დარეზერვებული სიტყვა **Rem**. ამ დროს დაკომენტირდება ტექსტი სტრიქონის ბლომდე.

With/End ოპერატორი

With ოპერატორი ერთიდაიგივე ობიექტის სახელის მრავალჯერ განმეორების თავიდან აცილების საშუალებას იძლევა.

სინტაქსი:

With ობიექტი

ოპერატორი 1

ოპერატორი 2

.....

ოპერატორი N

End with

მაგალითად, ოპერატორების მიმდევრობა:

UserForm1.TextBox1.Text = Date

UserForm1.TextBox2.Text = " "

UserForm1.Label1.Caption = " "

UserForm1.Label2.Caption = "დასახელება"

შეიძლება დაიწეროს ასე:

With *UserForm1*

.TextBox1.Text = Date

.TextBox2.Text = " "

.Label1.Caption = " "

.Label2.Caption = "Название"

End with

VBA-ს მმართველი სტრუქტურები

მმართველი სტრუქტურები განსაზღვრავენ პროგრამის შესრულების მიმდევრობას. მათ გარეშე პროგრამის ყველა ოპერატორი შესრულდება მარცხნიდან მარჯვნივ და ზემოდან ქვემოთ. ზოგჯერ მოითხოვება ინსტრუქციების ზოგიერთი ნაკრების მრავალჯერადი შესრულება ან

ამოცანის ამოხსნა სხვაგვარად, შესრულების დროს მომხმარებლის მიერ მიცემული პარამეტრების ან ცვლადების მნიშვნელობების მიხედვით. მსგავს შემთხვევებში დიდ როლს ასრულებენ მართვის კონსტრუქციები და ციკლები.

VBA ენაში არის შემდეგი მმართველი კონსტრუქციები:

If ... Then

If ... Then ... Else

Select Case

If... Then კონსტრუქცია

ეს კონსტრუქცია გამოიყენება, როდესაც რაღაც პირობის მიხედვით უნდა შესრულდეს ერთი ან რამოდენიმე ოპერატორი.

სინტაქსი:

I ხერხი:

If პირობა Then გამოსახულება

II ხერხი:

If პირობა Then

გამოსახულება

End If

ჩვეულებრივ, პირობა არის მარტივი შედარება, მაგრამ იგი შეიძლება იყოს ნებისმიერი გამოსახულება. თუ გამოსახულება ჭეშმარიტია, მაშინ გამოითვლება ყველა გამოსახულება, რომელიც განთავსებულია **Then** სიტყვის შემდეგ.

მაგალითად, შემდეგი ორი ოპერატორი ექვივალენტურია:

If $A \geq B$ Then $A = A * 2$

If $A \geq B$ Then

$A = A * 2$

End If

თუ პირობის შემდეგ უნდა შესრულდეს ოპერატორების მიმდევრობა, გამოყენებულ უნდა იქნას ბლოკური კონსტრუქცია: **If ... Then ... End If**.

If $A \geq B$ **Then**

$A = A * 2$

Sheets("amocana").Range("B2") = A

Rem "amocana" ფურცელზე B2 უჯრაში განთავსდეს მნიშვნელობა A

End If

თუ პირობა მცდარია, მაშინ **Then** გასაღებური სიტყვის შემდეგ განთავსებული ოპერატორები არ შესრულდება, მართვა გადაეცემა შემდგომ სტრიქონს ან ბლოკურ კონსტრუქციაში **End If** ოპერატორის შემდგომ სტრიქონს.

If ... Then ... Else და **If ... Then ... ElseIf** კონსტრუქციები

If ... Then ... Else კონსტრუქცია განსაზღვრავს ოპერატორების რამოდენიმე ბლოკს, რომელთაგან ერთი შესრულდება პირობის მიხედვით.

If პირობა **Then**

გამოსახულება 1

...

Else

გამოსახულება 2

...

End If

თავდაპირველად მოწმდება პირობა. თუ იგი ჭეშმარიტია, შესრულდება ოპერატორების შესაბამისი ბლოკი, შემდეგ მართვა გადაეცემა **End If** –ის შემდგომ ოპერატორს. წინააღმდეგ შემთხვევაში შესრულდება **Else** ოპერატორის ბლოკი.

If ... Then ... ElseIf კონსტრუქცია წინა კონსტრუქციის სპეციალური შემთხვევაა, რომელიც გამოიყენება ჩადგმული

კონსტრუქციების ორგანიზებისათვის. მოვიყვანოთ ორი რიცხვის შედარების მაგალითი:

```
Sub amocana1 ()  
Dim Rez As String  
a = (InputBox("Seitane a", "monacemTa Setana", 0))  
Dim a As Single, b As Single  
monacemTa Setana", 0))  
b = (InputBox("Seitane b", "monacemTa Setana", 0))  
If a < b Then  
Rez = "a < b"  
ElseIf a = b Then  
Rez = "a = b"  
Else Rez = "a > b"  
End If  
Msgbox Rez,64,"informacia"  
End Sub
```

If ... Then კონსტრუქციაში შეიძლება **ElseIf** ბლოკების ნებისმიერი რაოდენობის დამატება. თუმცა **ElseIf** ბლოკების დიდი რაოდენობის შემთხვევაში **If ... Then** კონსტრუქცია გახდება ძალიან დიდი და მოუხერხებელი. მსგავს შემთხვევაში მიზანშეწონილია **Select Case** კონსტრუქციის გამოყენება.

Select Case კონსტრუქცია

Select Case კონსტრუქცია წარმოადგენს **If . . . Then . . . Else** კონსტრუქციის ალტერნატივას. ამორჩევის რამოდენიმე ვარიანტის შემთხვევაში იგი პროგრამულ კოდს იოლად წასაკითხს ხდის.

Select Case კონსტრუქცია მუშაობს ერთადერთი შესამოწმებელი გამოსახულებით, რომელიც გამოითვლება ერთჯერ ამ კონსტრუქციაში შესვლისას. შემდგომ VBA ადარებს მიღებულ შედეგს **Case** ოპერატორში დასმულ მნიშვნელობას, მანამ სანამ არ დაემთხვევა.

Select Case *Sesamowmebeli_gamosaxuleba*

[**Case** *gamosaxulebis_sia 1*

[*operatorebis_bloki 1*]]

[**Case** *gamosaxulebis_sia 2*

[*operatorebis_bloki 2*]]

...

[**Case Else**

[*operatorebis_bloki n*]]

End Select

ცალკეული გამოსახულების სია შეიძლება შეიცავდეს ერთ ან მეტ მნიშვნელობას. ასეთ შემთხვევაში ისინი გამოიყოფებიან მძიმეებით. ოპერატორების ცალკეული ბლოკი შეიძლება შეიცავდეს რამოდენიმე ოპერატორს ან არცერთს. თუ აღმოჩნდება, რომ შესამოწმებელი გამოსახულება შეესაბამება **Case** ოპერატორის რამოდენიმე მნიშვნელობას, მაშინ შესრულდება პირველივე ნაპოვნი ოპერატორი. **Case Else** ოპერატორების ბლოკი შესრულდება მაშინ, თუ არ არის ნაპოვნი არცერთი შესაბამისობა.

If ... Then ... Else კონსტრუქცია შეიძლება შეიცვალოს **Select Case** კონსტრუქციით მხოლოდ მაშინ, თუ **If** ოპერატორი და ცალკეული **Elseif** ოპერატორი ითვლის ერთიდაიგივე გამოსახულებას.

ციკლის ოპერატორები

ციკლის ოპერატორები გამოიყენება განმეორებადი ფრაგმენტების დაპროგრამებისათვის ანუ ციკლური ალგორითმების რეალიზებისათვის.

არსებობს ციკლის ოპერატორის ორი სახე: ციკლის ოპერატორი განმეორებათა ფიქსირებული რიცხვით და

ციკლის ოპერატორი განმეორებათა ცვლადი რაოდენობით, რომელიც დამოკიდებულია პირობაზე.

For ციკლის ოპერატორი (განმეორებათა ფიქსირებული რიცხვი)

სინტაქსი:

For ცვლადი = M1 To M2 [Step M3]

ოპერატორები

Next

სადაც M1, M2, M3 გამოსახულებაა. ციკლის ოპერატორი იმეორებს ოპერატორების ჯგუფის შესრულებას, სანამ ცვლადი შეიცვლება საწყისი M1 მნიშვნელობიდან საბოლოო M2 მნიშვნელობამდე, მითითებული M3 ბიჯით. თუ ბიჯი არ არის მითითებული იგი ჩაითვლება 1.

მაგალითი:

Sum = 0

For I = 1 To 31

Sum = Sum + Sheets("temperatura").Cells(I,1)

Next

ST = Sum/31

მოყვანილ მაგალითში ნაჩვენებია, რომ ფურცელზე “ტემპერატურა” პირველ სვეტში ჩაწერილია ტემპერატურის მაჩვენებელი ივლისის თვეში. უნდა იქნას გამოთვლილი თვის საშუალო ტემპერატურა.

While ცილის ოპერატორი (განმეორებათა ცვლადი რიცხვით)

სინტაქსი:

Do While პირობა

ოპერატორები

Loop

ყველა ოპერატორი შესრულდება მანა, სანამ პირობა არის ჭეშმარიტი. თუ ციკლში შესვლისას პირობა მცდარია, ოპერატორები აღარ შესრულდება.

მაგალითი:

Rem მასივის ლუწინდექსიანი წევრები გააორმაგეთ და გამოიტანეთ მუშა ფურცელზე სახელით „ციკლი“

Dim A(10) As Byte, i As Byte, j As Byte

Rem მასივი უნდა შეავსოთ

i = 0

j = 0

Do While i < 10

j = j + 1

i = i + 2

A(i) = A(i) * 2

Sheets("cikli").Cells(j, 1) = A(i)

Loop

თემა №8. Excel – ის მუშა ფურცელზე ინფორმაციის დამუშავების ალგორითმები

განვიხილოთ ინფორმაციის დამუშავების ძირითადი ალგორითმები: უჯრების დიაპაზონის მნიშვნელობების ჯამის პოვნა, უჯრების დიაპაზონში ელემენტების რაოდენობის განსაზღვრა, უჯრების დიაპაზონში მაქსიმალური და მინიმალური ელემენტების განსაზღვრა.

დავალება: რიცხვების განაწილება ფურცლებზე

1. ფურცელზე 1 (რიცხვები) A1–A20 უჯრები შეივსოს შემთხვევითი რიცხვებით (-50; 50) ინტერვალიდან.

2. ფურცელზე 1 (რიცხვები) C1 უჯრაში ჩაიწეროს „რაოდენობა +“, ხოლო D1 უჯრაში განთავსდეს დათვლილი მნიშვნელობა დადებითი რიცხვების რაოდენობით.

3. ფურცელზე 1 (რიცხვები) C2 უჯრაში ჩაიწეროს „რაოდენობა -“, ხოლო უჯრაში D2 განთავსდეს უარყოფითი რიცხვების რაოდენობის დათვლილი მნიშვნელობა.

4. ფურცელზე 1 (რიცხვები) C3 უჯრაში ჩაიწეროს „რაოდენობა 0“, ხოლო უჯრაში D3 განთავსდეს ნულის ტოლი რიცხვების რაოდენობის დათვლილი მნიშვნელობა.

5. ფურცელზე 2 (დადებითები) B1 უჯრაში ჩაიწეროს „დადებითები“ და B2 უჯრიდან დაწყებული სვეტში განთავსდეს ყველა დადებითი რიცხვი.

6. ფურცელზე 3 (უარყოფიტები) C1 უჯრაში ჩაიწეროს „უარყოფითები“ და D1 უჯრიდან დაწყებული სტრიქონში განთავსდეს ყველა უარყოფითი რიცხვი.

7. შეიქმნას ღილაკი „რაოდენობა“ ფურცელზე „რიცხვები“.

8. შეიქმნას ღილაკი „გადატანა“ ფურცელზე „რიცხვები“.

9. შეიქმნას ღილაკი „გასუფთავება“ ფურცელზე „დადებითები“.

10. შეიქმნას ღილაკი „გასუფთავება“ ფურცელზე „უარყოფითები“.

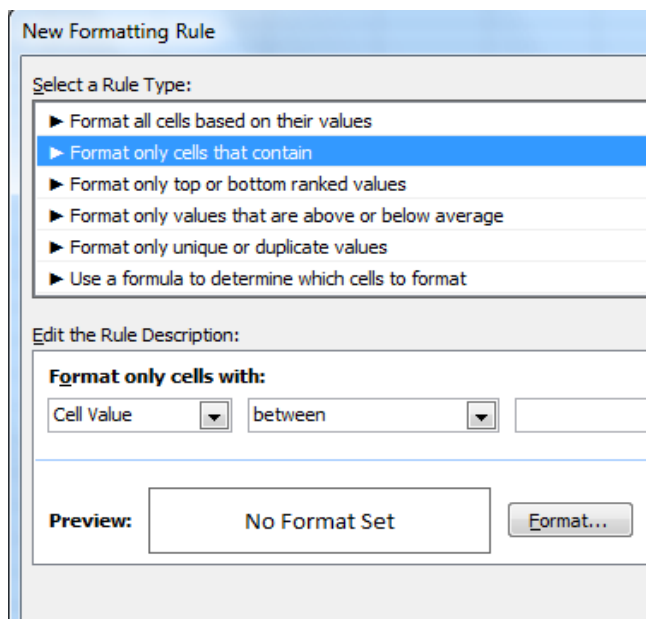
სამუშაოს გეგმა

1. ფურცელ 1 –ს დაარქვით „რიცხვები“, ფურცელ 2-ს „დადებითები“, ფურცელ 3-ს „უარყოფითები“.

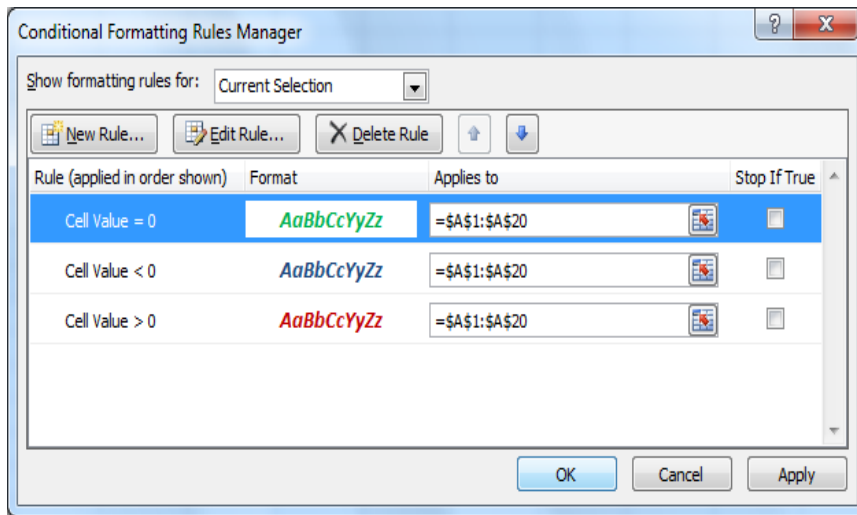
2. A სვეტისათვის (ფურცელი „რიცხვები“) გამოიყენეთ პირობითი ფორმატირება. უჯრების დიაპაზონის რიცხვებით შევსების შემდეგ დადებით რიცხვებს ექნებათ ასეთი ფორმატი: მსხვილი, დახრილი, წითელი ფერი; უარყოფითებს: მსხვილი, დახრილი, ლურჯი ფერი; ნულოვან მნიშვნელობებს: მსხვილი, დახრილი, მწვანე ფერი. ამისათვის:

– გამოყავით A სვეტი;

– შეასრულეთ ბრძანება: Home|Styles|Conditional Formatting|Manager Rules. გამოჩნდება დიალოგური ფანჯარა, რომელშიც დააჭირეთ New Rules, გამოსულ New Formatting Rules ფანჯარაში აირჩიეთ Format only cells that contain.



Format დილაკზე დაჭერის შემდეგ აირჩიეთ მნიშვნელობები ნიმუშის მიხედვით:



გადადიოთ VBA რედაქტორში.

– შექმენით მოდული ბრძანებით: **Insert | Module**.

– შექმენით მოდულში პროცედურა: **Insert | Procedure**.
დაარქვით სახელი „რიცხვები“.

– დაწერეთ პროგრამის ტექსტი ფურცელზე რიცხვების შესატანად.

Public Sub ricxvebi()

Dim I As Integer

Randomize Timer

For I = 1 To 20

Sheets("ricxvebi").Cells(I, 1) =

*Int(Rnd * 100) - 50*

Next I

– ფურცელზე „რიცხვები“ გამოიტანეთ დილაკი ან ავტოფიგურა. დაუნიშნეთ მას „რიცხვები“ პროგრამის შესრულების პროცედურა. რისთვისაც:

– მონიშნეთ ფიგურა;

– გამოიძახეთ კონტექსტური მენიუ;

- შეასრულეთ ბრძანება Assign Macro;
- გამოსულ დიალოგურ ფანჯარაში აირჩიეთ უკვე შექმნილი პროგრამა;
- შემოწმეთ პროგრამის მუშაობისუნარიანობა;
- შექმენით მაკროსი „რიცხვების გასუფთავება“, ფურცელზე „რიცხვები“ A1:D20 უჯრების დიაპაზონის გასუფთავებისათვის.
- გამოიტანეთ დილაკი ან ავტოფიგურა და მიაბით მაკროსი.
- შექმენით იმავე მოდულში კიდევ ერთი პროცედურა RaodDadUar დადებითი, უარყოფითი და ნულის ტოლი მნიშვნელობების დასათვლელად.

Public Sub RaodDadUar ()

```

Rem ცვლადების გამოცხადება
Rem Dad — ცვლადი დადებითი რიცხვების რაოდენობის დასათვლელად
Rem Uar — ცვლადი უარყოფითი რიცხვების რაოდენობის დასათვლელად
Rem Nul — ცვლადი ნულოვანი რიცხვების რაოდენობის დასათვლელად
Dim I As Integer, Dad As Integer, Uar As Integer, Nul As Integer
Rem ცვლადების განულება
Dad = 0
Uar = 0
Nul = 0
Rem ციკლის გახსნა რიცხვების შესამოწმებლად
For I = 1 To 20
If Sheets("ricxvebi").Cells(I, 1) > 0 Then
Dad = Dad + 1
ElseIf Sheets("ricxvebi").Cells(I, 1) < 0 Then
Uar = Uar + 1
Else

```

Nul = Nul + I

End If

Next I

Rem ფურცელზე „რიცხვები“ დათვლილი მნიშვნელობების გამოტანა

With Sheets(„ricxvebi“)

.Range(“C1”) = “raodenoba +”

.Range(“D1”) = Dad

.Range(“C2”) = “raodenoba -”

.Range(“D2”) = Uar

.Range(“C3”) = “raodenoba 0”

.Range(“D3”) = Nul

End With

End Sub

	A	B	C	D	E
1	12				
2	-12				
3	22				
4	-23			რიცხვები	
5	18				
6	-34				
7	36				
8	41			გასუფთავება	
9	-28				
10	11				
11	-10				
12	9				
13	-8				
14	15				
15	48				
16	-29				
17	35				
18	-36				
19	39				
20	-38				

– შექმენით ღილაკი „რაოდენობა“, ფურცელზე „რიცხვები“ და მიაბით პროგრამა **RaodDadUar**.

– შექმენით იმავე მოდულში კიდევ ერთი პროცედურა **gadatana** დადებითი და უარყოფითი რიცხვების სხვადასხვა ფურცლებზე გადასატანად.

Public Sub gadatana()

Rem ცვლადების გამოცხადება

Dim I **As Integer**, IndDad **As Integer**, IndUar **As Integer**

Rem ცვლადი-ინდექსი დადებითი ელემენტების უჯრების მისამართების ფორმირებისათვის

IndDad = 2

Rem ცვლადი-ინდექსი უარყოფითი ელემენტების უჯრების მისამართების ფორმირებისათვის

IndUar = 4

Rem B1 უჯრაში სიტყვა „დადებითები“ შეტანა

Sheets("dadebitebi").Range("B1") = "dadebitebi"

Rem B1 უჯრაში სიტყვა „უარყოფითები“ შეტანა

Sheets("uaryofitebi").Range("C1") = "uaryofitebi"

For I = 1 **To** 20

If Sheets("ricxvebi").Cells(I, 1) > 0 **Then**

Sheets("dadebitebi").Cells(IndDad, 2) =

Sheets("ricxvebi").Cells(I, 1)

IndDad = IndDad + 1

ElseIf Sheets("ricxvebi").Cells(I, 1) < 0 **Then**

Sheets("uaryofitebi").Cells(1, IndUar) =

Sheets("ricxvebi").Cells(I, 1)

IndUar = IndUar + 1

End If

Next I

End Sub

- შექმენით დილაკი „გადატანა“, ფურცელზე „რიცხვები“ და მიაბით პროგრამა gadatana.

	A	B	C	D
1	12		რაოდენობა +	11
2	-12		რაოდენობა -	9
3	22		რაოდენობა 0	0
4	-23			
5	18			
6	-34			
7	36			
8	41			
9	-28		რიცხვები	
10	11			
11	-10		გასუფთავება	
12	9			
13	-8			
14	15		რაოდენობა	
15	48			
16	-29			
17	35		გადატანა	
18	-36			
19	39			
20	-38			

– შექმენით ღილაკები „დადებითები“ და „უარყოფითები“ ფურცლებზე გადატანილი მნიშვნელობების გასასუფთავებლად.

– შეინახეთ სამუშაო.

დავალება: ჯამის გამოთვლა

მაქსიმალური და მინიმალური ელემენტების, უჯრების დიაპაზონის მნიშვნელობათა ჯამის, საშუალო მნიშვნელობის პონა შეიძლება სტანდარტული ალგორითმების გამოყენებით. Excel-ის დანართი შეიცავს მუშა ფურცლის 400-ზე მეტ ჩაშენებულ ფუნქციას. მათი გამოძახების საშუალებას იძლევა Application ობიექტი:

Application.FunctionSheets(არგუმენტები).

ამოცანა. უჯრების დიაპაზონში ჯამის პონა.

დავწეროთ პროგრამა, რომელიც გამოითვლის A1:A10 უჯრების დიაპაზონში განთავსებული რიცხვების ჯამს. შედეგი განმარტებითი წარწერით გამოვიდეს C2:D2 უჯრების დიაპაზონში.

ამოვხსნათ ამოცანა ორი ხერხით.

I ხერხი. სტანდარტული ალგორითმი.

საწყისი მონაცემების მომზადება.

1. მუშა ფურცელს დაარქვით სახელი „მაგალითი 1“.
2. A1:A10 უჯრების დიაპაზონი შეავსეთ მთელი რიცხვებით.

პროგრამის ტექსტის მომზადება.

3. შედით **VBA** რედაქტორში. ეკრანზე გამოვა პროექტის ფანჯარა, ჩასვით მოდული.
4. ჩასვით მოდულში ახალი პროცედურა სახელით **Cymmalxerxi**.
5. აკრიფეთ პროგრამის ტექსტი.

Public Sub SummaIxerxi()

Dim i As Byte, Sum As Integer

Sum = 0 ჯამის გამოთვლა

With Sheets("magaliTi 1")

For i = 1 to 10

Sum = Sum + .Cells(i,1)

Next i

'შედეგის გამოტანა

.Cells (2,3)="jami =": .Cells (2,4) = Sum

End With

End Sub

	A	B	C	D
1	10			
2	12		ჯამი =	410
3	34			
4	55			
5	14			
6	89			
7	56			
8	23			
9	45			
10	72			

6. შეინახეთ სამუშაო.

II ხერხი (ფუნქციის გამოყენება)

პროგრამის ტექსტი იქნება შემდეგი:

Public Sub Summa2xerxi()

Dim i **As** Byte, **Sum** **As** Integer

Sum=0 'ჯამის გამოთვლა

Sum = Application.Sum(Sheets("magaliTi 1").Range("A1:A10"))

Sheets("magaliTi 1").Cells(2,3)="jami ="

Sheets("magaliTi 1").Cells(2,4) = Sum

End Sub

დავალება: მუშა ფურცლის ფუნქციები

დავწეროთ პროგრამა, რომელიც იპოვის „მაგალითი 3“ ფურცელზე A1:A10 უჯრების დიაპაზონში განთავსებული რიცხვებიდან მინიმალურს, მაქსიმალურს, ასევე გამოითვლის საშუალო მნიშვნელობას და ნამრავლს. შედეგი, განმარტებითი წარწერებით გამოვიდეს C2:D2 უჯრებში. ამოცანა ამოვსნათ ორი ხერხით.

გამოვიყენოთ **Object Browser** ფანჯარა.

– შედით ობიექტების დათვალიერების ფანჯარაში.

– გახსნილ სიაში აირჩიეთ All Libraries.

– კლასების ფანჯარაში აირჩიეთ Application.

– კომპონენტების ფანჯარაში აირჩიეთ Worksheets.

ფანჯრის ქვედა ბოლოში გამოჩნდება Property Worksheets As Sheets(გზავნილი).

– დააჭირეთ გზავნილზე და კომპონენტების ფანჯარაში გამოჩნდება ყველა ფუნქციის სია.

ფუნქციების სიის გამოყენებით დავწეროთ პროგრამა.

Public Sub magaliTi3()

Dim I **As** byte, m **As** Integer, m1 **As** Integer,

SR **As** Long, pr **As** Long

Randomize Timer

For I = 1 **To** 10

Sheets("magaliTi 3").Cells(I, 1) = Int(Rnd * 10)

Next I

'მინიმუმის პოვნა

m = Application.Min(Sheets("magaliTi 3").Range("A1:A10"))

'მაქსიმუმის პოვნა

m1 = Application.Max(Sheets("magaliTi 3").Range("A1:A10"))

'ნამრავლი

pr = Application.Product(Sheets("magaliTi 3").Range("A1:A10"))

'საშუალო მნიშვნელობა

SR = Application.Average(Sheets("magaliTi 3").Range("A1:A10"))

'შედეგების გამოტანა

With Sheets("magaliTi 3")

.Cells(2, 3) = "min ="

.Cells(2, 4) = m

.Cells(3, 3) = "maqs ="

.Cells(3, 4) = m1

.Cells(4, 3) = "namravli ="

.Cells(4, 4) = pr

.Cells(5, 3) = "saSualo"

.Cells(5, 4) = SR

End With

End Sub

შედეგის ფანჯარა წარმოდგენილია სურათზე:

	A	B	C	D
1	6			
2	5		მინ =	1
3	8		მაქს =	9
4	7		ნამრავლი =	15240960
5	9		საშუალო	6
6	1			
7	6			
8	6			
9	4			
10	7			

დავალება: რიცხვების შედარება

დავწეროთ პროგრამა, რომელიც A და B სვეტში შეადარებს რიცხვებს და უდიდესს განათავსებს D სვეტში. მაგალითად A2 შეედარება B2-ს და მათ შორის უდიდესი განათავსდება D2 უჯრაში. შეექმნათ დილაკი პროგრამის გამოცხებისათვის და A, B და D სვეტების გასუფთავებისათვის.

პირობები:

– შესადარებელი რიცხვების რაოდენობა, ასევე თვით რიცხვები უნდა დაგენერირდეს შემთხვევითი სახით.

– უჯრებზე წარწერები შეტანილ უნდა იქნას პროგრამულად.

– პროგრამა ამუშავდება მუშა ფურცელზე განთავსებული დილაკიდან.

სამუშაოს გეგმა:

1. პროგრამა დილაკისათვის „გაშვება“

Public Sub udidesi()

Dim I As Integer, N1 As Integer, A As Integer, B As Integer

Randomize Timer

Rem N1 — წყვილების რაოდენობა

*N1 = 1 + Int(Rnd * 30)*

Rem უჯრების შევსება შემთხვევითი რიცხვებით (-50 +50)

დიაპაზონში.

With Sheets("udidesi")

For I = 2 To N1

*.Cells(I, 1) = Int(Rnd * 100) - 50*

*.Cells(I, 2) = Int(Rnd * 100) - 50*

Next I

Rem განმარტებითი წარწერების შეტანა და ფორმატირება

.Range("A1:D1").Font.Size = 11

.Range("A1:D1").Font.Bold = True

.Range("A1:D1").Font.Color = vbBlue

.Range("A1") = "ricxvi 1"

.Range("B1") = "ricxvi 2"

.Range("D1") = "udidesi"

.Range("A1:D1").Select

Rem სვეტის სიგანის ავტომატური შერჩევა

Selection.Columns.AutoFit

Rem უდიდესის პოვნა და შედეგის შეტანა

For *I = 2 To N1*

A = .Cells(I, 1).Value

B = .Cells(I, 2).Value

If *A > B Then*

.Cells(I, 4).Value = A

ElseIf *A = B Then*

.Cells(I, 4).Value = "tolia"

Else

.Cells(I, 4).Value = B

End If

Next *I*

End With

End Sub

შედეგის ფანჯარა წარმოდგენილია სურათზე:

	A	B	C	D	E	F
1	რიცხვი 1	რიცხვი 2		უდიდესი		
2	20	-2		20		
3	-14	23		23		
4	-1	25		25		
5	33	26		33		
6	-10	-13		-10		
7	-30	-37		-30		
8	49	17		49		
9	-18	42		42		ამოხსნა
10	-8	-27		-8		
11	-22	-8		-8		
12	-40	45		45		
13	-34	40		40		გასუფთავება
14	0	-11		0		
15	-41	6		6		
16	-10	-14		-10		
17	0	19		19		
18	-32	20		20		
19	11	-32		11		
20	5	-7		5		
21	-27	-34		-27		
22	-15	-48		-15		
23	25	-48		25		
24	27	48		48		
25	-8	7		7		

თემა 9. ამოცანების ამოხსნა ალგორითმული სტრუქტურების გამოყენებით

დავალება: მოწყობილობის ამორჩევა

ამოცანის დასმა

ცხრილში წარმოდგენილია მონაცემები, ფურცელზე გამოტანილ იქნას HP ფირმის საქონლის სია და მათი ფასები.

განვიხილოთ სია. მონაცემები დამკვეთზე ჩაწერილია მეორე სვეტში, დაწყებული მეორე სტრიქონიდან. თუ მეორე სვეტის დათვალიერებისას შეგვხვდება HP მნიშვნელობა, გამოვიტანთ ფურცელზე საქონლის დასახელებას, მის ფასს და ჯამურ ფასს.

საქონელი	მომწოდებელი	ფასი
პრინტერი	HP	1100
პლოტერი	HP	2000
ტელეფონი	SAMSUNG	40
კარტრიჯი	HP	100
პრინტერი	VS	400
მეხსიერება	HP	70
ქსელის კარტა	LG	60
კორპუსი	HP	50
ხმოვანი კარტა	HP	69

მოცემული ამოცანა აკეთებს შემდეგი ამოცანების რეალიზებას:

– ელემენტების სიის გამოტანა, რომლებიც აკმაყოფილებს განსაზღვრულ პირობებს;

– ელემენტების ჯამის გამოთვლა, რომლებიც აკმაყოფილებს განსაზღვრულ პირობებს;

– ელემენტების რაოდენობის დათვლა, რომლებიც აკმაყოფილებს განსაზღვრულ პირობებს.

ამოცანის ამოხსნისას გამოიყენება შემდეგი

კონსტრუქციები: For — Next, If — then — Else — End if, With — End With.

სამუშაოს მიმდევრობა

1. პირველ მუშა ფურცელს დაარქვით „ამოცანა“.
2. შექმენით ცხრილი „საქონელი-დამკვეთი-ფასი“
3. გადადით VBA სისტემაში.
4. შექმენით მოდული.
5. შექმენით პროცედურა HP სახელით.
6. შექმენით სიის ფორმირების მაკროსი.

Public Sub HP()

Dim I As Integer, y As Integer, S As Integer, j As Integer

'ცვლადი HP ფირმის საქონლის დასათვლელად
y = 0

'ცვლადი ჯამის დასაგროვებლად
S = 0

'ცვლადი სიაში საქონლის სტრიქონის ნომრის
ფორმირებისათვის

j = 1

'უხსნით ციკლს, სადაც I ციკლის პარამეტრი
სტრიქონის ნომერია, რომელიც იცვლება 2-დან 100-მდე
(ცხრილი შეიძლება შეიცავდეს 2-დან 100 სტრიქონამდე.)

With Sheets("amocana")

For I = 2 To 100

If .Cells(I, 2) = "HP" Then

'ფურცელზე გადაგვაქვს ცხრილის მონაცემები,
რომლებიც აკმაყოფილებენ პირობას.

j = j + 1

```

.Cells(j, 6) = .Cells(I, 1)
.Cells(j, 7) = .Cells(I, 3)
S = S + .Cells(I, 3)
y = y + 1
End If
Next I
'HP ფირმის საქონლის არსებობის შემოწმება
If y > 0 Then
.Cells(1, 6) = "saqoneli"
.Cells(1, 7) = "fasi"
.Cells(j + 1, 6) = "jami"
.Cells(j + 1, 7) = S
.Cells(j + 2, 6) = "firmis saqonlis raodenoba calobaSi"
.Cells(j + 2, 7) = y
Else
.Cells(1, 6) = "ასეთი საქონელი არა არის"
End If
End With
End Sub
7. ჩაწერეთ მაკრორეკორდერი „გასუფთავება“ სის
წასაშლელად.
Sub gasufTaveba()
'გასუფთავება
Range("E1:H21").Select
Selection.ClearContents
End Sub
8. მოამზადეთ ორი დილაკი ან გრაფიკული ობიექტი.
9. დაუნიშნეთ მათ გამზადებული პროგრამები.
10. შეამოწმეთ პროგრამის მუშაობისუნარიანობა.
11. შეინახეთ სამუშაო.

```


მიღებული შედეგი წარმდგენილია სურათზე:

	A	B	C	D	E	F
1	საქონელი	მომწოდებელი	ფასი		საქონელი	ფასი
2	პრინტერი	HP	1100		პრინტერი	1100
3	პლტერი	HP	2000		პლტერი	2000
4	ტელეფონი	SAMSUNG	40		კარტრიჯი	100
5	კარტრიჯი	HP	100		მესსიერება	70
6	პრინტერი	VS	400		კორპუსი	50
7	მესსიერება	HP	70		ხმოვანი კარტა	69
8	ქსელის კარტა	LG	60		ჯამი	3389
9	კორპუსი	HP	50		ფირმის საქონლის რაოდენობა ცალკობაში	6
10	ხმოვანი კარტა	HP	69		HP	გასუფთავება

დავალება: „თანამშრომლები“

ამოცანის დასმა

„თანამშრომლები“ ფურცლის შევსება სამომხმარებლო ფორმის „თანამშრომლის ბარათი“ დახმარებით. სამომხმარებლო ფორმაზე საპროექტებულ იქნას ველი სპეციალისტების სიით, რომელიც მდებარეობს „პროფესიები“ ფურცელზე.

მოცემული ამოცანა ახდენს სამომხმარებლო ფორმაში სიებიანი ველის შევსების და Excel-ის ფურცელზე მონაცემების ჩაწერის ალგორითმის რეალიზებას.

სიებიანი ველი (ComboBox) – არის მართვის ელემენტი, რომელიც გამოიყენება მნიშვნელობათა სიის შესანახად. მნიშვნელობათა სია წინასწარ იქმნება Excel-ის ფურცელზე

და პროგრამულად ფორმირდება სამომხმარებლო ფორმაზე გამოსაყენებლად.

ამოცანის თავისებურება იმაში მდგომარეობს, რომ Excel-ის ფურცელზე შეტანილი სპეციალისტების სიის დამატება ყოველთვის იყოს შესაძლებელი. ამიტომ რათა გავიგოთ, სპეციალისტებით შევსებული უჯრების დიაპაზონი და პროგრამულად გაგაკეთოთ სიებიანი ველის ფორმირება, საჭიროა განსაზღვრული ალგორითმის გამოყენება.

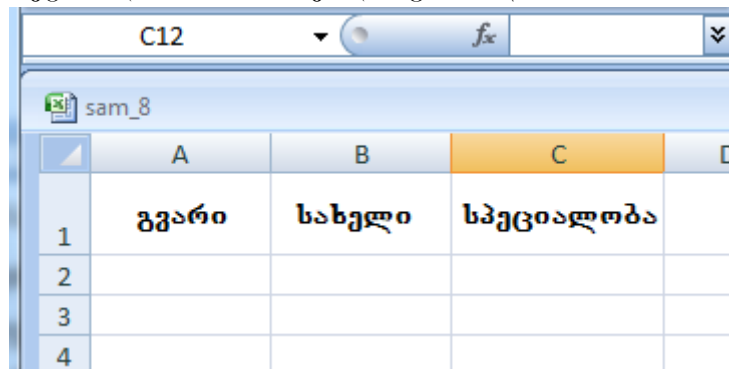
ამოცანის ამოხსნა შედგება ორი ეტაპისაგან:

– ეკრანზე ფორმის გამოძახება და სიებიანი ველის, სპეციალისტებით, ფორმირება;

– Excel-ის ფურცელზე შეტანილი ინფორმაციის ჩაწერა.

სამუშაოს მიმდევრობა

1. პირველ ფურცელს დაარქვით „თანამშრომლები“, ხოლო მეორეს - „პროფესიები“.
2. „თანამშრომლები“ ფურცელზე მოამზადეთ ცხრილის სათაური, რომელშიც ინფორმაცია უნდა იქნას შეტანილი სამომხმარებლო ფორმიდან.



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D
1	გვარი	სახელი	სპეციალობა	
2				
3				
4				

3. ფურცელზე „პროფესიები“ შეადგინეთ სპეციალობების სია.

	A	B	C
1	მასწავლებელი		
2	მენეჯერი		
3	მზარეული		
4	მღესაგი		
5	რეფერენტი		
6	ბულაღტერი		
7			
8			

ფორმა UserForm1 და დაარქვით მას ბარათი“.

4. დააპროექტეთ „თანამშრომლის“

TanamSromlis baraTi

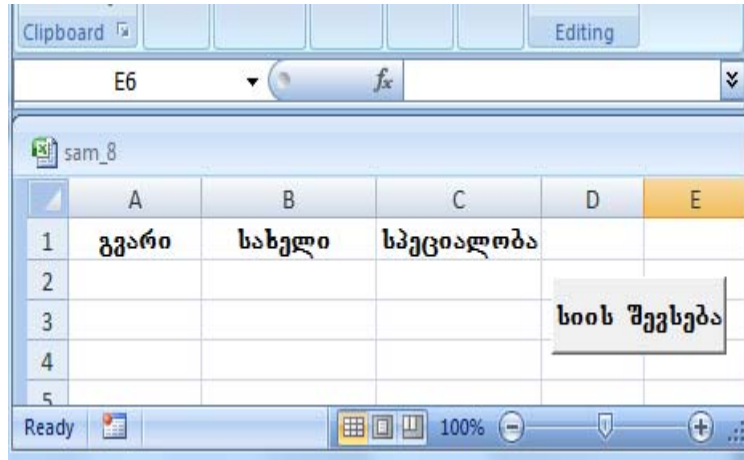
გვარი

სახელი

აირჩიე სპეციალობა

შესრულება დახურვა

5. „თანამშრომლები“ ფურცელზე მოამზადეთ ღილაკი „სის შევსება“. აღნიშნულ ღილაკზე დაჭერით გამოჩნდება მომზადებული სამომხმარებლო ფორმა. შევადგინოთ ღილაკზე დაჭერის ხდომილების პროცედურა.



სამომხმარებლო ფორმაზე სიაბიანი ველისათვის ინფორმაციის წყაროს ფორმირების ალგორითმი.

– „პროფესია“ ფურცელზე განსაზღვროთ სპეციალობებით შევსებული სტრიქონების რაოდენობა.

მითითებულ დიაპაზონში შევსებული უჯრების რაოდენობას ითვლის მუშა ფურცლის ფუნქცია **CountA**. დიაპაზონის სახით განვიხილავთ მთელ **A** სვეტს:

`Range("A:A").`

რადგანაც ეს არის ფუნქცია, აუცილებლად უნდა იქნას ნაჩვენები თუ რომელ ცვლადს მივანიჭებთ გამოთვლილ მნიშვნელობას. მოცემულ ამოცანაში ამ ცვლადს ვუწოდოთ „სია“.

`sia= Application.CountA(Sheets("Професии").Range("A:A"))`

ფუნქციის მუშაობის შედეგად ჩვენი ცვლადი „სია“ მიიღებს მნიშვნელობას – 7.

– გავაკეთოთ სპეციალობების სიის დიაპაზონის ფორმირება.

როგორც ვიცით, „პროფესია“ ფურცელზე დიაპაზონის საწყისი უჯრის მისამართი არის – A1, ბოლოს მისამართი კი მიიღება A სვეტის სახელის და იმ მნიშვნელობის შეერთებით, რომელიც იწინასება „სია“ ცვლადში. მაგრამ „სია“ ცვლადი გამოცხადებულია როგორც მთელი ტიპი, ხოლო შეერთება შეიძლება გამოყენებულ იქნას მხოლოდ სტრიქონული ცვლადებისათვის. ამიტომ „სია“ ცვლადისათვის უნდა გამოვიყენოთ ფუნქცია **CStr**: ეს ფუნქცია ცვლადის ტიპის შეცვლის საშუალებას იძლევა. ამის შემდეგ შეიძლება მოვახდინოთ შეერთება და მიღებული დიაპაზონის ახალი ცვლადისათვის მინიჭება.

$D_sia = "A1:A" \& CStr(sia)$

შედგად ახალ D_sia ცვლადს მიენიჭება A1 : A7 დიაპაზონი.

– ფორმირებულ დიაპაზონს მიანიჭეთ სახელი „სპეციალობები“. გამოვიყენოთ Range ობიექტის Name თვისება („პროფესია“ ფურცელზე).

$Sheets("profesia").Range(D_sia).Name = "specialobebi"$

შედგად „პროფესია“ ფურცელზე A1 : A7 დიაპაზონს მიენიჭება სახელი „სპეციალობები“

– სამომხმარებლო ფორმის ველს მიუერთოთ სპეციალობების სია.

ComboBox1 – არის სიებიანი ველი. **RowSource** – წყაროსტრიქონი, ობიექტის თვისებაა. ანუ სიებიანი ველის ფორმირებისათვის წყაროს წარმოადგენს უჯრების დიაპაზონი, სახელით „სპეციალობები“.

$.ComboBox1.RowSource = "specialobebi"$

სამომხმარებლო ფორმაზე სიებიანი ველისათვის ინფორმაციის წყაროს ფორმირების პროგრამას ექნება

სახე(„თანამშრომლები“ ფურცელზე განთავსებულ „სიის შევსება“ ღილაკზე ორჯერ დავაწკაპუნოთ):

Private Sub Button1_Click()

```
Dim sia As Integer
sia = Application.CountA(Sheets("profesiebi").Range("A:A"))
'განესაზღვროთ სპეციალობების სიის ღიაპაზონი
D_sia = "A1:A" & CStr(sia)
'სპეციალობების სიის ღიაპაზონს მივანიჭოთ სახელი
Sheets("profesiebi").Range(D_sia).Name = "specialobebi"
'გავასუფთაოთ უჯრები
With UserForm1
.TextBox1.Text = ""
.ComboBox1.Text = ""
'შევიტანოთ სია სიებიანი ველისათვის
.ComboBox1.RowSource = "specialobebi"
'გამოვიტანოთ სამომხმარებლო ფორმა ეკრანზე
.Show
End With
End Sub
```

6. შევქმნათ პროცედურა სამომხმარებლო ფორმიდან თანამშრომლების მონაცემების ჩასაწერად „თანამშრომლები“ ფურცელზე.

Private Sub CommandButton1_Click()

```
Dim gvari As String, specialoba As String
Dim striqoni As Integer, saxeli As String
'striqoni- ფურცელზე "თანამშრომლები" ბოლო
შევსებული სტრიქონის ნომერია.
'განესაზღვროთ ბოლო შევსებული სტრიქონის ნომერი
striqoni = Application.CountA(Sheets("TanamSromlebi").Range("A:A"))
With UserForm1
```

*'„გვარი“ და „სახელი“ ცვლადებს ვანიჭებთ TextBox1 და
TextBox2 ველების შემადგენლობას*

gvari = .TextBox1.Text

saxeli = .TextBox2.Text

*' „სპეციალობა“ ცვლადს ვანიჭებთ ComboBox1 ველის
შემადგენლობას*

specialoba = .ComboBox1.Text

End With

With Sheets("TanamSromlebi")

.Cells(striqoni + 1, 1) = gvari

.Cells(striqoni + 1, 2) = saxeli

.Cells(striqoni + 1, 3) = specialoba

End With

*' გავასუფთაოთ ComboBox1.Text, TextBox1.Tex და
TextBox2.Text შემადგენლობა*

UserForm1.ComboBox1.Text = ""

UserForm1.TextBox1.Text = ""

UserForm1.TextBox2.Text = ""

End Sub

7. დაგწეროთ ფორმის დახურვის პროცედურა.

Private Sub CommandButton2_Click()

UserForm1.Hide

End Sub

შეამოწმეთ სამუშაო და შეინახეთ.

შედგების ფანჯარა წარმოდგენილია სურათზე:

	A	B	C	D	E	F	G	H
1	გვარი	სახელი	სპეციალობა					
2	ტაბატაძე	ლევანი	მენეჯერი					
3	კეკელია	ვანო	მზარეული	სის შევსება				
4	ნოზაძე	ნათია	ბუღალტერი					
5	ლომჯარია	სანდრო	მენეჯერი					
6	ვანაძე	ლაშა	ბუღალტერი					
7	ბირკაძე	ნიკა	რეფერენტი					
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								

TanamSromlis baraTi

გვარი

სახელი

აირჩიე სპეციალობა

შესრულება დახურვა

თემა №10. მატრიცებთან მუშაობა

დავალება: მოქმედებები მატრიცებზე

ამოცანის დასმა

Excel-ის ფურცელზე მოვამზადოთ მატრიცა, რომლის ზომა ფორმირდება სტრიქონებისა და სვეტების რაოდენობაზე მოთხოვნის დახმარებით. ამავედროულად სტრიქონებისა და სვეტების რაოდენობა არ უნდა აღემატებოდეს ათს. მატრიცის ელემენტები აირჩევა შემთხვევითი რიცხვების გენერაციით (0, 9) დიაპაზონში. დააპროექტოთ სამომხმარებლო ფორმა „ოპერაციები მატრიცაზე“. ფორმაზე განთავსდეს ღილაკები „მატრიცის შეტანა“, „გამოთვლა“, „ფურცლის გასუფთავება“, „გამოსვლა“.

„მატრიცის შეტანა“ ღილაკზე დაჭერისას Excel-ის ფურცელზე ფორმირდება მატრიცა. „გამოთვლა“ ღილაკზე დაჭერისას უნდა იქნას გამოთვლილი მატრიცის ელემენტების ჯამი, მოხდეს უდიდესი ელემენტის მოძიება ან ლუწი ელემენტების შეცვლა კენტებით. აქედან თუ რომელი ოპერაცია უნდა შესრულდეს საჭიროა მითითებულ იქნას გადამრთველი ელემენტი (OptionButton).

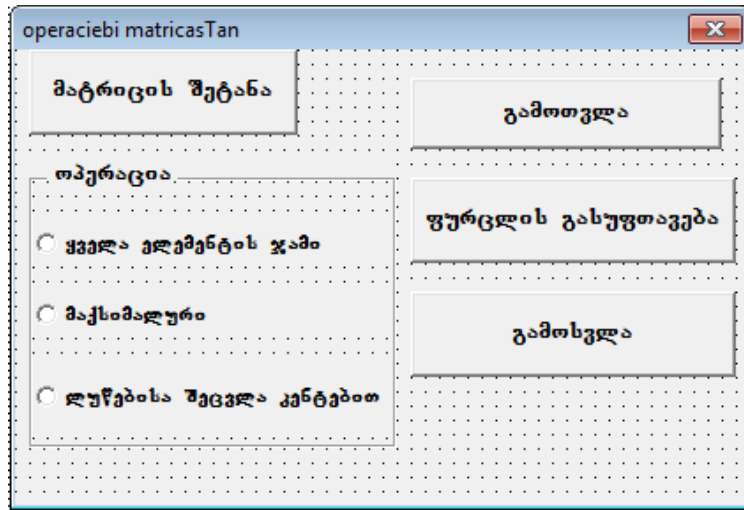
„ფურცლის გასუფთავება“ ღილაკზე დაჭერით „მატრიცა“ ფურცლიდან უნდა წაიშალოს თვით მატრიცა და ყველა შედეგი.

„გამოსვლა“ ღილაკი დახურავს ფორმას.

თვით სამომხმარებლო ფორმას გამოიძახებს Excel-ის ფურცელზე განთავსებული ღილაკი „მატრიცასთან მუშაობა“.

სამუშაოს მიმდევრობა

1. დააპროექტოთ სამომხმარებლო ფორმა UserForm1 - „ოპერაციები მატრიცასთან“.



2. დაგწეროთ პროგრამები ღილაკებისათვის.

ღილაკი „ფურცლის გასუფთავება“

```
Private Sub CommandButton3_Click()
For i = 1 To 30
For j = 1 To 30
Worksheets("matrica").Cells(i, j).Value = ""
Next j
Next i
End Sub
```

ღილაკი „გამოსვლა“

```
Private Sub CommandButton4_Click()
Userform1.Hide
End Sub
```

ღილაკი „მატრიცის შეტანა“

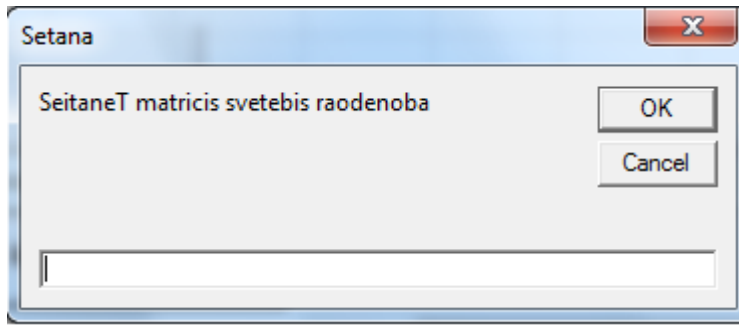
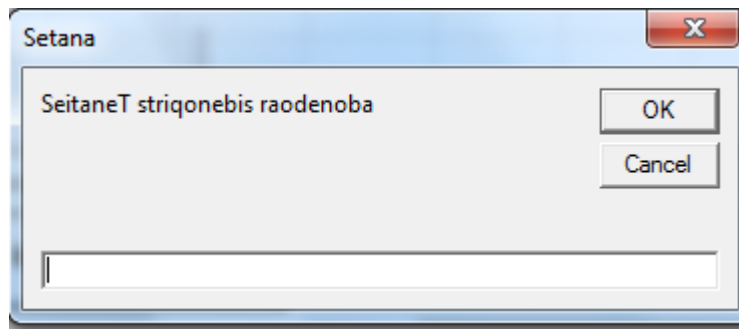
```
Dim i As Integer
Dim j As Integer
```

```

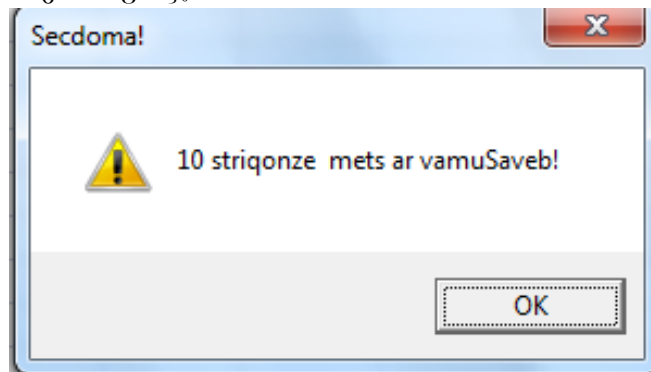
Dim m As Integer
Dim n As Integer
Dim matr() As Integer
Private Sub CommandButton1_Click()
n = InputBox("SeitaneT matricis strigonebis raodenoba", "Setana")
If m > 10 Then
MsgBox "10 svetze mets ar vamuSaveb!", 48, "Secdoma!"
GoTo metka
End If
m = InputBox("SeitaneT svetebis raodenoba", "Setana")
If n > 10 Then
MsgBox " 10 strigonze mets ar vamuSaveb!", 48, "Secdoma!"
GoTo metka
End If
Worksheets("matrica").Cells(1, 1).Value = "matrica"
For i = 1 To n
For j = 1 To m
Worksheets("matrica").Cells(i + 1, j).Value = Int(10 * Rnd)
Next j
Next i
metka:
End Sub

```

„მატრიცის შეტანა“ დილაკზე დაჭერისას გამოვა შეტყობინების ფანჯარა სტრიქონების და სვეტების რაოდენობის მოთხოვნით:



თუ შეტანილი რიცხვი 10-ზე მეტი იქნება, გამოვა შეტყობინების ფანჯარა:



დილაკი „გამოთვლა“

‘ მატრიცაზე მოქმედებების შესრულება

Private Sub CommandButton2_Click()

Dim MAX As Integer

Dim SUM As Integer

Dim flag As Integer

' ჯამის განსაზღვრა

SUM = 0

flag = 0

If OptionButton1.Value = True Then

For i = 2 To n + 1

For j = 1 To m

SUM = SUM + Worksheets("matrica").Cells(i, j)

Next j

Next i

Worksheets("matrica").Cells(11, 1).Value = "elementebis jami ="

Worksheets("matrica").Cells(12, 1).Value = SUM

MsgBox "=" & SUM, 48, "elementebis jami"

GoTo metka

End If

' მაქსიმალურის განსაზღვრა

If OptionButton2.Value = True Then

max = Worksheets("matrica").Cells(2, 1).Value

For i = 2 To n + 1

For j = 1 To m

If max < Worksheets("matrica").Cells(i, j) Then

max = Worksheets("matrica").Cells(i, j)

End If

Next j

Next i

Worksheets("matrica").Cells(11, 1).Value = "maqsimaluri elementi"

```

Worksheets("matrica").Cells(12, 1).Value = max
MsgBox "=" & max, 48, "maqsimaluri elementi"
GoTo metka
End If
'ლუწების შეცვლა
If OptionButton3.Value = True Then
For i = 2 To n + 1
For j = 1 To m
d = Worksheets("matrica").Cells(i, j)
If d / 2 = Int(d / 2) Then
Worksheets("matrica").Cells(i, j) = d + 1
flag = 1
End If
Next j
Next i
If flag = 1 Then
MsgBox "Cecvla Sesrulda", 64, "Secvla"
Else
MsgBox "Secvla ar Sesrulda, yvela elementi kentia", 64, "Secvla"
End If
End If
metka:
End Sub

```

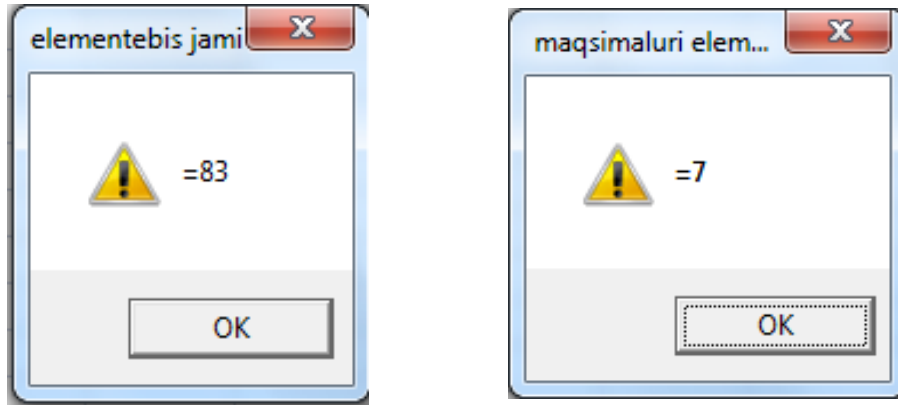
3. „მატრიცა“ ფურცელზე განათავსეთ ღილაკი „მატრიცასთან მუშაობა“, რომელიც გამოიძახებს სახმომხმარებლო ფორმას „ოპერაციები მატრიცაზე“. შეიტანეთ აღნიშნული ღილაკის პროგრამა:

Private Sub Button1_Click()

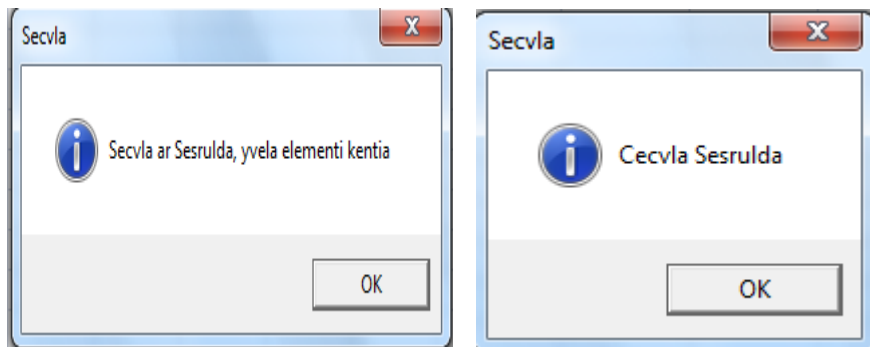
Userform1.Show

End Sub

მატრიცის ელემენტების გამოთვლილი ჯამი ან ნაპოვნი მაქსიმალური მნიშვნელობა გამოვა A12 უჯრაში, დამატებით გამოჩნდება შეტყობინების ფანჯარა:



ელემენტების შეცვლის ბრძანების არჩევისას შეტყობინების ფანჯარა შეიძლება იყოს ასეთი:



თემა №11. მატრიცებთან მუშაობა-2

აღნიშნულ სამუშაოში მოცემულია მატრიცებთან მუშაობის დროს მუშა ფურცლის ფუნქციების გამოყენება, რაც პროგრამას უფრო კომპაქტურს ხდის.

ამოცანის დასმა

ამოცანის დასმა წინა განხილულის მსგავსია, მაგრამ ოპერაციების რაოდენობას გავაფართოვებთ და არ გამოვიყენებთ ფორმას ოპერაციის არჩევისათვის. მატრიცაში ვიპოვოთ:

- ელემენტების ჯამი;
- მინიმალური ელემენტი;
- მაქსიმალური ელემენტი;
- ელემენტების წარმოებულნი;
- ელემენტების საშუალო მნიშვნელობა.

ამოცანის სირთულე მდგომარეობს იმაში, რომ ჩვენ არ ვიცით წინასწარ მატრიცის განზომილება და პროგრამულად უნდა განვსაზღვროთ მისი ფურცელზე განთავსების დიაპაზონი ფუნქციებში მათი შემდგომი გამოყენებისათვის.

სამუშაოს მიმდევრობა

ვიმუშაოთ Excel-ის იმავე ფაილში, როგორც წინა დავალების დროს.

- გადადით VBA რედაქტორში.
- შექმენით მოდული.
- შექმენით პროცედურა **vers**.
- დაწერეთ პროგრამა

Public Sub vers()

' ცვლადების გამოცხადება

Dim KStr As Integer

Dim KSvet As Integer

Dim Svet As Integer


```

Dim ad As String
Dim diap As String
Dim SUM As Integer
Dim m As Integer
Dim m1 As Integer
Dim namr As Long
Dim str As Integer
Dim sash As Single
' მატრიცაში სტრიქონების და სვეტების რაოდენობის
განსაზღვრა
KStr = Application.CountA(Sheets("matrica").Range("B:B"))
KSvet = Application.CountA(Sheets("matrica").Range("2:2"))
' მატრიცის მარჯვენა ქვედა ელემენტის მისამართის
განსაზღვრა
ad = (Sheets("matrica").Cells(KStr + 1, KSvet).Address())
' მატრიცის განთავსების დიაპაზონის ფორმირება
diap = "A2:" & ad
' ჯამის გამოთვლა
SUM = Application.SUM(Sheets("matrica").Range(diap))
' მინიმუმის პოვნა
m = Application.Min(Sheets("matrica").Range(diap))
' მაქსიმუმის პოვნა
m1 = Application.max(Sheets("matrica").Range(diap))
' ნამრავლი
carm = Application.Product(Sheets("matrica").Range(diap))
' საშუალო მნიშვნელობა
sash = Application.Average(Sheets("matrica").Range(diap))
' სტრიქონი შედეგების გამოსატანად
str = 1

```

```

' სვეტი შედეგების გამოსატანად
Svet = KSvet + 2
' შედეგების გამოტანა
Sheets("matrica").Cells(str, Svet) = "jami="
Sheets("matrica").Cells(str, Svet + 1) = SUM
str = str + 1
Sheets("matrica").Cells(str, Svet) = "minimumi ="
Sheets("matrica").Cells(str, Svet + 1) = m
str = str + 1
Sheets("matrica").Cells(str, Svet) = "maqsimumi ="
Sheets("matrica").Cells(str, Svet + 1) = m1
str = str + 1
Sheets("matrica").Cells(str, Svet) = "namravli ="
Sheets("matrica").Cells(str, Svet + 1) = namr
str = str + 1
Sheets("matrica").Cells(str, Svet) = "saSualo mniSvneloba ="
Sheets("matrica").Cells(str, Svet + 1) = sash
' სვეტის გამოყოფა კომენტარებით
Sheets("matrica").Columns(Svet).Select
' სვეტის სივანის ავტომატური შერჩევა
Selection.Columns.AutoFit
End Sub

```

- შექმენით Excel –ის ფურცელზე ღილაკი პროგრამის გასაშვებად.
 - შემოწმეთ პროგრამის მუშაობისუნარიანობა.
 - შეინახეთ სამუშაო.
- ჩვენი პროგრამის მუშაობის სისწორის შესამოწმებლად მატრიცის ელემენტების ჯამი შეიძლება გამოვითვალოთ ორივე ხერხით.
- შედეგის ფანჯარა წარმოდგენილია სურათზე:

	A	B	C	D	E	F	G	H
1	მატრიცა						ჯამი =	102
2		7	5	5	2	3	მინიმუმი =	0
3		7	0	7	8	7	მაქსიმუმი =	9
4		0	4	8	7	3	ნამრავლი =	0
5		9	8	0	9	3	საშუალო მნიშვნელობა =	5.1
6								
7								
8								
9								
10								
11	ელემენტების ჯამი =							
12		102						
13								

განვიხილოთ პროგრამის ზოგიერთი ფრაგმენტი, რომელიც განხილულ მასალაში სიახლეს წარმოადგენს.

მატრიცის ფურცელზე განთავსების დიაპაზონის განსაზღვრის ალგორითმი

- განსაზღვროთ ფურცელზე შევსებული სტრიქონებისა და სვეტების რაოდენობა. ამისათვის ვიყენებთ ფუნქციას **CountA**.

```
KStr = Application.CountA(Sheets("matrica").Range("B:B"))
```

```
KSvet = Application.CountA(Sheets("matrica").Range("2:2"))
```

პირველი სტრიქონი განსაზღვრავს შევსებული სტრიქონების რაოდენობას. დიაპაზონის რანგში ჩვენ ვიყენებთ B სვეტს, რადგან A სვეტში გამოდის ინფორმაცია კომენტარით. მეორე სტრიქონი განსაზღვრავს შევსებული სვეტების რაოდენობას. შედეგად ცვლადები მიიღებენ მნიშვნელობებს: Kstr=4; Ksvet=5.

- განვსაზღვროთ მატრიცის ქვედა მარჯვენა ელემენტის მისამართი და მივანიჭოთ მისი მნიშვნელობა ad სტრიქონულ ცვლადს. ამისათვის გამოვიყენოთ Cells ობიექტის Address მეთოდი.

Cells ობიექტის არგუმენტად ვიღებთ Kstr და Ksvet ცვლადებს. Kstr მნიშვნელობას ვზრდით ერთით, ვინაიდან პირველი სტრიქონი შეესებულ სტრიქონებში არ შედის.

ad = (Sheets("Mampuyá").Cells(KStrok + 1, KStolb).Address())

მუშაობის შედეგი: ad= \$D\$5

- განვსაზღვრავეთ მატრიცის განთავსების დიაპაზონს, რომელიც გამოიყენება ყველა ფუნქციაში. ვიყენებთ გაერთიანების უკვე ცნობილ ხერხს:

diap = "A2:"&ad

მუშაობის შედეგი: "A2 : \$E\$5"

ამის შემდეგ პროგრამა შეასრულებს გამოთვლებს და გამოიტანს შედეგებს.

თემა №12 VBA გამოყენება Word დანართისათვის

Word სისტემა **VBA**-სთან ერთად გვთავაზობს ტექსტებთან და ტექსტის ფრაგმენტებთან მუშაობის ფართო შესაძლებლობებს.

დოკუმენტის ამა თუ იმ ფრაგმენტზე წვდომისათვის გამოიყენება ისეთი ობიექტები, როგორცაა **Sections** (განყოფილებები), **Paragraphs** (აბზაცები), **Sentences** (წინადადებები), **Words** (სიტყვები) და ა.შ.

მაგალითად, **Words(6)** დოკუმენტში მეექვსე სიტყვაზე გადასვლის საშუალებას იძლევა. **Sentences(11)** - მე-11 წინადადებაზე გადასვლა. დოკუმენტის განსაზღვრულ ნაწილზე გადასვლის შემდეგ შეიძლება **Font** ობიექტის გამოყენება, რომელიც შრიფტის თვისების შეცვლის საშუალებას იძლევა. ეს ობიექტი შეიცავს მეთოდებს, რომელიც მდებარეობს **Home| Font** –ში.

Range ობიექტი არის საბაზო რედაქტირების ოპერაციაში. იგი შეიძლება განვიხილოთ როგორც ტექსტის უწყვეტი ფრაგმენტი, რომელიც არ არის დამოკიდებული კურსორის მდებარეობაზე. **Range** მოიცავს ისეთ ობიექტებს, როგორცაა **Words** (სიტყვები) ან **Sentences** (წინადადებები), მათზე მიმართვისას თვით **Range** ობიექტის მითითება არ არის საჭირო, მაგრამ **Sections** (განყოფილებები) და **Paragraphs** (აბზაცები) ობიექტებისათვის ეს აუცილებელია. **Range** ობიექტისათვის გათვალისწინებულია ძირითადი მეთოდები, რომლითაც სარგებლობენ ტექსტების ფორმატირებისას.

ამოცანების შესრულებისას საჭიროა სიმბოლურ ცვლადებთან მომუშავე ფუნქციების გამოყენება:

1	ASC(XS)	სიმბოლოს ორობითი კოდი გადაჰყავს ათობით რიცხვში.
2	LEN(XS)	განსაზღვრავს სიმბოლური ცვლადის სიგრძეს. ცარიელი ადგილების ჩათვლით.
3	MIDS(XS, Y, Z)	იძლევა Z სიმბოლოების მიმდევრობას XS სტრიქონში Y პოზიციიდან დაწყებული.
4	STRS (X)	X მნიშვნელობა რიცხვითი ფორმიდან გადაჰყავს სიმბოლურში.
5	CHRS(X)	გარდაქმნის კოდს სიმბოლოდ
6	LEFTS(AS, N)	გამოჰყოფს მარცხენა N სიმბოლოს AS სტრიქონიდან
7	RIGHTS(XS, N)	გამოჰყოფს მარჯვენა N სიმბოლოს AS სტრიქონიდან

განვიხილოთ მაგალითები:

Primer.Sentences(2).Font.ColorIndex=wdGreen - Primer დოკუმენტში, მეორე წინადადებაში შეირჩევა შრიფტის მწვანე ფერი.

Primer.Sentences(2).Font.Size=16 - Primer დოკუმენტში, მეორე წინადადებაში შეირჩევა შრიფტის ზომა 16.

Primer.Sentences(2).Font.Underline=wdUnderlineDouble - Primer დოკუმენტში, მეორე წინადადებაში შეირჩევა შრიფტი ორმაგი გახაზვით.

Primer.Sentences(2).Font.Name="Arial" - Primer დოკუმენტში, მეორე წინადადებაში შეირჩევა შრიფტი "Arial".

Primer.Range(1,15).Bold=True - სიმბოლოები 1-დან 15-ის ჩათვლით გამოიყოფა მსხვილი შრიფტით.

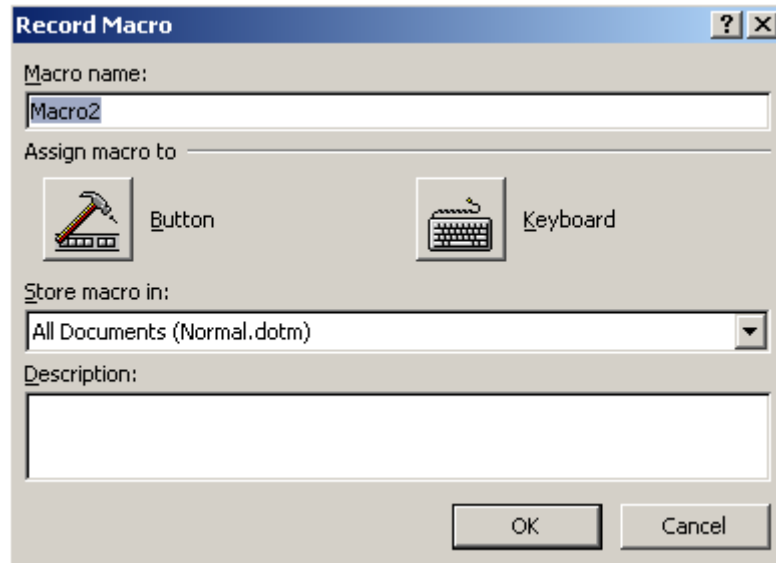
Primer.Paragraphs(5).Range.InsertAfter"Hello" - Primer დოკუმენტში, მეხუთე აბზაცის შემდეგ ჩაისმება ტექსტი "Hello".

Primer.Sentences(5).InsertAfter"Hello" - ტექსტი "Hello" ჩაისმება მეხუთე წინადადების შემდეგ.

Primer.Words(5). InsertAfter"Hello" - ტექსტი "Hello" ჩაისმება მეხუთე სიტყვის შემდეგ.

პროგრამული კოდის გაცნობისათვის შევქმნათ მაკროსი, ხოლო შემდეგ გავაანალიზოთ და შევცვალოთ მაკროსის კოდი.

1. ავირჩიოთ Developer\Record Macro;

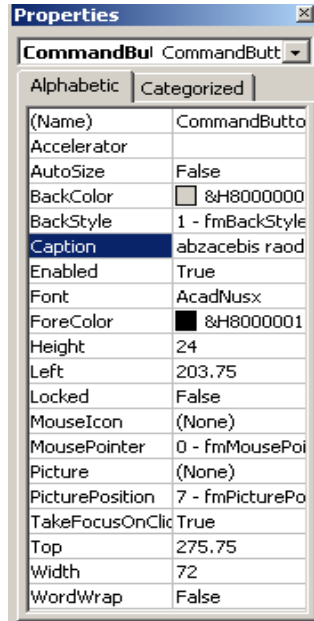


2. Macro name ველში მაკროსს დავარქვათ სახელი.
 3. Store macro in ველში ავირჩიოთ შაბლონი ან დოკუმენტი, რომელშიც შეინახება მაკროსი.
 4. Description ველში შევიტანოთ მაკროსის აღწერა.
 5. Keyboard ბრძანებიდან შეიძლება მაკროსს დაეუნიშნოს ცხელი ღილაკი.
 6. შევასრულოთ მოქმედებები, რომელიც გვინდა რომ შეასრულოს მაკროსმა.
 7. დავასრულოთ მაკროსის ჩაწერა.
- დოკუმენტის შენახვისათვის მენიუ Developer-ში ავირჩიოთ Macro Security, სადაც მოვნიშნოთ Enable all macros, შემდეგ ჩვეულებრივ შევინახოთ დოკუმენტი.

დავალება: დოკუმენტში აბზაცების რაოდენობის განსაზღვრა

1. ინსტრუმენტების ღილაკიდან Word-ის ფურცელზე განათავსეთ ღილაკი CommandButton1.

2. ღილაკს დაარქვით “აბზაცების რაოდენობა”, რისთვისაც გამოიყენეთ Properties ბრძანება, სადაც აირჩიეთ Caption თვისება.



აბზაცების რაოდენობა

3. ღილაკზე ორჯერ დააწკაპუნეთ, გამოჩნდება პროგრამული კოდის ფანჯარა.

სამუშაოს შესრულების ალგორითმი:

1. გამოვაცხადოთ ცვლადები:

k ცვლადი გამოიყენება დოკუმენტში აბზაცების რაოდენობის დასათვლელად;

Text – აბზაცების რაოდენობის შესახებ პასუხის ტექსტის ფორმირებისათვის;

Satauri – განსაზღვრავს არეს სათაურის გამოსატანად;

2. განესაზღვროთ აბზაცების რაოდენობა Paragraphs ობიექტის Count მეთოდით;
3. მოვამზადოთ შეტყობინების გამოტანის ტექსტი;
4. დოკუმენტის ბოლო აბზაცის შემდეგ ჩავსვათ დამატებითი აბზაცი შეტყობინების გამოსატანად;
5. გამოვიტანოთ ტექსტი-შეტყობინება მომზადებულ აბზაცში;
6. მოვამზადოთ არე სათაურის “პროტოკოლი” გამოსატანად;
7. გამოვიტანოთ დასახელება “პროტოკოლი”;
8. დავაყენოთ სათაურის შრიფტი და ზომები;
9. ჩავსვათ სიმბოლო აბზაცის ბოლოში იმისათვის, რომ დასახელება და ტექსტის დასაწყისი განთავსებული იყოს სხვადასხვა სტრიქონზე.
შევასრულოთ ამოცანა. დაგუშვათ, რომ შეტყობინება და აბზაცების რაოდენობა გამოვიდეს წითელი ფერის, 14 ზომის შრიფტით.

პროგრამული კოდი:

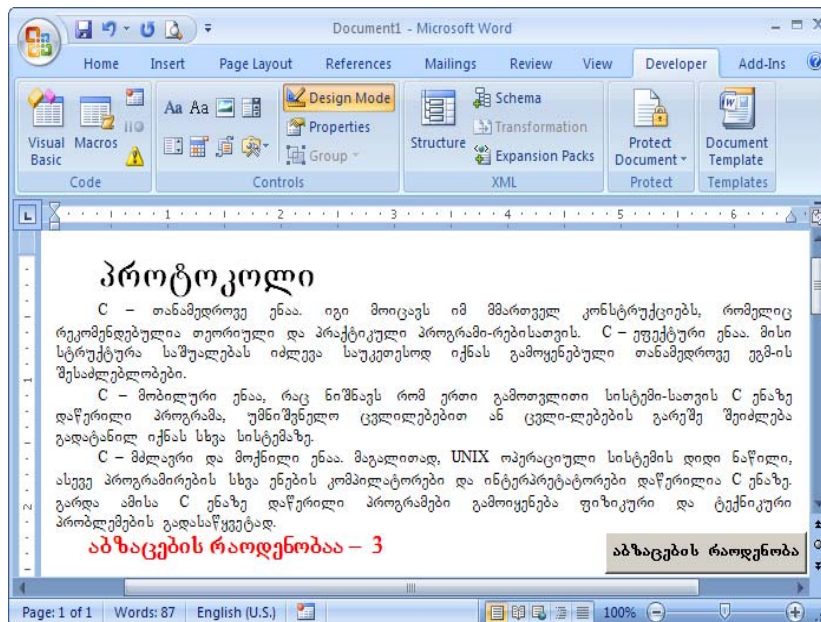
```
Private Sub CommandButton1_Click()  
Dim k As Integer  
Dim Text As String  
Dim SATAURI As Range  
Dim REZULTAT As Range  
k = ActiveDocument.Paragraphs.Count  
Text = "abzacebis raodenobaa - " & k & ". "  
ActiveDocument.Paragraphs(k).Range.InsertParagraphAfter
```

```

ActiveDocument.Paragraphs(k + 1).Range.InsertBefore Text
Set RESULTAT = ActiveDocument.Paragraphs(k + 1).Range
With RESULTAT
RESULTAT.Font.Name = "Arial"
RESULTAT.Font.Size = 14
RESULTAT.Font.ColorIndex = wdDarkRed
End With
Set SATAURI = ActiveDocument.Range(Start:=0, End:=0)
With SATAURI
SATAURI.InsertBefore "protokoli"
SATAURI.Font.Name = "Arial"
SATAURI.Font.Size = 24
SATAURI.InsertParagraphAfter
End With
End Sub

```

შესრულების შედეგი წარმოდგენილია სურათზე:



დავალება: აბზაცში „ა“ სიმბოლოს რაოდენობის დათვლა

დავალება შეიძლება შესრულდეს იგივე ტექსტისათვის, რაც უკვე განვიხილეთ. პროგრამამ უნდა განსაზღვროს აბზაცის ნომერი, რომელშიც უნდა დაითვალოს **a** სიმბოლოს რაოდენობა; უნდა დაითვალოს **a** სიმბოლოს რაოდენობა; დოკუმენტის ბოლოში გამოიტანოს შესაბამისი შეტყობინება.

სამუშაოს მიმდევრობა:

1. განათავსეთ ფურცელზე ღილაკი CommandButton2;
2. ღილაკს დაარქვით “**a** სიმბოლოს რაოდენობა აბზაცში”;
3. შეუცვალეთ ღილაკს ფორმატი, რათა შესაძლებელი იყოს მისი გადაადგილება;
4. ორჯერ დააწკაპუნეთ ღილაკზე, გამოჩნდება პროგრამული კოდის ფანჯარა.

სამუშაოს შესრულების ალგორითმი:

1. ცვლადების გამოცხადება:
 - b – სტრიქონული ცვლადია დოკუმენტში არჩეული აბზაცის განსათავსებლად;
 - k - ცვლადია დოკუმენტში აბზაცების რაოდენობის შესანახად;
 - dl – ცვლადია აბზაცის სიგრძის შესანახად;
 - Text – ცვლადია შედეგი – შეტყობინების ფორმირებისათვის;
 - Nab – ცვლადია აბზაცის ნომრის შესატანად, რომელშიც უნდა მოხდეს დათვლა;
 - I – ცვლადია ციკლის ორგანიზებისათვის;
 - Kol – ცვლადია **a** სიმბოლოს დასათვლელად;
 - REZULTAT – ცვლადია, რომელიც განსაზღვრავს არეს სათაურის გამოსატანად;

2. შევიტანოთ აბზაცის ნომერი, დიალოგური ფანჯრის გამოყენებით;
3. განვსაზღვროთ დოკუმენტში აბზაცების რაოდენობა;
4. შევამოწმოთ შეტანილი აბზაცის ნომრის არსებობა და თუ ასეთი არ არის გამოვცეთ შესაბამისი შეტყობინება;
5. ხ ცვლადში შევიტანოთ მთელი აბზაცის ტექსტი;
6. განვსაზღვროთ აბზაცის სიგრძე;
7. გავუკეთოთ a სიმბოლოს მომძებნ ციკლს ორგანიზება;
8. შევქმნათ შეტყობინების გამოტანის ტექსტი;
9. ჩავსვათ დამატებითი აბზაცი დოკუმენტის ბოლო აბზაცის შემდეგ, შეტყობინების გამოსატანად;
10. გამოვიტანოთ შეტყობინების ტექსტი მომზადებულ აბზაცში;
11. დავაყენოთ განსაზღვრული შრიფტი, შეტყობინების ტექსტის ზომა და ფერი.

პროგრამული კოდი:

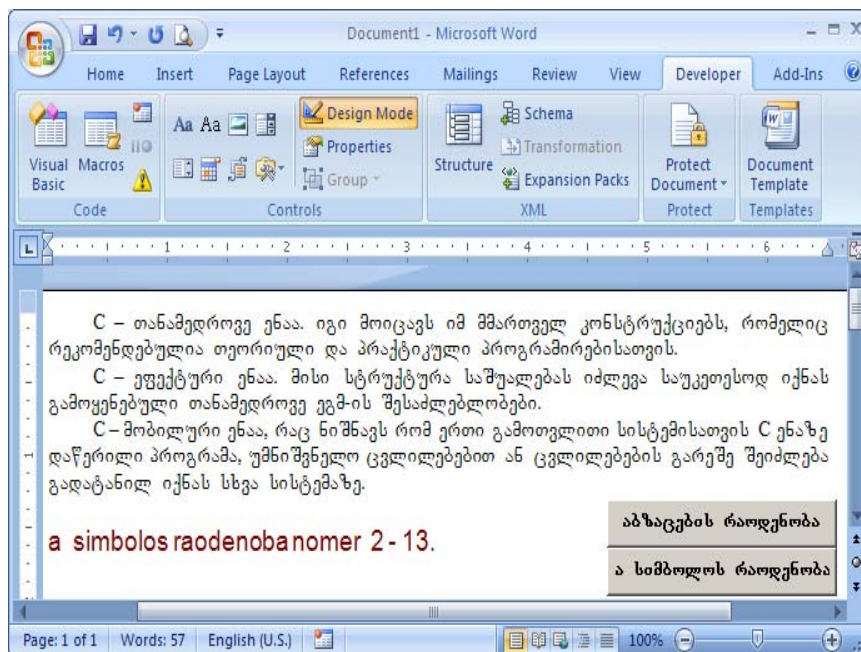
```
Private Sub CommandButton1_Click()
Dim b As String
Dim k As Integer
Dim dl As Long
Dim Text As String
Dim Nab As Integer
Dim i As Long
Dim REZULTAT As Range
kol = 0
Nab = InputBox("SeitaneT abzacis nomeri", "viTvliT a simbolos")
k = ActiveDocument.Paragraphs.Count
If Nab > k Then
MsgBox "teqstSi ar aris aseTi abzaci", 48, "gafriTxileba"
End
End If
```

```

b = ActiveDocument.Paragraphs(Nab).Range
dl = Len(b)
For i = 1 To dl
If Mid(b, i, 1) = "a" Or Mid(b, i, 1) = "A" Then kol = kol + 1
Next i
MsgBox kol
Text = "a simbolos raodenoba nomer " & Nab & " - " & kol & ".
ActiveDocument.Paragraphs(k).Range.InsertParagraphAfter
Set RESULTAT = ActiveDocument.Paragraphs(k + 1).Range
With RESULTAT
REZULTAT.InsertBefore Text
REZULTAT.Font.Name = "Arial"
REZULTAT.Font.Size = 14
REZULTAT.Font.ColorIndex = wdDarkRed
End With
End Sub

```

შესრულების შედეგი წარმოდგენილია სურათზე:



თემა №13. ტექსტებთან მუშაობა

დავალება: აბზაცში წინადადებების დათვლა

ამოცანა მდგომარეობს მასში, რომ პროგრამულად განვსაზღვროთ აბზაცის ნომერი, რომელშიც წინადადებების მეტი რაოდენობაა; დიალოგური ფანჯრის გამოყენებით გამოვიდეს შეტყობინება აბზაცის ნომრისა და წინადადებების რაოდენობის შესახებ; ეს აბზაცი გამოიყოს წითელი ფერით. სამუშაო შევასრულოთ იგივე ტექსტზე.

სამუშაოს მიმდევრობა.

1. ტექსტის ქვემოთ განვათავსოთ ღილაკი CommandButton;
2. ღილაკს დავარქვათ “აბზაცში წინადადებები”;
3. ღილაკს შევუცვალოთ ფორმატი, რათა თავისუფლად გადაადგილდეს;
4. ორჯერ დავაწკაპუნოთ ღილაკზე, რითაც გადავიღივართ პროგრამული კოდის ფანჯარაში.

სამუშაოს შესრულების ალგორითმი:

1. ცვლადების გამოცხადება:
k – ცვლადი დოკუმენტში აბზაცების რაოდენობის შესანახად;
kol – ცვლადი აბზაცში წინადადებების რაოდენობის შესანახად;
i - ციკლის პარამეტრი;
Mas() – მასივია აბზაცებში წინადადებების რაოდენობის შესატანად;
Pas – ცვლადია ტექსტი-პასუხის თავისი შედეგებით ფორმირებისათვის;
Max – ცვლადია აბზაცში წინადადებების მაქსიმალური რაოდენობის შესანახად;

ind – ცვლადია მაქსიმალური რაოდენობის წინადადებების შემცველი აბზაცის ნომრის შესანახად;

REZULTAT – ცვლადია, რომელიც განსაზღვრავს აბზაცის არეს, რომელშიც უნდა იქნას გამოყენებული ფორმატირების ელემენტები.

მასივის გამოცხადებისას ჩვენთვის უცნობია დოკუმენტში აბზაცების რაოდენობა. ამისათვის მასივის გამოცხადებას ვაწარმოებთ ორ ეტაპად: პირველი ეტაპი გამოცხადება ზომის გარეშე: Dim Mas() As Integer; მეორე ეტაპი – დოკუმენტში აბზაცების რაოდენობის განსაზღვრის შემდეგ: ReDim Mas(k) As Integer. ასეთ მასივებს ჰქვია დინამიური.

გარდა ამისა რათა მასივის ინდექსაცია დაიწყო 1-დან და არა 0-დან მოხერხებულობის თვალსაზრისით, რედაქტორში ფურცლის დასაწყისში მიუთითეთ ბრძანება: “Option Base 1”.

2. დოკუმენტში აბზაცების რაოდენობის განსაზღვრა;
3. მასივის ზომის ხელახლა განსაზღვრა;
4. ციკლის ორგანიზება, რომელიც განსაზღვრავს აბზაცებში წინადადებების რაოდენობას და მათ განზომილებას შეიტანს მასივში, რომელიც შეესაბამება აბზაცის ნომერს.
5. ციკლის ორგანიზება, რომელიც იპოვის აბზაცებში წინადადებების უდიდეს რაოდენობას (ფორმირებულ მასივთან მუშაობა) სტანდარტული ალგორითმის მიხედვით.
6. შეტყობინების ტექსტის ფორმირება;
7. დიალოგურ ფანჯარაში შედეგის შეტყობინების გამოტანა;
8. აბზაცის არეს განსაზღვრა ფორმატირებისათვის;
9. მაქსიმალური რაოდენობის წინადადების შემცველი აბზაცისათვის განსაზღვრული შრიფტის, ზომის და ფერის დაყენება.

პროგრამული კოდი:

```
Private Sub CommandButton3_Click()  
Dim k As Integer  
Dim kol As Integer  
Dim i As Integer  
Dim Mas() As Integer  
Dim pas As String  
Dim Max As Integer  
Dim ind As Byte  
Dim REZULTAT As Range  
kol = 0: k = 0  
k = ActiveDocument.Paragraphs.Count  
ReDim Mas(k) As Integer  
Rem პროგრამის დასაწყისი  
For i = 1 To k  
kol = ActiveDocument.Paragraphs(i).Range.Sentences.Count  
Mas(i) = kol  
Next i  
Max = Mas(1)  
ind = 1  
For i = 2 To k  
If Mas(i) > Max Then  
Max = Mas(i)  
ind = i  
End If  
Next i  
Rem შედეგების გამოტანა  
pas = "cinadadebebis kvelaze didi raodenoba aris nomriT " & ind & "  
abzacSi - " & Max  
MsgBox pas  
Set REZULTAT = ActiveDocument.Paragraphs(ind).Range  
With REZULTAT  
REZULTAT.Font.Name = "Arial"
```

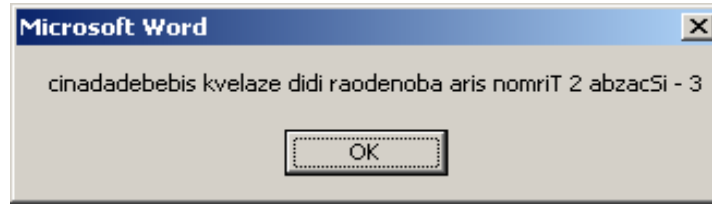

REZULTAT.Font.Size = 12

REZULTAT.Font.ColorIndex = wdDarkRed

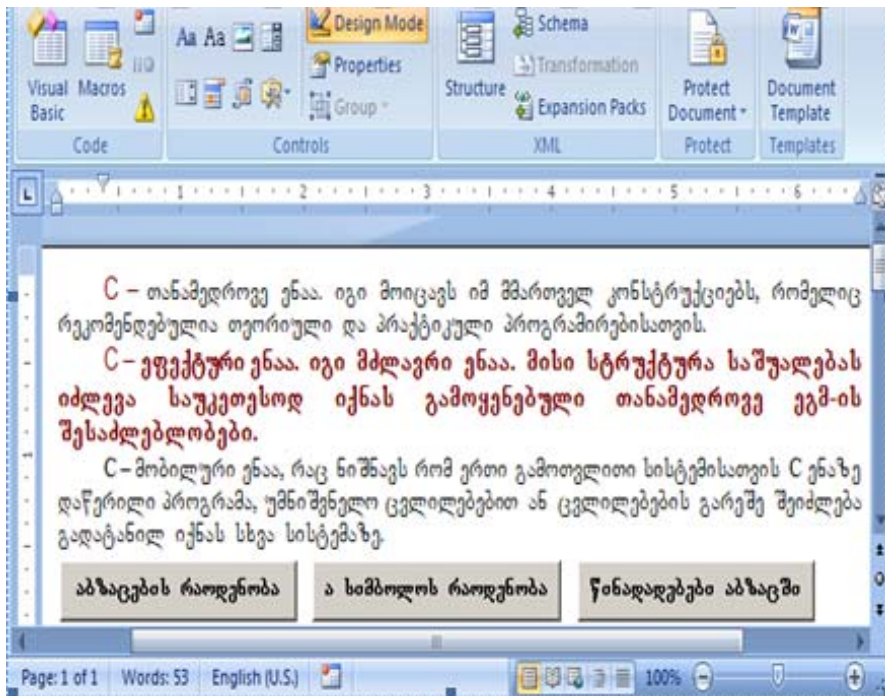
End With

End Sub

დიალოგური ფანჯარა შედეგის შეტყობინებით გამოიყურება შემდეგი სახით:



დავალების შესრულების შედეგი წარმოდგენილია სურათზე:



თემა №14. VBA ენა PowerPoint დანართისათვის

PowerPoint დანართში **VBA** ენის დახმარებით შეიძლება მრავალი ამოცანის ავტომატიზება. განვიხილოთ სამომხმარებლო ფორმის გამოყენება ტესტების დაპროექტებისათვის. არსებობს არა მარტო ინფორმაციის შეტანის საშუალება, არამედ მართვის ელემენტების გამოყენების შესაძლებლობები. ამისათვის საჭიროა გავითვალისწინოთ ორი მარტივი წესი:

1. მაკროსი საჭიროა დაენიშნოს ინსტრუმენტების პანელის ღილაკებს;
2. მართვის ელემენტები საჭიროა განთავსდეს დიალოგურ ფანჯრებში.

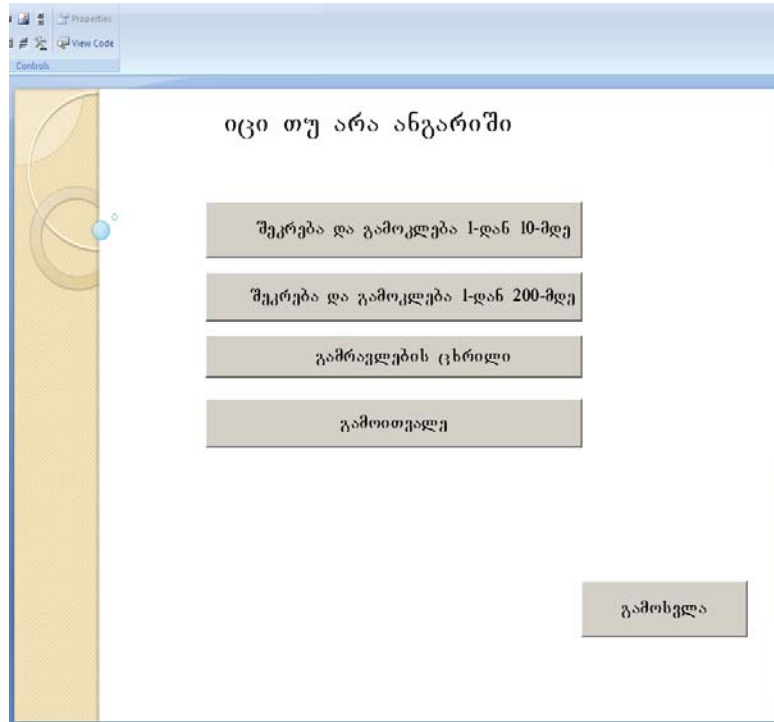
დავალება: “იცი თუ არა ანგარიში”

შევქმნათ პრეზენტაცია, რომლის მეშვეობითაც შეიზღებთ შემოწმდეს და შეფასდეს ზეპირი ანგარიშის ჩვევები.

სამუშაოს მიმდევრობა:

1. გავხსნათ **PowerPoint** და შევქმნათ სატიტულო სლაიდი. განვათავსოთ მასზე ხუთი ღილაკი:

- შეკრება და გამოკლება 1-დან 10-მდე;
- შეკრება გამოკლება 1-დან 200-მდე;
- გამრავლების ცხრილი;
- გამოითვალე;
- გამოსვლა.



○

2. დაგწეროთ სამომხმარებლო ფორმის გამოძახების პროგრამები:

```
Private Sub CommandButton1_Click()
```

```
UserForm1.Show
```

```
End Sub
```

```
Private Sub CommandButton2_Click()
```

```
UserForm2.Show
```

```
End Sub
```

```
Private Sub CommandButton3_Click()
```

```
UserForm3.Show
```

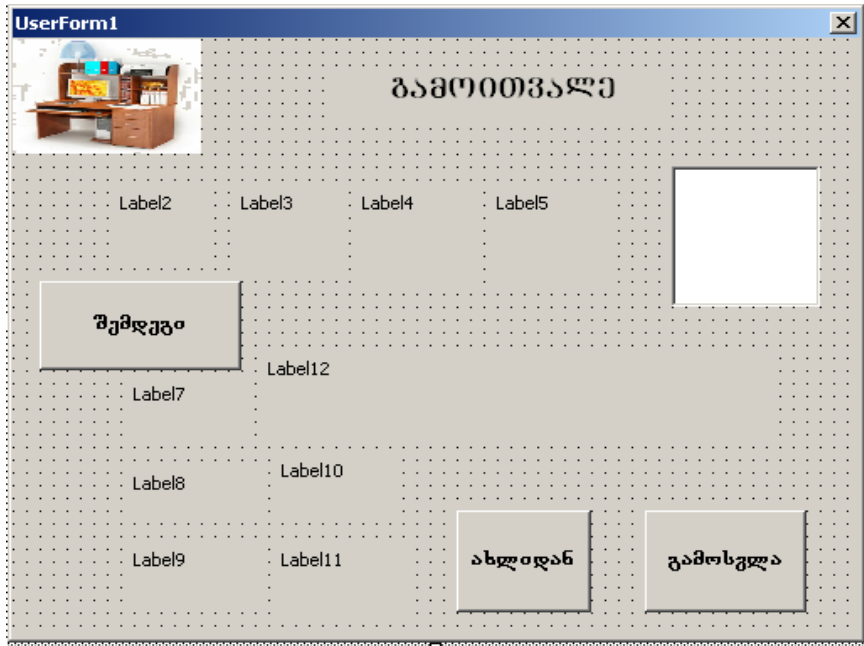
```
End Sub
```

Private Sub CommandButton4_Click()

UserForm4.Show

End Sub

3. “გამოსვლა” ღილაკმა დაასრულოს ჩვენება.
4. დავაპროექტოთ Visual Basic რედაქტორში სამომხმარებლო ფორმა UserForm1: “შეკრება და გამოკლება 1-დან 10-მდე”. დავაფორმატოთ ყველა ელემენტი.



4. შემდგომ დავამუშაოთ ღილაკებზე დაჭერის ხლომილებების პროცედურები:

შემოვიტანოთ ცვლადები:

- a* — ცვლადია პირველი რიცხვისათვის
- b* — ცვლადია მეორე რიცხვისათვის
- R* — ცვლადია შედეგის მისაღებად

v, n — ცვლადებია სწორი და არასწორი პასუხების დასათვლელად

f — ამორჩევის ოპერატორის ცვლადია

პროგრამა დილაკისათვის “შემდეგი”:

გლობალური ცვლადების გამოცხადება:

Public a As Integer

Public b As Integer

Public R As Integer

Public v As Integer

Public n As Integer

Public f As Integer

Private Sub CommandButton1_Click()

$f = f + 1$

Select Case f

Case 1

‘a,b ცვლადებს ვანიჭებთ მნიშვნელობებს (0;9)

ინტერვალთან შემთხვევითი რიცხვების გენერირებით

$b = \text{Int}(10 * \text{Rnd}())$

$a = \text{Int}(10 * \text{Rnd}())$

‘ვამზადებთ ჩანაწერს - ან + u =

Label3.Caption = "-"

Label5.Caption = "="

‘ განვსაზღვროთ ამ რიცხვებიდან უდიდესი, იგი იქნება პირველი რიცხვი, ხოლო უმცირესი – მეორე რიცხვი.

If a > b Then

Label2.Caption = a

Label4.Caption = b

Else

Label2.Caption = b

```

Label4.Caption = a
End If
' ვითვლით შედეგს
R = Abs(a - b)
Case 2
' ვადარებთ შედეგს და შეტანილ პასუხს, ვითვლით
რაოდენობას.
If Val(R) = Val(TextBox1) Then
v = v + 1
Label12.Caption = "sworia"
Else
n = n + 1
Label12.Caption = "arasworia"
End If
Case 3
CLS
Label12.Caption = ""
b = Int(10 * Rnd())
a = Int(10 * Rnd())
Label2.Caption = a
Label4.Caption = b
Label3.Caption = "+"
Label5.Caption = "="
R = a + b
Case 4
If Val(R) = Val(TextBox1) Then
v = v + 1
Label12.Caption = "sworia"
Else
n = n + 1
Label12.Caption = "arasworia"

```

```

End If
Case 5
CLS
Label12.Caption = ""
b = Int(10 * Rnd())
a = Int(10 * Rnd())
Label3.Caption = "-"
Label5.Caption = "="
If a > b Then
Label2.Caption = a
Label4.Caption = b
Else
Label2.Caption = b
Label4.Caption = a
End If
R = Abs(a - b)
Case 6
If Val(R) = Val(TextBox1) Then
v = v + 1
Label12.Caption = "sworia"
Else
n = n + 1
Label12.Caption = "arasworia"
End If
Case 7
CLS
Label12.Caption = ""
b = Int(10 * Rnd())
a = Int(10 * Rnd())
Label2.Caption = a
Label4.Caption = b

```

```

Label3.Caption = "+"
Label5.Caption = "="
R = a + b
Case 8
If Val(R) = Val(TextBox1) Then
v = v + 1
Label12.Caption = "sworia"
Else
n = n + 1
Label12.Caption = "arasworia"
End If
Case 9
CLS
Label12.Caption = ""
b = Int(10 * Rnd())
a = Int(10 * Rnd())
Label2.Caption = a
Label4.Caption = b
Label3.Caption = "+"
Label5.Caption = "="
R = a + b
Case 10
If Val(R) = Val(TextBox1) Then
v = v + 1
Label12.Caption = "sworia"
Else
n = n + 1
Label12.Caption = "arasworia"
End If
‘ გამოგვაქვს სწორი და არასწორი პასუხების რაოდენობა
Case 11

```



```

Label7.Caption = "Tqveni Sedegi"
Label8.Caption = "sworia"
Label10.Caption = Str(v)
Label9.Caption = "arasworia"
Label11.Caption = Str(n)
If v = 5 Then Label12.Caption = " yoCaR!!!" Else
Label12.Caption =
"kidev imecadine!!!"
Case 12
UserForm1.Hide
End Select
End Sub

```

პროგრამა დილაკისათვის “ახლიდან”

```

Private Sub CommandButton2_Click()
CLS
' განულებო შედეგის გამომთვლელ ცვლადებს
n = 0
v = 0
' წარწერებში ვშლით ინფორმაციას
Label10.Caption = ""
Label11.Caption = ""
Label12.Caption = ""
Label2.Caption = ""
Label4.Caption = ""
Label3.Caption = ""
Label5.Caption = ""
Label7.Caption = ""
Label8.Caption = ""

```

Label9.Caption = ""

f = 0

End Sub

Sub CLS()

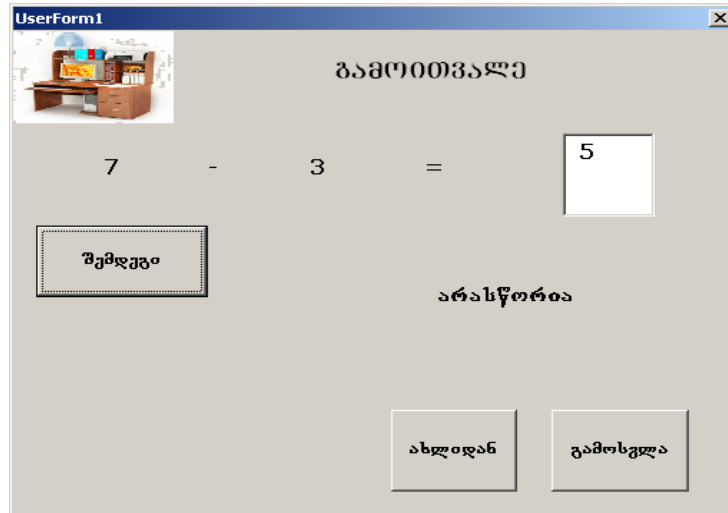
TextBox1.Text = ""

End Sub

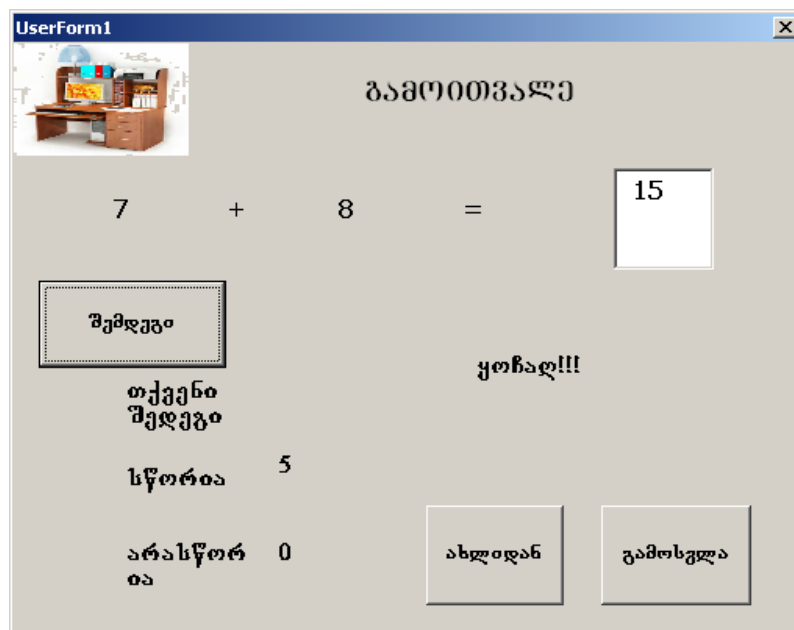
მუშა მდგომარეობაში ფორმა სწორი შედეგის შემთხვევაში გამოიყურება შემდეგი სახით:

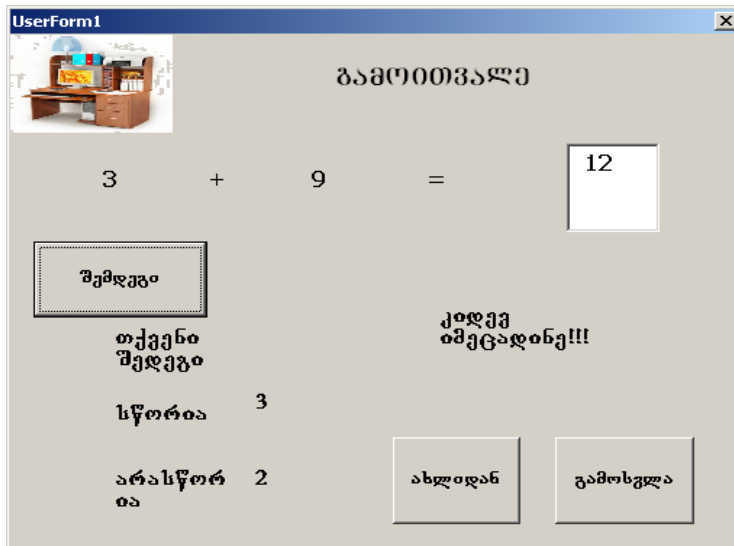
The screenshot shows a VBA UserForm window titled "UserForm1" with a small icon of a desk and computer. The form has a light gray background. At the top center, the text "ბამოთმალე" is displayed. Below this, the equation "5 - 2 =" is shown. To the right of the equals sign is a white text box containing the number "3". On the left side, there is a button with a dashed border labeled "შემდეგი". In the center, there is a button labeled "სწორია". At the bottom, there are two buttons: "ახლიდან" on the left and "გამოსვლა" on the right.

არასწორი შედეგის შემთხვევაში ფორმა გამოიყურება შემდეგი სახით:



შედეგის ფორმას აქვს სახე:





5. დააპროექტეთ სამომხმარებლო ფორმები: UserForm2, UserForm3, UserForm4.

6. დაწერეთ ღილაკებისათვის პროგრამები.

UserForm4 “გამოითვალე” შეიძლება იყოს ასეთი:



7. შეამოწმეთ პროექტის მუშაობისუნარიანობა და შეინახეთ სამუშაო.

თემა №15. ტესტირების სისტემის დაპროგრამება

დავალება: ვიზუალური ინტელექტის ტესტი

შევქმნათ პრეზენტაცია, რომლის დახმარებითაც შეიძლება შემოწმდეს და შეფასდეს ვიზუალური ინტელექტი.

სამუშაოს მიმდევრობა:

1. გაეხსნათ PowerPoint , შევქმნათ სატიტულო სლაიდი.
2. სლაიდზე განვათავსოთ ღილაკი წარწერით “განსაზღვრე საკუთარი ვიზუალური ინტელექტი” და დავწეროთ მისთვის პროგრამული კოდი, რომელიც გამოიძახებს UserForm1 სამომხმარებლო ფორმას.

```
Private Sub CommandButton1_Click()
```

```
UserForm1.Show
```

```
End Sub
```

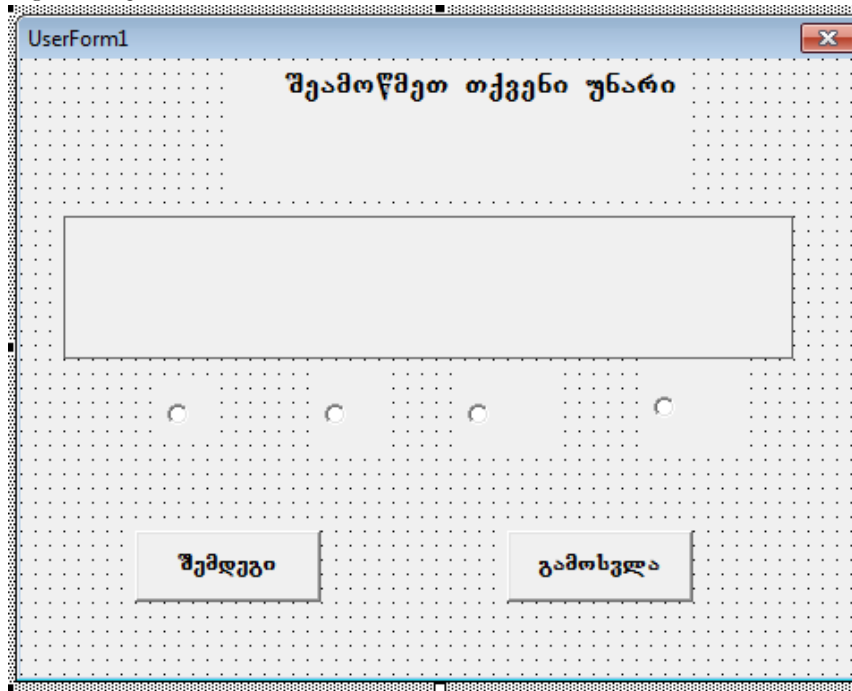
3. შევქმნათ “გამოსვლა” ღილაკი და შესაბამისი პროგრამული კოდი, რომელიც დაასრულებს ჩვენებას.



4. Visual Basic რედაქტორში დავაპროექტოთ სამომხმარებლო ფორმა შემდეგი მართვის ელემენტებით:

მართვის ელემენტი	დანიშნულება
Label1	წარწერა
OptionButton1 OptionButton2 OptionButton3 OptionButton4	პასუხის ვარიანტების არჩევა
CommandButton1 CommandButton2	ღილაკები

საწყისი სამომხმარებლო ფორმა წარმოდგენილია სურათზე:



პროგრამა ღილაკისათვის “შემდეგი”

Public I As Integer

Public x As Integer

```

' x ცვლადი ითვლის სწორი პასუხების რაოდენობას
Private Sub CommandButton1_Click()
i = i + 1
Select Case i
Case 1
' "შემდეგი" დილაკზე პირველად დაჭერისას იცვლება
წარწერა და სურსათი
Label1.Caption = "romeli ar Seesabameba"
UserForm1.Image1.Picture = LoadPicture("C:\test1.JPG")
Case 2
' "შემდეგი" დილაკზე მეორედ დაჭერისას დაითვლება
სწორი პასუხები, დაეშვება ალმები OptionButton
კომპონენტებზე და იცვლება სურათი.
If OptionButton4.Value = True Then x = x + 1
OptionButton1.Value = False
OptionButton2.Value = False
OptionButton3.Value = False
OptionButton4.Value = False
UserForm1.Image1.Picture = LoadPicture("C:\test2.jpg")
Case 3
' "შემდეგი" დილაკზე მესამედ დაჭერისას დაითვლება
სწორი პასუხები, დაეშვება ალმები OptionButton
კომპონენტებზე და იცვლება სურათი.
If OptionButton1.Value = True Then x = x + 1
OptionButton1.Value = False
OptionButton2.Value = False
OptionButton3.Value = False
OptionButton4.Value = False
UserForm1.Image1.Picture = LoadPicture("C:\test3.jpg")
Case 4

```

“შემდეგი” ღილაკზე მეოთხედ დაჭერისას დაითვლება სწორი პასუხები, დაეშვება ალმები **OptionButton** კომპონენტებზე და იცვლება სურათი.
If **OptionButton3.Value = True** Then $x = x + 1$
OptionButton1.Value = False
OptionButton2.Value = False
OptionButton3.Value = False
OptionButton4.Value = False
UserForm1.Image1.Picture = LoadPicture("C:\test4.jpg")

Case 5

“შემდეგი” ღილაკზე მესამედ დაჭერისას დაითვლება სწორი პასუხები, დაეშვება ალმები **OptionButton** კომპონენტებზე და იცვლება სურათი.
If **OptionButton3.Value = True** Then $x = x + 1$
OptionButton1.Value = False
OptionButton2.Value = False
OptionButton3.Value = False
OptionButton4.Value = False
UserForm1.Image1.Picture = LoadPicture("C:\test5.jpg")

Case 6

“შემდეგი” ღილაკზე მეექვსედ დაჭერისას დაითვლება სწორი პასუხები, დაეშვება ალმები **OptionButton** კომპონენტებზე და გამოვა შედეგი.
If **OptionButton4.Value = True** Then $x = x + 1$
OptionButton1.Value = False
OptionButton2.Value = False
OptionButton3.Value = False
OptionButton4.Value = False
Label1.Caption = "shfaseba " & x

Case 7

```
UserForm1.Hide
```

```
End Select
```

```
End Sub
```

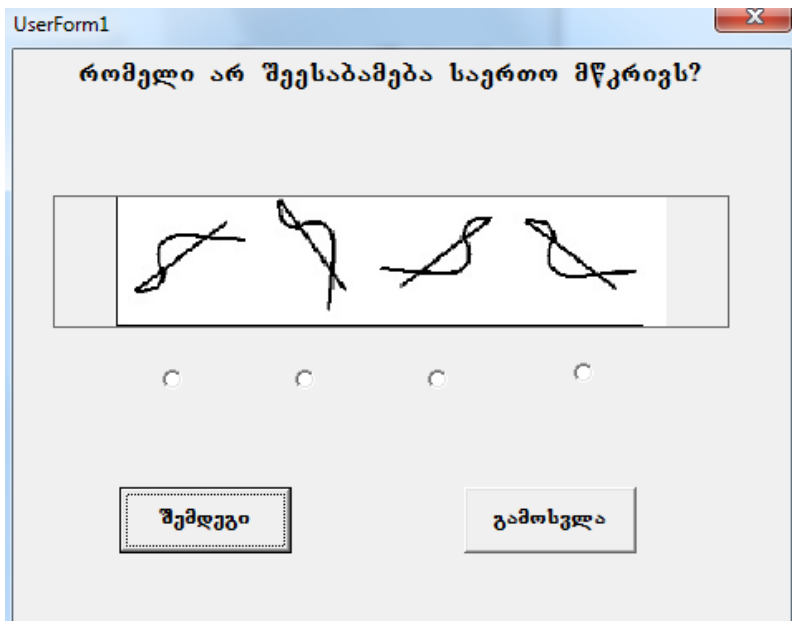
პროგრამა “გამოსვლა” ღილაკისათვის:

```
Private Sub CommandButton2_Click()
```

```
UserForm1.Hide
```

```
End Sub
```

”შემდეგი” ღილაკზე დაჭერისას ფორმა გამოიყურება შემდეგი სახით:



სამუშაოს შესრულების შედეგი მოცემულია სურათზე:

UserForm1

შეფასება 5

შემდეგი

გამოსვლა

დავალება: შექმენით გამოკითხვის ტესტი. შეარჩიეთ მასალა გამოკითხვის ტესტისათვის. იგი შეიძლება იყოს ტესტი საგნის მიხედვით.

UserForm1

რომელია საქართველოს ფართობი?

50000 კვ.კმ

70000 კვ.კმ

75000 კვ.კმ

80000 კვ.კმ

შემდეგი

გაუქმება

გამოსვლა

ლიტერატურა

1. Уокенбах, Джон. Профессиональное программирование на VBA в Excel 2003. : Пер. с англ. – М. : Издательский дом “Вильямс”, 2005. – 800 с. : ил. – Парал.тит. англ. ISBN 5-8459-0771-3 (рус.)

2. Excel 2010 Power Programming with VBA by John Walkenbach Published by Wiley Publishing, Inc. 111 River Street Hoboken, NJ 07030-5774 www.wiley.com

იგეგმება ავტორთა მიერ წარმოდგენილი ნახატი

გადაეცა წარმოებას 15.02.2012. ქაღალდის ზომა 60X84 1/8. პირობითი
ნაბეჭდი თაბახი 9. ტირაჟი 100 ეგზ.

