УДК 004 **Информационные технологии**

Корягин Сергей Викторович, *кандидат технических наук, доцент кафедры «Прикладные информационные технологии», Российский Технологический Университет МИРЭА*

Рябов Константин Андреевич, *студент кафедры «Прикладные информационные технологии», Российский Технологический Университет МИРЭА*

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ ВЕКТОРНОЙ ГРАФИКИ

Аннотация: В данной статье предлагаются к рассмотрению проблемно-ориентированные языки программирования векторной-графики. С помощью этих языков можно создавать, хранить и редактировать изображения. Предоставлен сравнительный анализ существующих решений и их критическая оценка. На основе проведенного анализа сформированы требования к разрабатываемому проблемно-ориентированному языку и соответствующее ему транслирующее средство. Описание исследуемого объекта предлагается в виде набора команд аффинных преобразований. Даются формальное описание проблемно-ориентированного языка и примеры его использования.

Ключевые слова: проблемно-ориентированные языки программирования, векторная графика, аффинные преобразования, форма Бэкуса-Наура.

Abstract: In this article, problem-oriented programming languages of vector graphics are proposed for consideration. You can use these languages to create, store, and edit images. A comparative analysis of existing solutions and their critical evaluation is provided. Based on the analysis, the requirements for the developed problem-oriented language and the corresponding translation tool are formed. The

description of the object under study is proposed in the form of a set of affine transformation commands. A formal description of the problem-oriented language and examples of its use are given.

Keywords: problem-oriented programming languages, vector graphics, affine transformations, Backus-Naur form.

Введение. С появлением вычислительной техники создано великое множество языков программирования. Из года в год, разрабатывая новые конструкции, внедряя их в прошлые наработки, появляются более гибкие и удобные средства человеко-машинного взаимодействия. Каждый язык имеет свои достоинства и недостатки (скорость выполнения, объем занимаемой памяти, соответствие/несоответствие решению конкретных задач), что и является предпосылкой для дальнейшего развития языков.

Одним из подобного рода специальных направлений являются языки программирования векторной графики [1; 2; 5]. Они рассматриваются как проблемно-ориентированные или узкоспециализированные, главной задачей которых является работа с векторными изображениями любой сложности. Такие изображения, по сравнению с растровыми, могут масштабироваться без потерь и занимать значительно меньше памяти.

Редакторы векторной графики представляют собой набор инструментов, упрощающий процесс рисования. С помощью редакторов можно вращать, перемещать, отражать, растягивать, выполнять основные аффинные преобразования над объектами, изменять их порядок и комбинировать примитивные в более сложные объекты.

Работа с таким программным обеспечением выполняется с помощью мыши, графического планшета или клавиатуры. Последний подход встречается чаще.

Обзор реализованных решений. С помощью рассматриваемых различных языков описания векторной графики можно создавать, хранить и редактировать векторные изображения [6].

Одним из таких инструментов является язык описания страниц PostScript (PS) [7; 8], который в основном используется в издательских системах. В 1982 году Adobe разработал его для использования в принтерах как альтернатива растровым изображениям. Он реализован в виде небольшой серии текстовых команд, способной создавать простые двумерные изображения. Однако этот язык имеет свои недостатки:

a) для каждой платформы необходимо писать свой PostScript из-за различающихся параметров вывода;

b) язык считается нестабильным;

c) в некоторых случаях при изменении разрешения изображения возможно искажение формы контура кривой Безье;

d) наблюдается избыточность за счет хранения невидимых частей изображения.

На сегодняшний день наблюдается распространение современной версии этого языка – Encapsulated PostScript (EPS).

В 1986 году вышел в свет популярный инструмент gnuplot [3] для визуализации разных видов как двухмерных, так и трехмерных графиков. Это свободная и кроссплатформенная утилита, обладающая собственной системой команд и способная работать интерактивно (в режиме командной строки) или выполнять скрипты, читаемые из файлов.

В 1998 году был разработан язык векторной разметки (Vector Markup Language - VML) [4] и представлен компаниями Microsoft, AutoDesk, HewlettPackard и Macromedia. Написанные на нем объекты помещаются внутрь Web-страниц, среди обычного HTML-кода. Фигуры VML при этом являются такими же объектами для скриптовых языков типа JavaScript, как и объекты HTML, таким образом, их можно динамически модифицировать, а также можно значительно уменьшить код страницы, заменяя объекты тегами VML. Несмотря на то, что этот язык представляется довольно удобным, мощным и безопасным, он не получил широкого распространения.

С 1999 года консорциумом Всемирной паутины (World Wide Web Consortium – W3C) был создан язык разметки масштабируемой векторной графики (Scalable Vector Graphics - SVG) [6; 9]. Он предназначен для описания двумерной векторной и смешанной векторной/растровой графики в формате XML и поддерживает как неподвижную, так и анимированную интерактивную графику. Этот формат очень хорошо зарекомендовал себя в виду хорошей читаемости, масштабируемости, совместимости с таблицей стилей CSS, а также способности сохранения текста (с возможностью дальнейшего редактирования). Данный вариант языка включает в себя все преимущества XML (но, к сожалению, и недостатки).

Постановка задачи. Исходя из представленных выше преимуществ и недостатков рассмотренных средств, можно перечислить следующие важные качества проблемно-ориентированного языка описания векторной графики:

- простота и интуитивность описания изображения
- интерфейс, ориентированный на пользователя-непрограммиста,
- возможность динамической модификации изображений

Приняв за основу эти качества, можно поставить задачу разработки проблемно-ориентированного языка для описания векторной графики. Отличительной особенностью этого языка является возможность выполнять слияние и отсечение контуров фигур.

Форма Бэкуса-Наура проблемно-ориентированного языка

Язык = **операция**..**операция**

операция = [**выполнение**!**определение**] ";"

выполнение = ["draw" "(" ПЧ(figure!point)..ПЧ(figure!point) ")"]![**"rotate"** "(" **переменная** "," **ПЧ**(num) ")"]

определение = </ **тип** /> **переменная** "=" **ПЧ**(тип)

тип = "var"!"num"!"figure"!"point"!"color"!"vector"

ПЧ = </"-"/> **блок** зн1.. **блок**

зн1 = "+"!"-"

блок = **объект** зн2.. **объект**

зн1 = "*"!"/"

объект = **переменная** ! **фигура** ! **точка** ! **вектор** ! **цвет** ! **число** ! ["(" **ПЧ** ")"]

переменная = буква </символ..символ/>

символ = буква ! цифра

точка = "(" **ПЧ**(num) "," **ПЧ**(num) ")" </**индекс**/>

вектор = "[" **ПЧ**(num) "," **ПЧ**(num) "]" </**индекс**/>

фигура = "{" **ПЧ**(point)","..**ПЧ**(point) "}" [</**индекс**/>!**параметр фигуры**]

параметр фигуры = [["FillColor"!"DotColor!"StrokeColor"] </**параметр цвета**/>] ! "StrokeColor" ! "DotRadius"

параметр цвета = "Red" ! "Green" ! "Blue" ! "Alpha"

цвет = "[" **ПЧ**(num) "," **ПЧ**(num) "," **ПЧ**(num) </ "," **ПЧ**(num) /> "]"

индекс = "[" **ПЧ**(num) "]"

число = цел.ч </ "." цел.ч. />

цел.ч. = цифра..цифра

буква = "a"!..!"я"!"A"!..!"Я"!"a"!..!"z"!"A"!..!"Z"

цифра = "0"!..!"9"

Каждая операция, разделяемая символом «;», может либо объявлять и присваивать новые значения объектам, либо рисовать и поворачивать выбранные объекты. Объекты в данном языке могут иметь один из пяти типов данных: число, точка, вектор, фигура и цвет. При их объявлении можно также использовать динамическую типизацию с «var». Язык обладает минимальным математическим аппаратом, состоящим из аддитивных и мультипликативных функций.

Пример 1 - «pac-man» (рисунок 1). В начале строится квадрат, у которого есть повернутая и масштабированная копия. Потом оригинальная фигура масштабируется и получает смещение по заданному вектору. К ней добавляются 4 точки, образуя восьмиугольник. Из полученного отсекается

раннее полученная копия на заданном смещении. Конечная фигура отображается на экран.

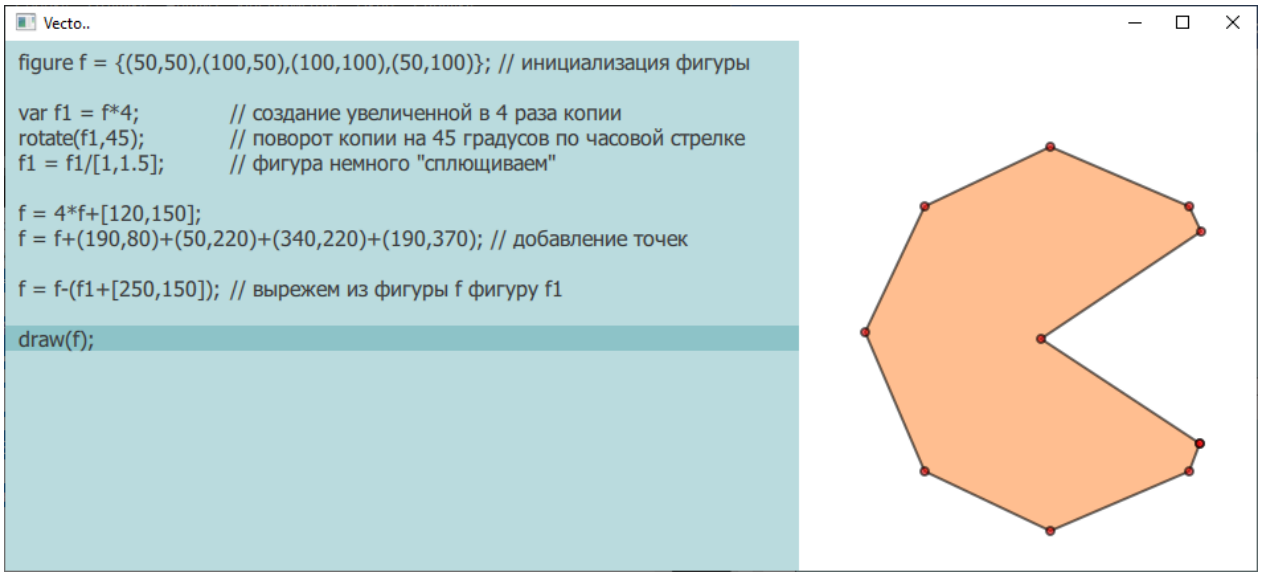

Рисунок 1. Пример 1

Пример 2 - «радостное солнышко» (рисунок 2). В начале строится квадрат. Он увеличивается в 4 раза и смещается на заданный вектор. Потом к квадрату добавляются еще 4 точки, получая восьмиугольник. Далее эта фигура объединяется с повернутой на 20 градусов копией, образуя «шероховатый» круг. После этого отдельно строятся «глаза» к этому кругу, представляющие собой треугольники, и «рот». Далее назначаются параметры отображения ранее полученных фигур. Конечный результат визуализируется на экран.

Рисунок 2. Пример 2

В предлагаемом проблемно-ориентированном языке выполняется диагностика до первой ошибки. Сообщение об ошибке включает указание места возникновения этой ошибки и перечисление возможных вариантов. Для вычислительной части предусмотрено дерево разбора (рисунок 3).

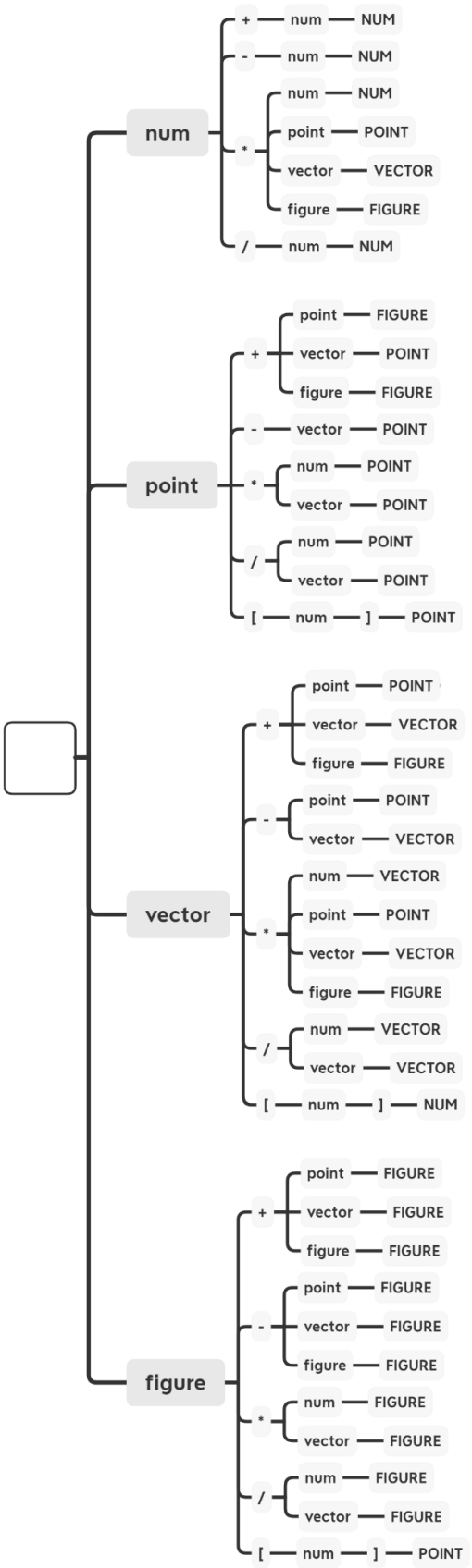

Рисунок 3. Дерево разбора (1й слой – тип левого операнда, 2й – операция, 3й – тип правого операнда, 4й – ожидаемый тип результата)

Примеры диагностики:

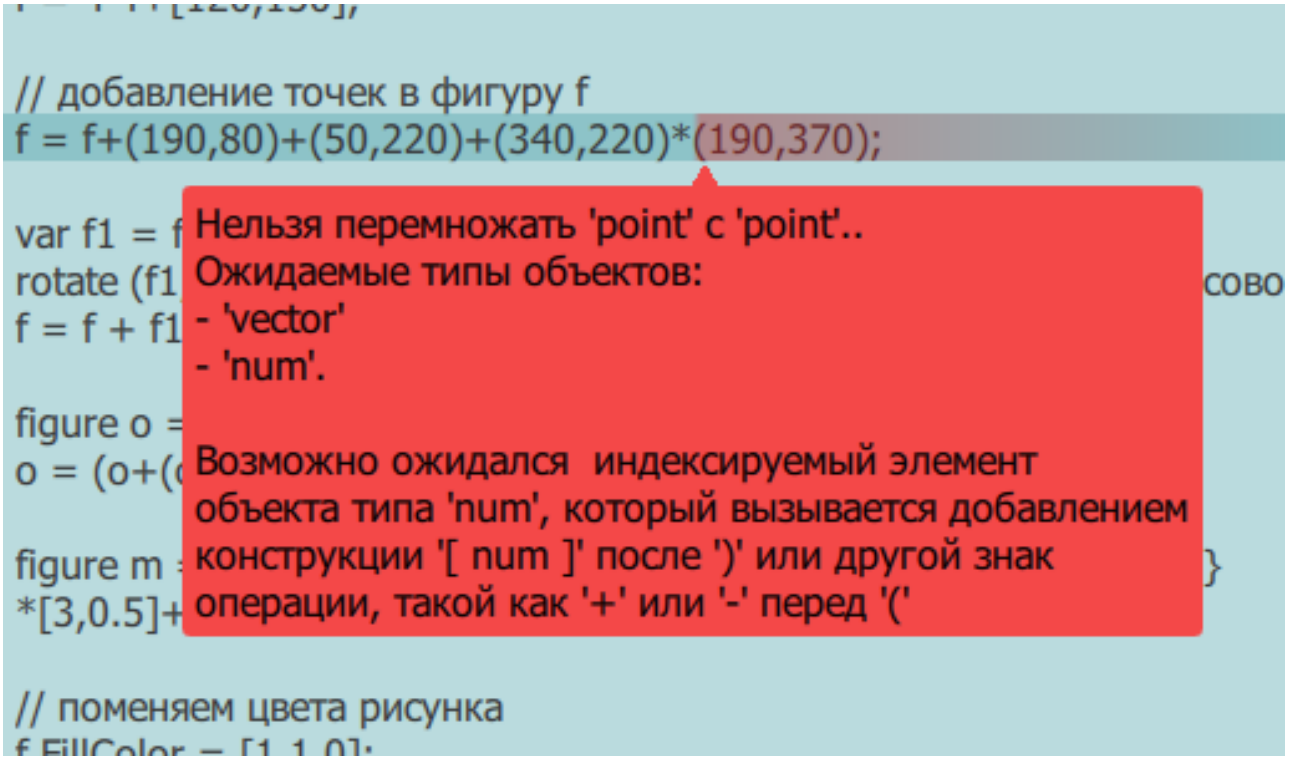

Рисунок 4. Пример диагностики 1

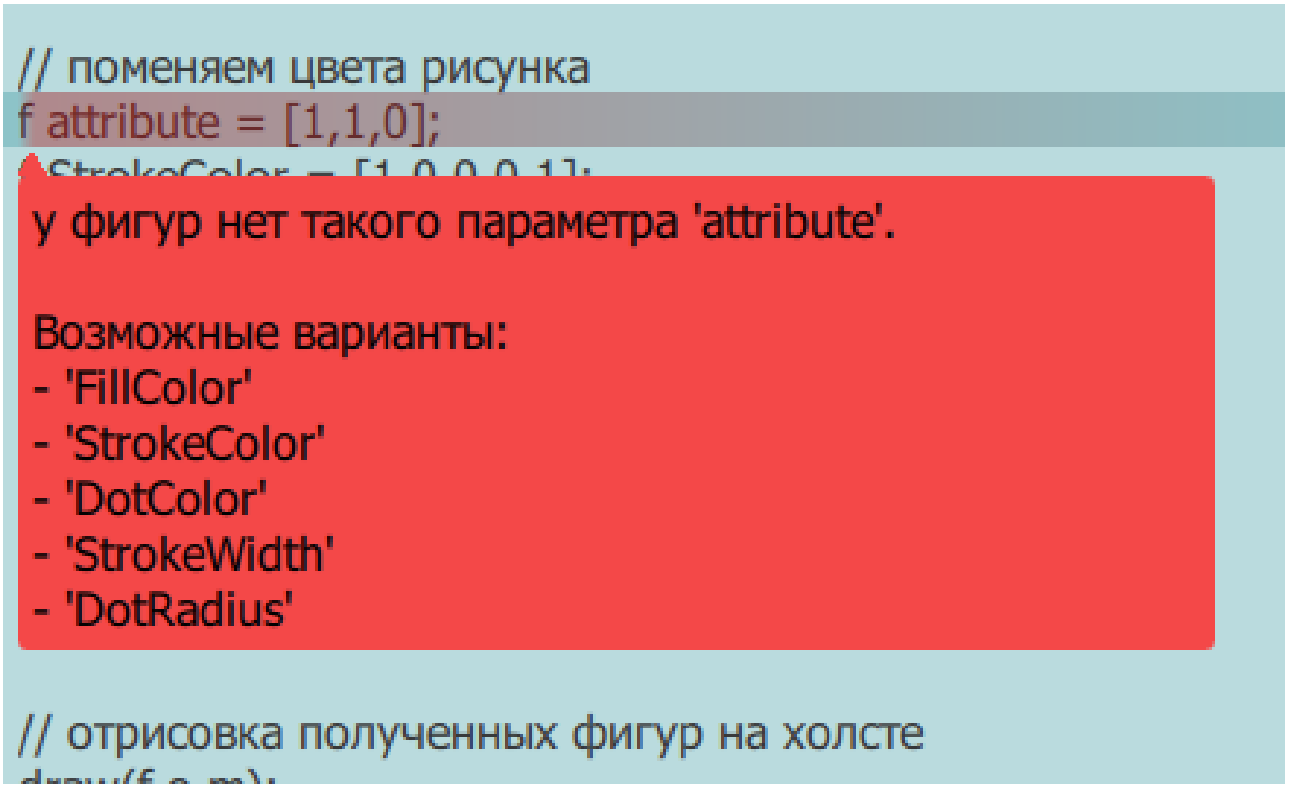

Рисунок 5. Пример диагностики 2

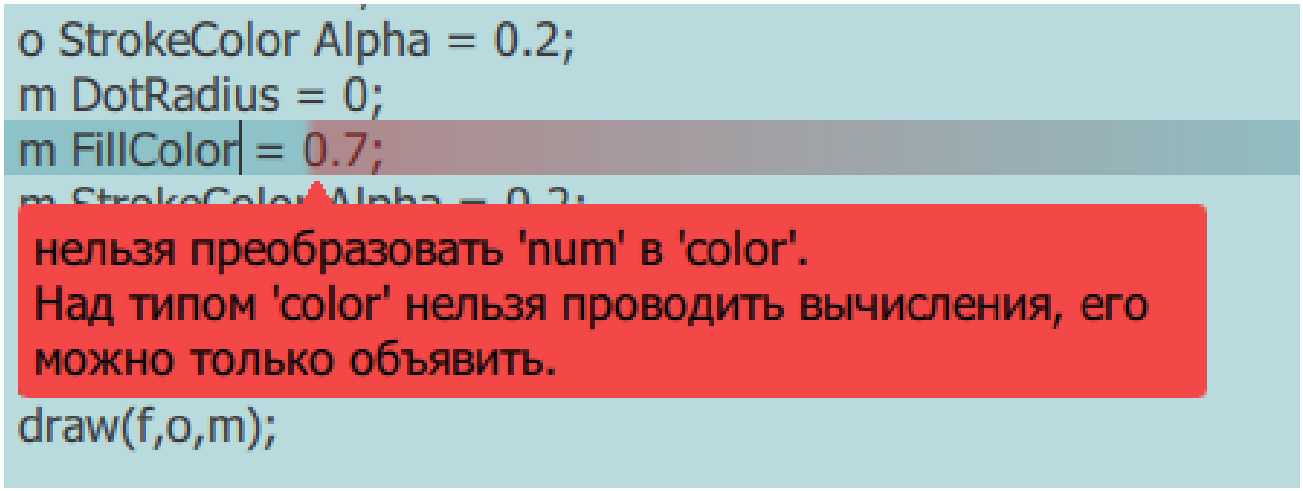

Рисунок 6. Пример диагностики 3

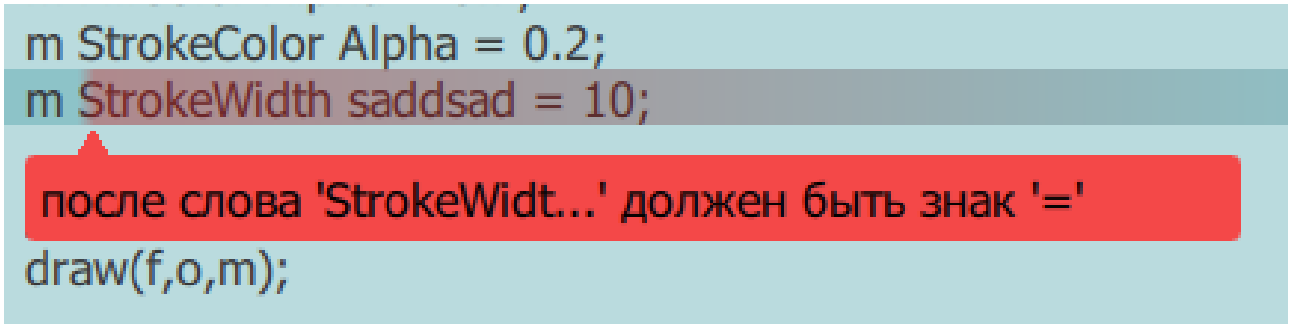

Рисунок 7. Пример диагностики 4

Выводы. В результате проведенных исследований были разработаны проблемно-ориентированный язык программирования и специализированное интерпретирующее средство, соответствующее этому языку. В нем реализованы базовые функции, способные сформировать изображение. В дальнейшем в этот язык можно добавлять новый функционал, а также провести некоторые оптимизационные работы.

Библиографический список:

1. P. Hudak. Modular Domain Specific Languages and Tools // Proceedings. Fifth International Conference on Software Reuse (Victoria, BC, Canada, Canada, 5-5 June 1998) – IEEE, 1998. – pp. 134-142.

2. W. Taha. Domain-Specific Languages // Proceedings. IFIP TC 2 Working Conference, DSL 2009 (Oxford, UK, July 15-17, 2009) – Berlin, Germany, 2009. - 411 p.

3. gnuplot [Электронный ресурс]. URL: http://www.gnuplot.info/.

4. Vector Markup Language (VML) [Электронный ресурс]. 1998 г. URL: https://www.w3.org/TR/NOTE-VML

5. М. Фаулер. Проблемно-ориентированные языки – Addison-Wesley Professional, 2010. – 640 стр.

6. Кариев Ч. А. Курс: Масштабируемая векторная графика [Электронный ресурс] // ИНТУИТ. 2007 г. URL: https://intuit.ru/studies/courses/1063/210/info.

7. Проект: интерпретатор языка PostScript для 2D графики [Электронный ресурс]. URL: https://ps-group.github.io/compilers/project_postscript.

8. Моисеев А. POSTSCRIPT УМЕР ДА ЗДРАВСТВУЕТ PDF?! [Электронный ресурс] // журнал «Publish». 2001 г. URL: https://www.publish.ru/articles/200102_4043043.

9. SVG — учебное руководство [Электронный ресурс]. URL: https://developer.mozilla.org/ru/docs/Web/SVG/Tutorial.