

УДК 004.02

DOI 10.52928/2070-1624-2022-39-11-40-46

**УПРОЩЕННЫЙ АЛГОРИТМ ГРАФИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ТРЕХМЕРНЫХ ДАННЫХ ПРИ RAD-РАЗРАБОТКЕ ОКОННЫХ ПРИЛОЖЕНИЙ В СРЕДЕ PascalABC.NET**

**Я. Н. СУПРУНЧИК, П. А. КОХАН, канд. техн. наук, доц. В. В. ДАВЫДОВСКАЯ**  
(Мозырский государственный педагогический университет имени И. П. Шамякина)

*Разработан алгоритм отображения трехмерной графической зависимости в среде PascalABC.NET с использованием класса Bitmap посредством «поверхностей уровней» при RAD-разработке пользовательских оконных приложений (Windows Form Application). Проведен сравнительный анализ графической интерпретации результатов вычисления в системах компьютерной математики MATLAB, MathCAD и PascalABC.NET.*

**Ключевые слова:** трехмерная графика, поверхности уровней, PascalABC.NET, RAD-разработка, Windows Form Application, класс Bitmap.

**Введение.** При проведении любого научного исследования неотъемлемой частью является анализ и интерпретация полученных результатов. Графическое представление и визуализация – один из наиболее приоритетных способов представления полученных данных.

Современные интегрированные математические пакеты (например, MATLAB, MathCAD, Maple) обладают широкими возможностями по работе с декартовыми и полярными графиками, графиками поверхностей и т. д. Однако работа в данных пакетах имеет определенную специфику, которая требует специальных навыков работы в них. Для многих исследователей, студентов и школьников использование систем компьютерной математики является проблематичным.

В настоящее время широко используются программы с графическим интерфейсом. Для повышения скорости создания таких программ применяют среды быстрой разработки.

Среды быстрой разработки представляют собой развитие концепции интегрированных сред разработки. Интегрированная среда разработки – это пакет программ. В нем, помимо компилятора и компоновщика, содержатся редактор исходного кода и отладчик. Редактор исходного кода представляет собой текстовый редактор с подсветкой синтаксиса [1].

В основе быстрых средств разработки лежит идея визуального проектирования интерфейса при помощи редактора форм. Сама работа программы определяется совокупностью обработчиков событий. Событием называется действие пользователя или операционной системы [2].

Часто возникает необходимость внедрения в такие приложения научной или математической графики, что требует от программиста глубоких знаний по использованию сторонних библиотек и модулей и специфике работы с ними.

Студентам педагогического профиля, обучающимся по специальности 1-02 05 01 «Математика и информатика» на дисциплинах «Методы алгоритмизации», «Технологии программирования», «Языки программирования школьной информатики» предлагается для изучения среда PascalABC.NET, которая позволяет писать программы на высокоуровневом языке программирования в легкой, компактной и понятной форме. PascalABC.NET может использоваться как в образовательных, так и в научных целях.

В рамках дисциплины «Программирование в визуализированных средах» изучаются основы программирования в мультимедийной среде «Scratch», которая позволяет пробудить у учащихся интерес к созданию компьютерных программ и освоить современную технологию объектно-ориентированного программирования.

В среде PascalABC.NET также содержится дизайнер форм, который позволяет разрабатывать пользовательские оконные приложения (Windows Form Application) с использованием RAD-технологии (Rapid Application Development – быстрое создание приложений).

В рамках сотрудничества вузов с учреждениями среднего образования, работы филиалов кафедр, при прохождении студентами педагогической и преддипломной практики, нередко возникают вопросы, связанные с разработкой и оформлением исследовательских работ учащихся школ. Одной из часто возникающих проблем является помещение в приложения, созданные в PascalABC.NET, пространственных поверхностей как функций двух переменных, в то время как введение в объектно-событийное программирование, разработка приложений с графическим интерфейсом изучаются в школе только в 11-м классе [3].

Решением проблемы может быть использование класса Bitmap, который предназначен для работы с растровыми изображениями, содержит методы создания и обработки точечных изображений, а также обладает возможностью построения так называемых контурных графиков, представляющих собой совокупность линий, каждая из которых соответствует одинаковому значению функции, зависящей от двух переменных (изолинии).

**Теория.** В статье описан разработанный алгоритм получения трехмерной графической зависимости в среде PascalABC.NET. Одним из способов отображения таких зависимостей являются «поверхности уровней», которые представлены во всех современных интегрированных пакетах.

Основная идея заключается в том, что сначала высчитываются значения матрицы, соответствующей заданной функции в известной области, затем весь интервал значений от минимального до максимального значения функции разбивается на десять уровней, и при «прорисовке» функции точка (пиксель) окрашивается в соответствующий цвет в зависимости от значения функции в данной точке.

Цвета для создания контурного графика определены заранее с помощью массива:

colors: **array of Color**:= (Color.Purple, Color.Teal, Color.Blue, Color.Aqua, Color.SkyBlue, Color.Green, Color.Lime, Color.Yellow, Color.Orange, Color.Red);

Предполагается, что в приложение добавлен элемент PictureBox, в который будет добавляться изображение, представляющее собой контурный график функции двух переменных.

В отличие от двумерных графиков, которые используют дискретные аргументы и функции, трехмерные графики требуют предварительного создания матрицы значений поверхности.

Построение контурного графика будет осуществляться поточно с использованием метода SetPixel, который содержится в классе Bitmap.

По аналогии с построением декартова графика [3] приложение также будет предусматривать ввод интервалов построения для переменных  $x \in [x_0; x_n]$  и  $y \in [y_0; y_n]$  с помощью элементов TextBox. Цветовая шкала (ColorMap): соответствие цветов значениям функции организовано с помощью элемента DataGridView, в котором ячейки первого столбца будут заливаться цветами в соответствии с ранее определенным массивом цветов, во втором столбце будут выведены соответствующие значения функции.

Построение графика будет осуществляться по нажатию на кнопку «Вычислить».

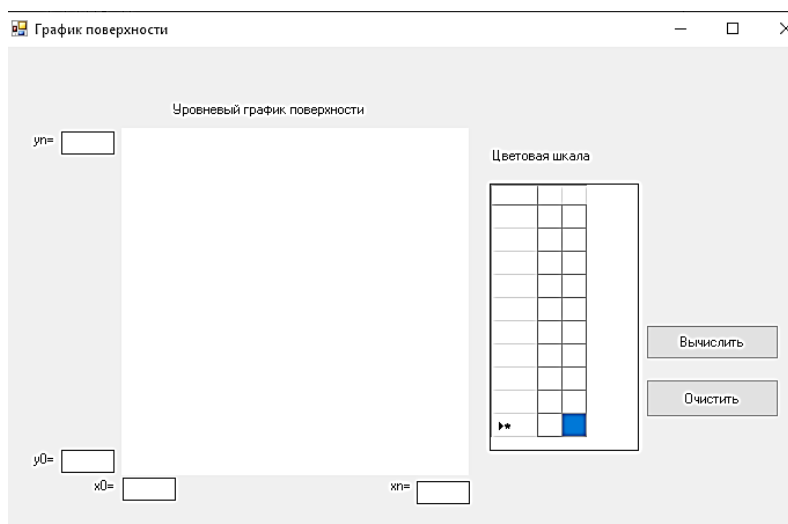


Рисунок 1. – Интерфейс приложения, созданного в PascalABC для построения графика функции

Элемент PictureBox имеет свою систему координат. По умолчанию начало отсчета находится в левом верхнем углу формы. Ось  $x$  направлена вправо, ось  $y$  – вниз.

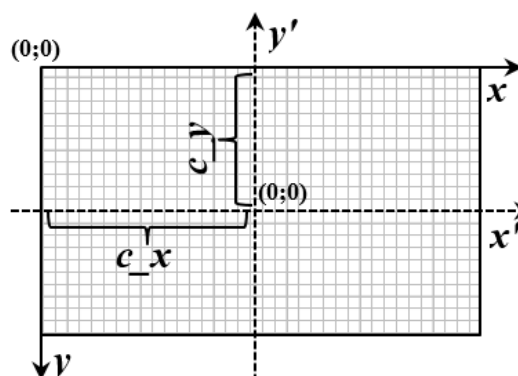


Рисунок 2. – Преобразование системы координат для построения графика функции

Как показано на рисунке 2, при построении графика поверхности использовалось стандартное расположение декартовых осей координат. Для этого был выполнен ряд преобразований: смещение начала координат на величины  $c_x$  и  $c_y$  вдоль осей  $x$  и  $y$  соответственно, поворот оси  $y$ , а также для использования произвольного интервала построения было проведено масштабирование по размеру элемента PictureBox, координаты умножались на масштабирующий коэффициент  $k$ . В нашем приложении область рисования имеет размеры  $300 \times 300$  пикселей и одинаковый шаг изменения  $x$  и  $y$ , поэтому для обеих осей коэффициент  $k$  будет одинаковым.

```
k:=trunc(PictureBox1.Width/(xn-x0));
c_x:=PictureBox1.Width div 2;
c_y:=PictureBox1.Height div 2;
```

Пиксель – это минимально возможная точка, физические размеры которой зависят от установленной в операционной системе разрешающей способности экрана. В приложении каждый пиксель соответствует определенному значению функции.

Для выполнения всех последующих программных блоков первоначально следует вычислить матрицу, соответствующую значениям функции.

Для примера рассмотрим построение поверхности уровней, соответствующей функции  $z(x, y) = x^2 - y^2$  в области  $x \in [-10; 10]$ ,  $y \in [-10; 10]$ :

```
for var i:=1 to N do begin
  x[i]:=x0+(i-1)*h;
  y[i]:=y0+(i-1)*h;
end;
for var i:=1 to N do begin
  for var j:=1 to N do
    z[i,j]:=sqr(x[i])-sqr(y[j]);
  end;
```

Далее для создания цветовой карты необходимо вычислить интервал между максимальным и минимальным значениями функции:

```
max:=z[1,1];
min:=z[1,1];
for var i:=1 to N do begin
  for var j:=1 to N do begin
    if z[i,j]>max then max:=z[i,j];
    if z[i,j]<min then min:=z[i,j];
  end;
end;
int:=(max-min)/10; //10 цветовых уровней
```

Затем с помощью элемента DataGridView формируется цветовая шкала:

```
DataGridView1.RowCount:=10;
DataGridView1.ColumnCount:=2;
for var i:=0 to 9 do begin
  DataGridView1.Item[0,i].Style.BackColor:=colors[i];
  DataGridView1.Item[1,i].Value:=trunc((i)*int+min);
end;
```

Последним программным блоком является пиксельная прорисовка контурного графика поверхности:

```
for var i:=1 to N do begin
  for var j:=1 to N do begin
    zk:=z[i,j];
    if (trunc(-k*y[j])+c_y>=0) and (trunc(-k*y[j])+c_y<2*c_y) then begin
      cvet_point(zk,int,min,colors,cvp);
      bm.SetPixel(trunc(k*x[i])+c_x,trunc(-k*y[j])+c_y,cvp);
    end;
  end;
end;
```

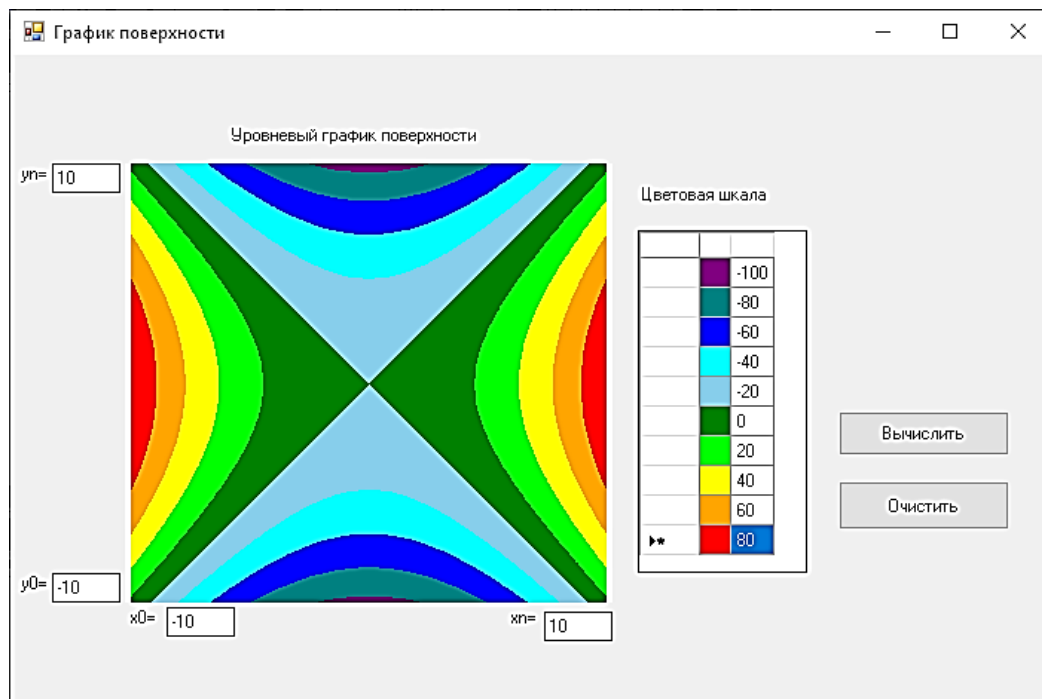
В описанном выше фрагменте листинга присутствует пользовательская процедура `cvet_point`, которая отвечает за выбор соответствующего цвета для закраски пикселей:

```

procedure cvet_point(zk, int,min: real;colors:array of Color; var cvp:Color);
begin
  cvp:=colors[0];
  for var i:=1 to 9 do
    if zk>=(i)*int+min then cvp:=colors[i];
end;
    
```

Также в программе введены следующие обозначения:

bm: Bitmap;  
 cvp: Color;



**Рисунок 3. – Результат выполнения разработанного алгоритма построения поверхностей уровня**

Следует отметить, что при решении многих задач такой способ отображения поверхностей является приоритетным (например, задачи распределения температуры, задачи упругости и др.)

Для примера рассмотрим задачу о нахождении стационарного распределения температуры в квадратной пластине со стороной 1, описываемого уравнением Лапласа

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

с краевыми условиями вида

$$u(0, y)=0, (0 \leq y \leq 1), u(1, y)= \sin(y), (0 \leq y \leq 1),$$

$$u(x, 0)=0, (0 \leq x \leq 1), u(x, 1)= \sin(x), (0 \leq x \leq 1).$$

Рассмотрим решение данной задачи в среде PascalABC методом верхней релаксации [4] с последующим графическим отображением распределения температуры в квадратной пластине с использованием приведенного выше алгоритма. Отличительной особенностью этого построения будет являться то, что начало осей координат сместится в нижний левый угол (рисунок 4).

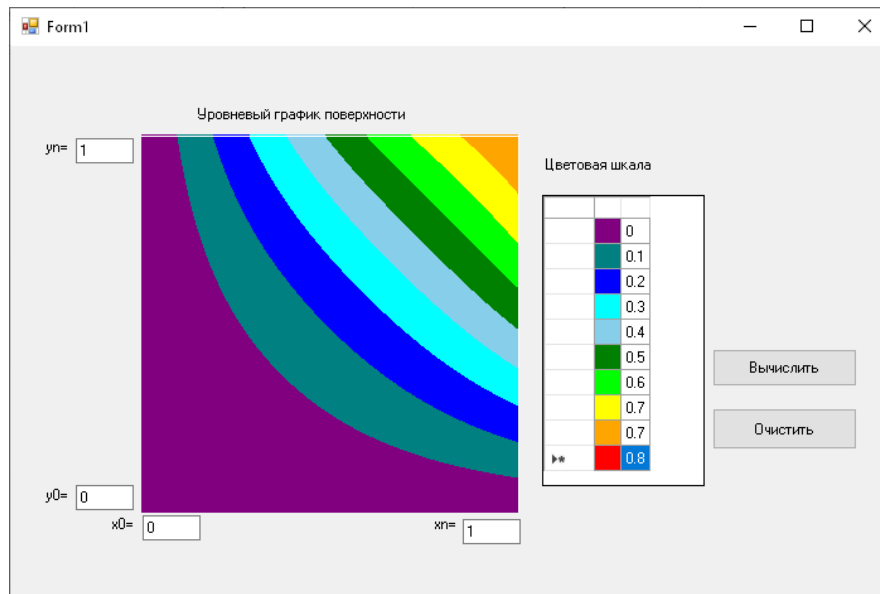


Рисунок 4. – Решение уравнения Лапласа в среде PascalABC.NET

Для сравнения рассмотрим решение данной задачи в PDEToolbox, среда MATLAB (рисунок 5).

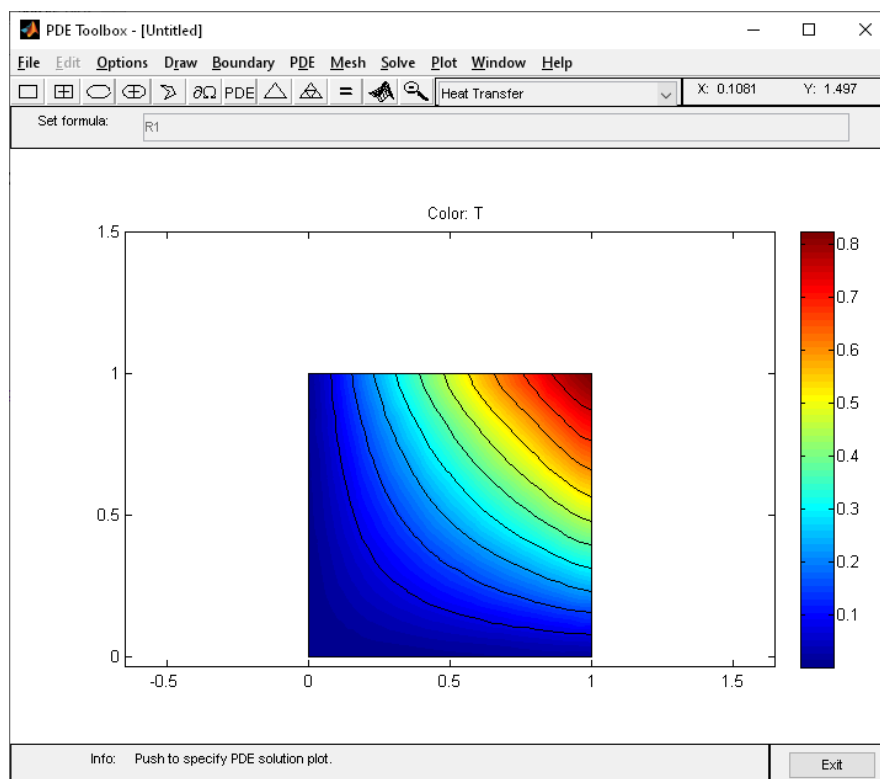


Рисунок 5. – Решение уравнения Лапласа в MATLAB

При сопоставлении рисунков 4 и 5 можно сделать вывод, что предложенный алгоритм графического отображения в среде PascalABC.NET не уступает встроенным возможностям MATLAB. Полученное распределение может быть подробно проанализировано, что позволит сделать корректные выводы. Тем не менее следует отметить тот факт, что скорость вычислений будет выше в математических пакетах, однако предложенный алгоритм является очень простым и не требует глубоких познаний в программировании и может быть реализован в любой системе программирования, имеющей возможности работы с растровой пиксельной графикой с небольшими изменениями в синтаксисе основного программного кода. Данный

алгоритм был также реализован в среде визуального программирования Scratch с использованием инструмента «Перо» [5].

После подключения библиотеки System.Windows.Forms.DataVisualization станет доступным элемент Chart (Диаграмма) из WindowsForms.

Работа с Chart не изучается в школьном курсе информатики, но может существенно упростить построение декартовых графиков и диаграмм без пересчета координат, рисования осей координат и нанесения шкалы. Проведя некоторые преобразования, элемент Chart можно использовать и для построения графиков поверхностей функций двух переменных, однако в некоторых случаях такое отображение уступает описанному выше алгоритму, который был нами разработан, т. к. требуется дополнительное преобразование цветowych шкал, настройки осей, поворота и освещения элемента Chart, а начинающие программисты, тем более учащиеся школ, не всегда обладают этими навыками.

Для примера рассмотрим построение графика поверхности, представленного на рисунке 3, с использованием элемента Chart (рисунок 6):

```
//удаление серии, добавленной по умолчанию
chart1.Series.RemoveAt(0);
//добавление необходимого количества серий на диаграмму
for var i:=1 to N do
  chart1.Series.Add('Series' + i.ToString());
//построение диаграммы
for var i:=1 to N do begin
  x[i]:=x0+(i-1)*h;
  for var j:=1 to N do begin
    y[j]:=y0+(j-1)*h;
    Chart1.Series['Series' + i.ToString()].Points.AddXY(j,f(x[i],y[j]));
  end;
end;
```

Присутствующая во фрагменте функция  $f$  отвечает за вычисление значений функции  $f(x, y)$ :

```
function f (x,y:real):real;
begin
  f:=x*x-y*y;
end;
```

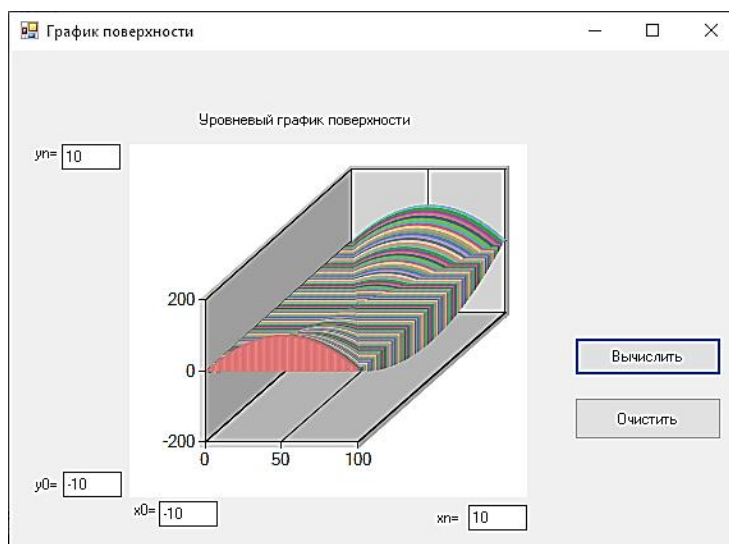


Рисунок 6. – График поверхности, построенный с использованием элемента Chart

В [3] предлагается способ построения декартовых графиков, гистограмм и круговых диаграмм с использованием классов Witmar и Graphics, однако в данном случае, в отличие от графиков поверхностей, использование Chart может заметно упростить эту задачу.

**Заключение.** В статье показано, что разработанный алгоритм отображения трехмерной графики на Windows Pascal Forms в ряде случаев дает лучшее представление об особенностях ее распределения и тем самым упрощает ее описание и анализ.

Все теоретические результаты, полученные в данной статье, могут служить методическим дополнением к современному учебнику информатики для 11-го класса в процессе обучения основам работы с Windows Forms в среде PascalABC.NET с целью оформления пользовательских приложений с необходимостью добавления высокоуровневой графики, а также при подготовке исследовательских работ и работе с одаренными учащимися на факультативных занятиях.

#### ЛИТЕРАТУРА

1. Гагарина, Л. Г. Технология разработки программного обеспечения / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул. – М. : Форум : Инфра-М, 2016. – 423 с.
2. Лукин, В. В. Технология разработки программного обеспечения : учеб. пособие / В. В. Лукин, В. Н. Лукин, Т. В. Лукин. – М. : Вузовская книга, 2015. – 214 с.
3. Информатика : учеб. пособие / В. М. Котов [и др.]. – Минск : Народная асвета, 2021. – 112 с.
4. Самарский, А. А. Математическое моделирование: Идеи. Методы. Примеры / А. А. Самарский, А. П. Михайлов. – 2-е изд., испр. – М. : Физматлит, 2002. – 320 с.
5. Косярский, А. А. Элементы программирования с использованием среды Scratch 2.0 / А. А. Косярский. – Казань : Бук, 2020. – 266 с.

#### REFERENCES

1. Gagarina, L. G., Kokoreva, E. V., & Visnadul, B. D. (2016). *Tekhnologiya razrabotki programmnogo obespecheniya [Software Development Technology]*. Moscow: Forum, Infra-M. (In Russ.).
2. Lukin, V. V., Lukin, V. N., & Lukin, T. V. (2015). *Tekhnologiya razrabotki programmnogo obespecheniya [Software Development Technology]*. Moscow: Vuzovskaya kniga. (In Russ.).
3. Kotov, V. M., Lapo, A. I., Bykadorov, Yu. A., & Voitekhovich, E. N. (2021). *Informatika [Computer science]*. Minsk: Narodnaya asveta. (In Russ.).
4. Samarskii, A. A., & Mikhailov, A. P. (2002). *Matematicheskoe modelirovanie: Idei. Metody. Primery [Mathematical modeling: Ideas. Methods. Examples]*. Moscow: Fizmatlit. (In Russ.).
5. Косярский, А. А. (2020). Элементы программирования с использованием среды Scratch 2.0. Казань: Бук. *Kosyarskii, A. A. (2020). Elementy programmirovaniya s ispol'zovaniem sredy Scratch 2.0 [Programming elements using Scratch 2.0 environment]*. Kazan': Buk.

Поступила 07.09.2022

#### SIMPLIFIED ALGORITHM FOR GRAPHIC REPRESENTATION OF THREE-DIMENSIONAL DATA IN RAD DEVELOPMENT OF WINDOW APPLICATIONS IN THE PASCALABC.NET ENVIRONMENT

*Y. SUPRUNCHYK, P. KOHAN, V. DAVYDOUSKAYA*  
(*Mozyr State Pedagogical University named after I. P. Shamyakin*)

*An algorithm for presentation a three-dimensional graphical dependency in the PascalABC.NET environment using the Bitmap class, by means of "contour surfaces" in the RAD development of custom window applications (Windows Form Application) has been developed. A comparative analysis of the graphical interpretation of the calculation results in the computer mathematics systems MATLAB, MathCAD and PascalABC.NET has been carried out.*

**Keywords:** *3D-graphics, contour surfaces, PascalABC.NET, RAD development, Windows Form Application, Bitmap class.*