

Дмитрий Усенков

**Занимательные задачи
по программированию
обработки растровой графики**



ЛитРес: Самиздат
Москва

УДК 087
ББК 74
У74

Усенков Д. Ю.

Занимательные задачи по программированию обработки
растровой графики / Усенков Д. Ю. — Москва: ЛитРес:
Самиздат, 2020. — 78 с.

ISBN 978-5-532-03190-6

Пособие позволяет на практике познакомиться с основными алгоритмами компьютерной обработки растровой графики, выполняя задания по программированию на языке Visual Basic или Pascal версии ABC.Net. Материал может быть использован для внеурочной учебной деятельности, дополнительного образования, при выполнении учебных проектов и пр. Для школьников, учителей информатики, преподавателей дополнительного образования и для всех, интересующихся принципами работы компьютера с графическими данными.

ISBN 978-5-532-03190-6

© Усенков Д. Ю., текст, 2020
© Omiko Trading LTD., оформление, 2020

Содержание

Введение	5
Visual Basic как лабораторная среда	7
Создание экранной формы	9
Графические возможности Visual Basic	14
PascalABC.Net как лабораторная среда	15
Особенности работы с растровой графикой в PascalABC.Net	18
Пишем первую программу (Visual Basic)	21
Пишем первую программу (PascalABC.Net)	23
Копирование растрового изображения	26
Задача 1.	26
Решение (Visual Basic).	26
Решение (PascalABC.Net).	27
Задача 2.	31
Решение (Visual Basic)	31
Решение (PascalABC.Net)	33
Изменение яркости	35
Задача 3.	35
Решение (Visual Basic)	35
Решение (PascalABC.Net)	37
Изменение цветовой гаммы	39
Задача 4.	39
Решение (Visual Basic)	39
Решение (PascalABC.Net)	40
Изменение контраста	42
Задача 5.	42
Решение (Visual Basic)	42
Решение (PascalABC.Net)	44
Инверсия цветов (негатив)	46
Задача 6.	46
Решение (Visual Basic)	46
Решение (PascalABC.Net)	47

Преобразование цветного рисунка в монохромный	49
Задача 7.	49
Решение (Visual Basic)	49
Решение (PascalABC.Net)	50
Задача 8.	52
Решение (Visual Basic)	52
Решение (PascalABC.Net)	53
Выделение цветового канала	55
Задача 9.	55
Решение (Visual Basic)	55
Решение (PascalABC.Net)	58
Выделение контуров изображения	60
Задача 10.	60
Решение (Visual Basic)	60
Решение (PascalABC.Net)	66
Динамическое преобразование рисунка	73
Задача 11.	73
Объяснение решения.	73
Решение (Visual Basic)	75
Решение (PascalABC.Net)	76

Введение

Такая тема, как машинная графика и ее применение, весьма популярна и среди преподавателей информатики, и среди школьников. Однако при программировании задания чаще всего ограничиваются лишь рисованием на экране различных геометрических фигур или составленных из них чуть более сложных картинок. При этом остается в стороне другой, также очень существенный и интересный «пласт» алгоритмов, касающихся графических преобразований растровых изображений – тех, которые мы привыкли выполнять средствами того или иного графического редактора, не задумываясь об их сущности. Это изменение яркости и контраста, преобразование цветных изображений в монохромные, графическая фильтрация и т. д.

Ниже приводится ряд занимательных задач по реализации некоторых наиболее простых графических преобразований, которые помогут понять смысл обычно скрытых от пользователя процессов, происходящих «в недрах» графического редактора при выполнении той или иной команды обработки изображений. Различные вариации этих задач могут быть предложены учащимся для самостоятельного решения на факультативах и олимпиадах по программированию.

Решение таких задач можно рассматривать не только как иллюстрацию принципов работы соответствующих графических преобразований или как задания по программированию, но и как учебный проект по реализации «собственной версии» графического редактора (в том числе коллективный, когда каждый исполнитель проекта реализует какое-то одно графическое преобразование, сложность которого соответствует его уровню подготовки).

При решении предлагаемых в данной брошюре задач (при написании, исполнении и отладке соответствующих программ) предлагается использовать среду визуального программирования Visual Basic (VB) либо транслятор с языка Паскаль – PascalABC.Net. Каждое из этих двух инструментальных средств обладает своими пре-

имуществами и особенностями при работе с растровой графикой, которые будут рассмотрены далее.

Visual Basic как лабораторная среда

Выбор среды визуального программирования Visual Basic (VB) в качестве лабораторной среды обусловлен следующими соображениями.

Во-первых, язык Бейсик, на котором реализуются программные модули в VB, всем хорошо знаком и понятен, а сама оболочка Visual Basic достаточно популярна. (Для написания предлагаемых читателям листингов использована версия Visual Basic 6.0, которая в свое время распространялась по школам, но эти программы могут быть переписаны на новые версии VB или на другом языке программирования, в котором реализован функционал для работы с растровой графикой.)

Во-вторых, концепция *визуального программирования* позволяет легко и быстро «нарисовать» на экране интерфейс простейшего программного приложения, требуемого для испытания разрабатываемых алгоритмов (экранную форму, содержащую исходное изображение, поле для вывода результата и «пусковую» кнопку), сосредоточившись именно на самих алгоритмах графических преобразований.

Наконец, в третьих, Visual Basic предоставляет удобное средство загрузки на экранную форму исходного изображения из любого внешнего файла (например, формата BMP), не требуя разработки соответствующей подпрограммы и обращения к библиотекам работы с растровой графикой.

Методическая ценность использования Visual Basic в качестве лабораторной среды при изучении сложных алгоритмов также существенно повышается благодаря наличию в VB широкого набора встроенных средств отладки программ, что позволяет легко организовать отслеживание работы алгоритмов – в частности, контроль значений выбранных переменных. При этом режим отладки позволяет выполнять программу «по шагам» (по одной строке листинга при каждом нажатии клавиши **F8**), просматривая в отдельном окне

выведенные системой отладки текущие значения отслеживаемых переменных.

Напомним, что управление средствами отладки сосредоточено в меню **Debug (Отладка)**:

- пункт **Add Watch (Добавить Наблюдателя)** позволяет добавить любую из имеющихся в листинге переменных в список отслеживания значений (можно предварительно выделить в листинге имя требуемой переменной, тогда оно автоматически будет занесено в соответствующую графу окна добавления переменной, даже если выделенное имя не было скопировано в буфер обмена);

- пункт **Edit Watch (Правка Наблюдателя)** раскрывает окно управления списком отслеживаемых переменных, где можно добавлять новые переменные или удалять существующие, ставшие ненужными;

- пункт **Toggle Breakpoint (Включить Финиш-Точки)** либо горячая клавиша **F9** позволяют создавать (или, при повторном вызове, снимать) в листинге контрольные точки останова. При вызове этого пункта или нажатии клавиши **F9** текущая строка листинга, на которой находился текстовый курсор, выделяется бордовой фоновой подсветкой, а запущенная на выполнение программа, дойдя до отмеченной строки, приостанавливается, и Visual Basic переходит в режим отладки.

Запуск написанных программ и просмотр получаемых результатов мы будем производить средствами оболочки VB, хотя ничто не запрещает при желании и оттранслировать результаты работы в виде полноценного Windows-приложения.

Создание экранной формы

После запуска среды Visual Basic на экране появляется ряд отдельных окон и панелей (рис. 1), среди которых для нас важными являются:

- основная панель сверху экрана (главное меню и строка кнопок);
- панель инструментов (слева), содержащая кнопки для добавления на создаваемую экранную форму тех или иных объектов интерфейса;
- сама экранная форма – при открытии VB создается по умолчанию, для добавления новой экранной формы нужно воспользоваться кнопкой



– окно проекта (справа сверху), содержащее перечень созданных экранных форм и стандартных библиотек (если созданная экранная форма отсутствует на экране, ее окно можно раскрыть двойным щелчком мыши на соответствующей строке окна проекта);

– окно свойств объекта – раскрывается автоматически; при закрытии может быть повторно раскрыто с помощью кнопки



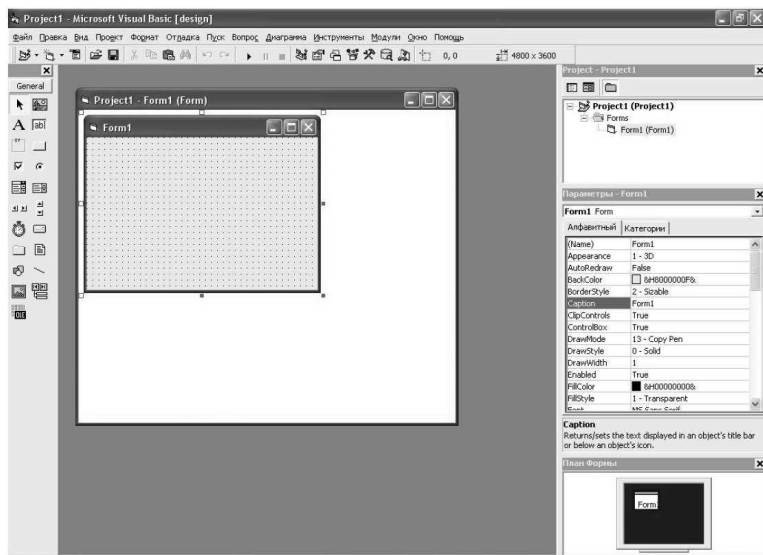


Рис. 1

Подготовим простейшую экранную форму, с помощью которой будет производиться проверочный запуск программ, созданных в качестве решения задач.

1. Перетаскивая мышью углы и границы окна, увеличим размеры предлагаемой экранной формы до требуемых.

2. Выбирая в панели инструментов кнопку



разместим («нарисуем») на экранной форме два объекта «поле рисунка» (**Picture Box**), расположив их рядом по горизонтали.

3. Выбрав в панели инструментов кнопку

