

## РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ФОРМИРОВАНИЯ БЕЛОРУССКИХ ОРНАМЕНТОВ В ДОПЕЧАТНЫХ ПРОЦЕССАХ ПОЛИГРАФИИ

In article the description of the software for creation of digital images of the Byelorussian ornaments is resulted. The choice of the program environment for the decision of a task in view is proved. The assessment of works of the program is given and the basic directions of its development are outlined.

В настоящее время наблюдается дефицит программного обеспечения для автоматизированного создания белорусских орнаментов, несмотря на актуальность этой задачи в полиграфии. Использование существующих программных средств компьютерной графики для разработки цифровых орнаментальных изображений не очень эффективно, так как сопряжено с большой долей ручного труда. Вместе с тем полная разработка специализированной прикладной программы связана с некоторыми недостатками: 1) необходимостью привлечения ряда узких специалистов в области программирования и полиграфии; 2) значительными временными затратами. Наиболее приемлемым вариантом автоматизации процесса создания белорусских орнаментов представляется расширение функциональных возможностей действующих программ векторной графики за счет встроенных средств автоматизации работы. По результатам аналитического обзора современных средств машинной графики [1] можно сделать вывод, что наиболее целесообразно для решения поставленной задачи использовать программу Corel Draw. Она адаптирована для использования в допечатных процессах полиграфического производства, позволяет создавать дополнительные программные модули на языке Visual Basic for Application, располагает развитыми средствами ввода и отладки программного кода, а также предоставляет широкие возможности по созданию и модификации элементов пользовательского интерфейса.

В соответствии с вышесказанным, программная реализация принципов синтеза белорусских орнаментов, разработанных в предыдущих работах [2, 3], была выполнена в среде Corel Draw 11. Программный код при этом написан на языке Visual Basic for Application 6.3.

Процедуры, входящие в созданный программный модуль, позволяют реализовать симметрические преобразования базового изобразительного элемента, а также создавать фрагменты орнаментальных розеток и сами розеточные узоры различного уровня сложности.

Для выполнения симметрических преобразований служат следующие процедуры:

1. *Repenos* — дублирование изобразительного элемента путем параллельного переноса в

горизонтальном, вертикальном или диагональном направлении заданное число раз.

2. *Flip* — отражение дубликата исходного узора относительно горизонтальной либо вертикальной линии симметрии, смещенной от центра фигуры на заданное расстояние.

3. *Rotate* — 3-кратное дублирование базового объекта путем поворота вокруг оси симметрии 4-го порядка, перпендикулярной плоскости изображения. Возможны два варианта расположения поворотной оси симметрии относительно рисунка. В первом случае ось смещается от базовой точки в горизонтальном направлении. Базовая точка при этом находится в середине дискретного элемента, являющегося крайним элементом исходного объекта справа и центральным по вертикали. Во втором случае ось вращения смещается по диагонали, а базовая точка находится в центре дискретного элемента, расположенного в правом нижнем углу векторного изображения.

4. *Flip\_sliding* — многократное дублирование исходного фрагмента узора путем скользящего отражения относительно горизонтальной или вертикальной линии, смещенной от центра на заданное расстояние. Под скользящим отражением в теории симметрии [4] понимается последовательное выполнение двух преобразований: параллельного переноса и отражения.











В то же время в случае базовых операций трансформирования векторных объектов не требуется написания отдельных процедур, поскольку для этих целей имеются эквивалентные операторы языка VBA. Так, при необходимости осуществить однократное дублирование с изменением положения дубликата применяется оператор *Duplicate*. Параллельный перенос векторного объекта реализуется посредством оператора *Move*.

На основе дискретного элемента орнамента (квадрата), генерируемого подпрограммой *Fig*, в результате выполнения в заданной последовательности перечисленных преобразований возможен синтез розеточных орнаментов и их фрагментов.

Для этих целей используются следующие процедуры:

1. *Line* — создание диагонального зубчатого отрезка заданной длины и толщины.

Параметры орнаментальных фигур, генерируемых процедурами

Процедуры	Пример фигуры	Параметры фигуры
Kruki		$dx$ — толщина линии в горизонтальном разрезе; $l_1$ — длина первого линейного фрагмента крючкообразного элемента; $l_2$ — длина второго фрагмента; $rad$ — расстояние от угловых узоров до центра симметрии; $four_p$ — количество пар крючкообразных элементов, равное 4 или 2 (для примера фигуры $dx=1, l_1=6, l_2=4, rad=9, four_p=4$ )
Gor_resh		$nx, ny$ — количество переносов по 2-м направлениям; $ann$ — активизация наложения с вычитанием; $x, y$ — периоды переноса по 2-м направлениям (для примера фигуры $nx=15, ny=5, ann=false, x=1, y=2$ )
Romb		$l$ — длина стороны, выраженная в количестве зубцов по внешнему краю; $dx$ — толщина линии в горизонтальном разрезе (для примера фигуры $l=12, dx=2$ )
Resheto		$l, dx$ — параметры линии; $per$ — периодичность элементов решетки; $n\_line$ — количество линий, образующих решетку (для примера фигуры $l=15, dx=3, per=3, n\_line=4$ )
Romb_skos 2		$h$ — высота 8-угольника; $dh$ — размер отсекаемой от прямоугольника верхней области; $d$ — толщина стороны 8-угольника в горизонтальном разрезе (для примера фигуры $h=25, dh=5, d=3$ )
Zvezd_k1		$dy$ — размер стороны окончания по вертикали; $n\_per$ — количество элементов штриховки; $rad$ — расстояние от края узора до центра симметрии (выражается количеством дискретных элементов); $per$ — расстояние между соседними элементами штриховки (периодичность) (для примера фигуры $dy=4, n\_per=5, rad=3, per=2$ )
Zvezd_k2		$dx, dy$ — размеры стороны окончания по горизонтали и вертикали; $rad$ — расстояние от края узора до центра симметрии (выражается количеством дискретных элементов); $complex$ — тип создаваемого узора (простой/сложный) (для примера фигуры $dx=4, dy=6, rad=8, complex=true$ )
Vetki		$per$ — периодичность элементов «ветки»; $n\_per$ — кол-во элементов в «ветке»; $rad$ — расстояние от «ветки» до центра симметрии; $d\_line$ — наличие диагональных отрезков, проходящих по центру «веткою» (для примера фигуры $per=3, n\_per=4, rad=1, d\_line=true$ )
Romb_resh1		$l, dx$ — параметры линии решетки; $per$ — периодичность элементов решетки; $n\_line$ — количество линий, образующих решетку; $lr, dxr$ — параметры квадрата (для примера фигуры $l=17, dx=3, per=3, n\_line=4, lr=11, dxr=3$ )
Romb_resh2		$dx, per, n\_line$ — параметры решетки; $lr, dxr$ — параметры квадрата; $l\_vn$ — размер отсекаемой области внутри фигуры (для примера фигуры $dx=3, per=3, n\_line=3, lr=18, dxr=3, l\_vn=9$ )
Okantovka		$dx$ — толщина линии окантовки; $per$ — периодичность элементов окантовки; $n\_line$ — количество линий, образующих окантовочную решетку; $dxr$ — параметр толщины внешней линии (для примера фигуры $dx=1, per=2, n\_line=5, dxr=2$ )

2. Kruk — создание крючкообразного зубчатого элемента из двух отрезков заданной длины и толщины.

3. Kruki — построение системы из двух или четырех пар крючкообразных элементов.

4. Gor\_resh — синтез узора, состоящего из пересекающихся горизонтальных и вертикальных отрезков.

5. Quadr — создание квадрата, лежащего на основании.

6. Romb — генерирование фигуры в виде квадрата, опирающегося на вершину.

5. Resheto — формирование розетки в виде пересекающихся зубчатых диагональных отрезков, расположенных с заданной периодичностью.

6. Romb\_skos — генерирование зубчатого восьмиугольника со сплошной заливкой.

7. Romb\_skos 2 — построение зубчатого восьмиугольника со сторонами заданной толщины.

8. Zvezd\_k1 — формирование звездообразных окончаний, расположенных вдоль горизонтали.

9. Zvezd\_k2 — построение звездообразных окончаний, расположенных вдоль диагонали.

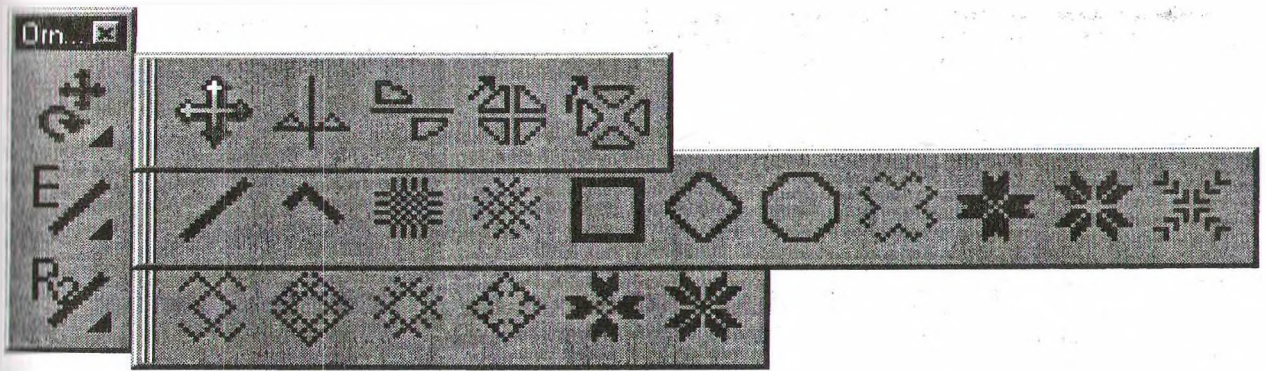


Рис. 1. Панель инструментов для синтеза белорусских орнаментов

10. *Vetki* — генерирование узора в виде стилизованного изображения веток.

11. *Kruki\_12* — синтез изображения в форме квадрата, поставленного на вершину, в сочетании с системой крючкообразных элементов в углах.

12. *Romb\_resh1* — создание узора в форме квадрата, поставленного на вершину, в сочетании с системой диагональных отрезков по внешнему краю.

13. *Romb\_resh2* — образование фигуры в форме квадрата, поставленного на вершину, в сочетании с системой диагональных отрезков по внутреннему краю.

14. *Okantovka* — построение орнаментальной розетки в виде поставленного на вершину квадрата с внутренней окантовкой.

15. *Zvezd\_12\_k1* — формирование узора в форме квадрата, который поставлен на вершину, в сочетании с системой звездообразных окончаний, расположенных вдоль горизонтали.

16. *Zvezd\_12\_k2* — генерирование розетки в форме квадрата, который поставлен на вершину, в сочетании с системой звездообразных окончаний, расположенных вдоль диагонали.

Подпрограммы, позволяющие получать орнаментальные изображения более высокого уровня сложности, базируются на ранее написанных процедурах. Например, процедуры *Romb*, *Resheto* используют процедуру *Line*; *Romb\_resh1*, *Romb\_resh2*, *Okantovka* — *Romb*, *Resheto*.

Различные режимы наложения двух векторных объектов, требуемые для получения некоторых орнаментальных розеток, реализуются посредством операторов *Weld* (сложение), *Combine* (наложение с удалением пересекающихся частей), *Trim* (вычитание одной фигуры из другой). При необходимости сложить несколько векторных изображений используется подпрограмма *Union*.

В таблице приведены параметры отдельных процедур, реализующих синтез орнаментальных розеток, а также примеры узоров, которые получены в результате работы этих подпрограмм.

Применяя к созданным орнаментальным розеткам процедуры, отвечающие за параллельный перенос и скользящее отражение, можно получить бордюрные и сетчатые орнаменты.

Для организации взаимодействия с пользователем используются такие элементы интерфейса, как панель инструментов, формы и окна с сообщениями.

Окна с сообщениями информируют пользователя о некорректности введенных параметров или о необходимости предварительно совершить определенные действия. Например, при попытке запуска процедуры, отвечающей за выполнение того или иного симметрического преобразования, в случае отсутствия выделенных фигур на экран выводится сообщение о необходимости предварительно выбрать объект трансформирования.

Форма представляет собой диалоговое окно, дающее возможность ввода и редактирования параметров операции или создаваемого изображения. В последнем случае (рис. 2) форма дополнительно оснащена кнопками изменения положения синтезированного объекта на странице (для повышения удобства сопоставления нескольких мотивов), а также кнопкой удаления активного узора в случае получения не устраивающих пользователя результатов. Наличие перечисленных кнопок вызвано невозможностью изменять вручную содержимое страницы до закрытия формы. Для заполнения формы значениями параметров используются такие элементы, как *TextBox* (текстовое поле) и *ComboBox* (комбинированный список). Последний применяется в случае, когда осуществляется выбор из ограниченного количества возможных вариантов. В ситуации, когда пользователь по каким-то причинам хочет вернуться к исходному варианту формы, может быть использована кнопка сброса введенных данных.

Программный код каждой формы содержит обращение к соответствующей процедуре, которая автоматически запускается при нажатии кнопки «ОК» или «Синтез».

На рис. 2 приведена форма для ввода параметров создания розеточного орнамента, содержащего элементы звезды. Как видно из рисунка, для выбора типа звездообразного узора, а также указания особенностей заполнения его

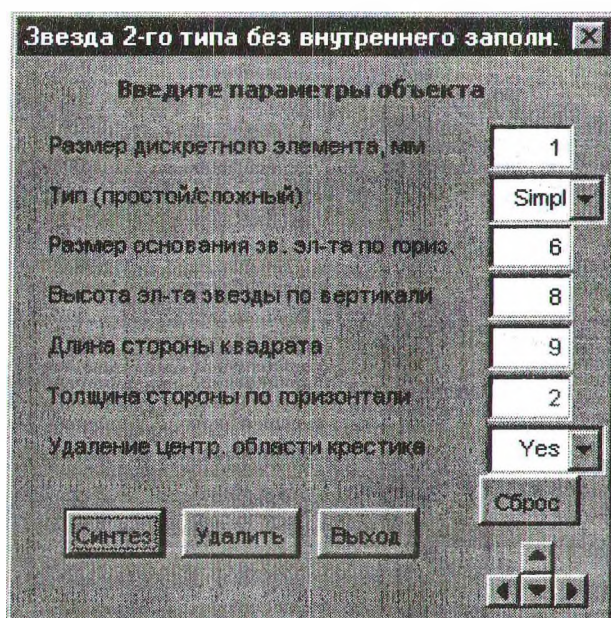


Рис. 2. Пример формы для создания розетки

центральной области используются раскрывающиеся списки. Это обусловлено тем, что количество возможных значений каждого из названных параметров ограничено двумя. Кроме того, часть параметров, свойственных только альтернативному типу розетки, является недоступной для редактирования в данный момент.

Для обеспечения вывода на экран необходимой формы средствами Corel Draw создана пользовательская панель инструментов (рис. 1), включающая три группы команд, каждая из которых содержит ряд кнопок. Кнопки первой группы предоставляют доступ к операциям трансформирования, второй и третьей групп — к синтезу розеточных элементов и собственно орнаментальных розеток. При этом последний из трех упомянутых элементов панели инструментов обеспечивает генерирование более сложных по составу розеточных орнаментов, чем второй. Для всех кнопок пользовательской панели созданы соответствующие их функциональному назначению рисунки и всплывающие подсказки. При нажатии на кнопку запускается

связанная с ней подпрограмма, которая выводит на экран нужную форму.

Описанное программное обеспечение позволяет в значительной степени автоматизировать процесс создания цифровых изображений белорусских орнаментов, обеспечивая вместе с тем их технологичность (высокую графическую и цветовую точность, малый объем графических файлов векторных форматов). Интеграция в распространенный графический пакет Corel Draw и сохранение преемственности при разработке интерфейса создают предпосылки для освоения созданной программы в сравнительно короткий срок. В то же время создание сложных по составу и симметрии орнаментальных изображений предполагает последовательный запуск нескольких процедур, т. е. полностью не избавляет пользователя от ручных операций.

В целом результаты работы созданного программного обеспечения свидетельствуют об эффективности разработанных ранее принципов синтеза белорусских орнаментов и целесообразности использования среды Corel Draw для его программной реализации. Дальнейшие направления развития программы видятся в сокращении доли ручных операций и совершенствовании пользовательского интерфейса.

## Литература

1. Сипайло С. В. Средства и методы автоматизированного проектирования цифровых изображений белорусских орнаментов // Труды БГТУ. Сер. IX. Издат. дело и полиграфия. — 2004. — Вып. XII. — С. 37—40.
2. Сипайло С. В. Разработка принципа синтеза белорусских орнаментов // Издательско-полиграфический комплекс на пороге третьего тысячелетия: Материалы Междунар. науч.-техн. конф. / БГТУ. — Мн., 2001. — С. 133—138.
3. Долгова Т. А., Сипайло С. В. Автоматизированное проектирование белорусских орнаментов на основе теории групп // Труды БГТУ. Сер. VI. Физико-математические науки и информатика. — 2005. — Вып. XIII. — С. 27—29.
4. Шубников А. В., Копчик В. А. Симметрия в науке и искусстве. — М.: Наука, 1972. — 340 с.