

ი. ქართველიშვილი

ავტომატიზებული ამომცნობი სისტემის პროგრამული კომპლექსი

რეზიუმე

ნაშრომში წარმოდგენილია ავტომატიზებული ამომცნობი სისტემის პროგრამული კომპლექსი. მოყვანილია კონსტრუქტორები, პროცედურები, ფუნქციები და თითოეული მეთოდის ალგორითმისთვის დამუშავებულია პროგრამული კოდები, რომლებიც წარმოდგენილია სქემატურად და თითოეული მათგანი აღწერილია თავისი ფუნქციონალური დანიშნულებით.

**საკვანძო სიტყვები:** კონსტრუქტორები, პროცედურები, ფუნქციები.

**1. შესავალი:**

ხელნაწერი სიმბოლოების ავტომატიზებული ამომცნობი სისტემის პროგრამულ კომპლექსში იგულისხმება ის პროგრამები, რომლებიც ამოცანის ამოსხნის ალგორითმებს წარმოადგენს და მისი ჩაწერა ხდება ობიექტ-ორიენტირებული დაპროგრამების ენის საშუალებით.

ავტომატიზებული ამომცნობი სისტემის პროგრამული კომპლექსის დამუშავებისას გამოყენებულია ორი ძირითადი კლასი: გამოსახულების კლასი – TImage და სიმბოლოს კლასი – TSymbol. თითოეული კლასისთვის შექმნილია კონსტრუქტორები, პროცედურები, ფუნქციები და თითოეული მეთოდის ალგორითმისთვის დამუშავებულია პროგრამული კოდები, რომლებიც მოყვანილია სქემატურად და თითოეული მათგანი აღწერილია თავისი ფუნქციონალური დანიშნულებით.

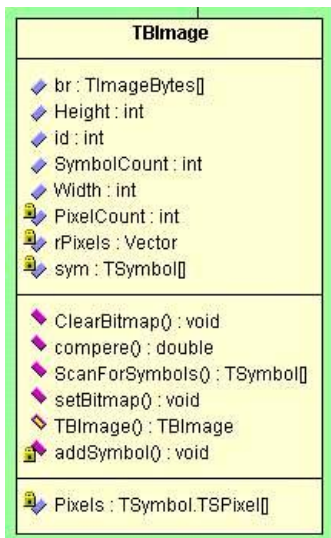
**2. ძირითადი ნაწილი:**

ხელნაწერი სიმბოლოების ავტომატიზებული ამომცნობი სისტემა შექმნილია ობიექტ-ორიენტირებული დაპროგრამების ენაზე – Borland Delphi 6.0, რომელიც წარმოადგენს ვიზუალური გაწყობის ენას. ამ სისტემას გააჩნია დიდი საშუალება მართოს მონაცემთა ბაზები, როგორც პერსონალური კომპიუტერისთვის, ასევე განაწილებული ლოკალური და გლობალური ქსელებისთვის. პროგრამულ უზრუნველყოფაში გამოყენებულია აგრეთვე პროგრამები, რომლებიც ქმნიან და არეგულირებენ მონაცემთა ბაზების მართვის სისტემებს. ისინი მომხმარებელს უადვილებენ მათი ამოცანების ჩაწერას და გადაწყვეტას.

აღნიშნულ სისტემაში გამოყენებულია მბმს (მონაცემთა ბაზების მართვის სისტემა) Paradox 7.0, რომელიც მისი სიმარტივის გამო გამოირჩევა დიდი სისწრაფით. განვიხილოთ ავტომატიზებული ამომცნობი სისტემის პროგრამული კომპლექსის სტრუქტურა (ნახ.1).

თითოეული კლასისთვის ცალკეულად მოვიყვანოთ მათი სტრუქტურა და ავლწეროთ კლასებში გამოყენებული კონსტრუქტორები, ფუნქციები და პროცედურები თავისი ფუნქციონალური

დანიშნულებით. განვიხილოთ TImage – გამოსახულების კლასი (ნახ. 2). აღნიშნულ კლასში გამოყენებულია:



ნახ. 2

კონსტრუქტორი: Create (Bitmap : TImage BytesArray), რომლის საშუალებითაც იქმნება მთლიანი გამოსახულება;

პროცედურები: 1) SetBitmap (Bitmap:TImageBytesArray) – იქმნება ბიტური გამოსახულება; 2) ClearMatrix () – გამოსახულების გასუფთავება.

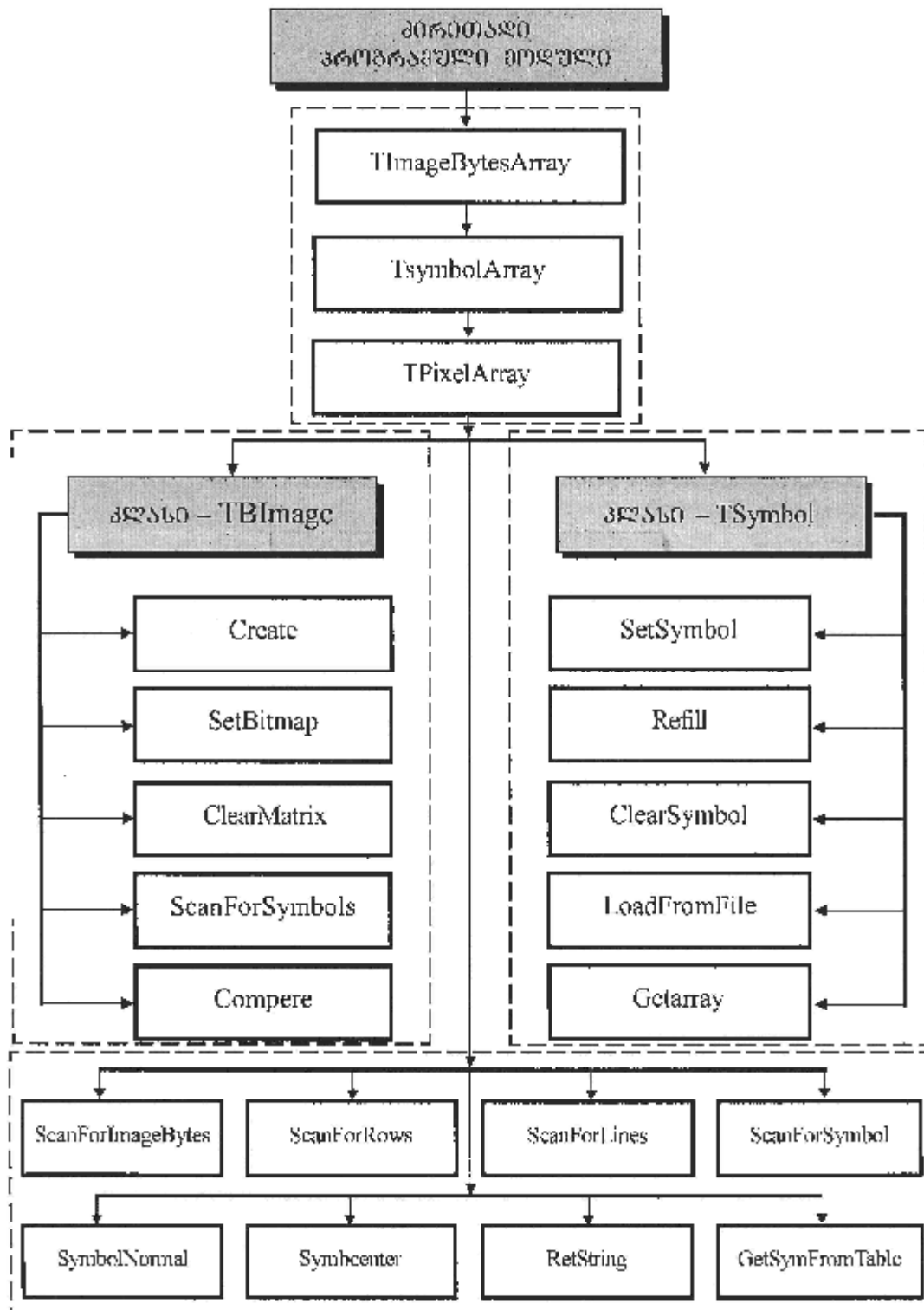
ფუნქციები: 1) ScanForSymbols():TSymbolArray – ეძებს მთლიან გამოსახულებაში ცალკეულ სიმბოლოებს და გადააქვს ის სიმბოლოების მასივებად; 2) Compere (Sym1, Sym2 : TImageBytesArray) – ორი სიმბოლოს შედარება, რომელიც აბრუნებს შედარების პროცენტულ მაჩვენებელს.

სიმრავლეები: 1) TImageBytesArray = Array of TImageBytes – სიმბოლოს ბაიტების სიმრავლე; 2) TsymbolArray = Array of Tsymbol – სიმბოლოების სიმრავლე.

განვიხილოთ TSymbol – სიმბოლოს კლასი (ნახ. 3). აღნიშნულ კლასში გამოყენებულია: პროცედურები: 1) SetSymbol (Pixels:TPixelArray) – პიქსელების

კოორდინატების გადაცემა, რის საფუძველზეც იქმნება სიმბოლო; 2) Refill () – სიმბოლოს გარეგანი დაგლუვება; 3) ClearSymbol () – სიმბოლოს დამუშავება; 4) LoadFromFile () – სიმბოლოს ამოღება ფაილიდან. ფუნქციები: 1) Getarray () : TImageBytesArray – აბრუნებს ბაიტების ორგანიზაციის

მასივს. სიმრავლეები: 1) TPixelArray = Array of TPixel – პიქსელების სიმრავლე; 2) TImageBytesArray = Array of TImageBytes – სიმბოლოს ბაიტების სიმრავლე.



ნახ. 1

განვიხილოთ პროგრამულ კომპლექსში გამოყენებული მეთოდები თავისი ფუნქციონალური დანიშნულებებით:

ScanForImageBytes () – გადააქვს სკანირებული გამოსახულება ბაიტების მატრიცაში;

ScanForRows () – მატრიცის სეგმენტაცია სტრიქონებად;

ScanForLines () – თითოეული სტრიქონის სეგმენტაცია სიტყვებად;

ScanForSymbol () – თითოეული სეგმენტირებული სიტყვიდან სიმბოლოების ამოღება;

SymbolNormal () – თითოეული სეგმენტირებული სიმბოლოს ნორმალიზაცია;

TSymbol	
◆ Height : int	
◆ ID : int	
◆ Index : int	
◆ Line : int	
◆ Percent : double	
◆ Row : int	
◆ sMinX : int	
◆ sMinY : int	
◆ SymbolCount : int	
◆ Width : int	
◆ br : TSymbol.TImageBytes[]	
◆ ClearSymbol() : void	
◆ getArray() : TSymbol.TImageBytes[]	
◆ LoadFromFile() : void	
◆ Refill() : void	
◆ setArray() : void	
◆ setSymbol() : void	
◆ setSymbolFromMemo() : void	
◆ TSymbol() : void	

Symbcenter () – სიმბოლის ცენტრის მოძებნა თითოეული სიმბოლოსათვის და მათი კლასტიფიკაცია სახეობების მიხედვით;

RetString () – ერთი სიმბოლოს მატრიცა, რომელიც აბრუნებს სიმბოლოს ტექსტურ მნიშვნელობას, თუ ვერ იპოვა ბაზაში ასეთი სიმბოლო, სისტემა შედის მომხმარებელთან დიალოგში და ახდენს ახალი სიმბოლოს დამატებას ბაზაში;

GetSymFromTable (Hei, Wid, Pix : Integer) : TsymbolArray – ფილტრავს ბაზაში არსებულ სიმბოლოებს შესაბამისი კრიტერიუმების მიხედვით (Hei – სიმაღლე, Wid – სიგანე, Pix – პიქსელების რაოდენობა) და მოამზადებს შედარებისთვის.

### 3. დასკვნა:

დამუშავებული იქნა ხელნაწერი სიმბოლოების ავტომატიზებული ამომცნობი სისტემის პროგრამული კომპლექსი და მისი სტრუქტურა. შექმნილი იქნა პროგრამული კომპლექსის კონსტრუქტორები, პროცედურები, ფუნქციები და მეთოდები.

### 4. ლიტერატურა:

1. ვერულავა ო., ხუროძე რ. ამომცნობი სისტემების თეორიის საფუძვლები. სტუ, თბილისი, 2001.
2. დოჭვირი ბ. ალაბთობის თეორიისა და მათემატიკური სტატისტიკის ელემენტები. თსუ, თბილისი 1986.
3. Tappert C.C. Recognition system for run-on handwritten characters. United States patent, 4,731,857, March 1988.
4. ქართველიშვილი ი. ორიგინალებისა და უცნობი ხელნაწერი სიმბოლოების შედარების კომპიუტერული ანალიზი. თბილისი, სტუ-ს შრომები №4 (432), 2000. გვ. 82-87.

**КАРТВЕЛИШВИЛИ ИОСЕБ**

## **АВТОМАТИЗИРОВАННАЯ ОПОЗНАТЕЛЬНАЯ ПРОГРАММНАЯ СИСТЕМА**

### **Резюме**

В труде предоставлена автоматизированная опознательная программная система. Приведены конструкторы, процедуры, функции и единичные методы для алгоритмов, разработаны программные коды, которые представлены схематический и описано функциональное назначение каждого.

*KARTVELISHVILI IOSEB*

## **PROGRAM GROUP OF AUTOMATIC IDENTIFICATION SYSTEM**

### **Resume**

Work describes program group of automatic identification system, constructors, procedures and functions. The program codes are worked out for each algorithm method; They are represented schematically and are described by their function.