

Е. А. СИДОРОВА, И. Л. САЛЯ

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ ГРАФИКИ

ОМСК 2018

Министерство транспорта Российской Федерации
Федеральное агентство железнодорожного транспорта
Омский государственный университет путей сообщения

Е. А. Сидорова, И. Л. Саля

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ ГРАФИКИ

Утверждено методическим советом университета
в качестве учебно-методического пособия к выполнению
самостоятельной работы

Омск 2018

УДК 004.92 (075.8)
ББК 32.972.131.2я73
С34

Теоретические основы компьютерной графики: Учебно-методическое пособие к выполнению самостоятельной работы / Е. А. Сидорова, И. Л. Саля; Омский гос. ун-т путей сообщения. Омск, 2018. 28 с.

Приведены теоретические сведения о компьютерной графике, ее классификация и характеристики. Описаны принципы кодирования графической информации и методика расчета информационного объема растровых и векторных изображений. Приведены основные форматы файлов, применяемых для хранения графической информации, и практические задания.

Предназначено для студентов первого курса очной и заочной форм обучения всех направлений подготовки (специальностей), изучающих дисциплины информационного профиля: «Информатика», «Информационные технологии», «Компьютерные технологии и информатика», «Информационные системы и базы данных» и др., а также может быть использовано в качестве самоучителя для любых категорий пользователей персонального компьютера.

Библиогр.: 3 назв. Табл. 6. Рис. 13.

Рецензенты: доктор техн. наук, профессор В. Н. Горюнов;
доктор пед. наук, канд. техн. наук, доцент О. Ф. Пиралова.

ОГЛАВЛЕНИЕ

Введение	5
1. Понятие и классификация компьютерной графики	6
1.1. Растровая графика	6
1.1.1. Преимущества и недостатки растровой графики	8
1.2. Векторная графика	9
1.2.1. Преимущества и недостатки векторной графики	11
1.3. Фрактальная графика	12
1.4. Трехмерная графика	12
2. Кодирование графической информации	14
2.1. Разрешение изображения, экрана и принтера	14
2.2. Цветовое разрешение	14
2.3. Цветовые модели	15
2.4. Информационный объем изображения	17
2.4.1. Информационный объем растрового изображения	18
2.4.2. Информационный объем векторного изображения	19
3. Контрольные вопросы	22
4. Практические задания	23
Задание 1. Растровая графика (РГ)	23
Задание 2. Векторная графика (ВГ)	26
Библиографический список	27

ВВЕДЕНИЕ

Компьютерная графика представляет собой современную технологию создания различных изображений с помощью аппаратных и программных средств компьютера, отображения их на экране монитора и затем сохранения в файле или печати на принтере. Изображения бывают рисованными или фото-реалистичными, плоскими или трехмерными, статичными или анимированными.

Компьютерная графика применяется в следующих областях:

издательском деле для производства рекламных плакатов, визитных карточек, иллюстраций в книгах и журналах (специализированное программное обеспечение *Adobe Photoshop*, *Adobe Illustrator*, *PageMaker*, *Corel Draw* и др.);

системах автоматического проектирования для создания и редактирования чертежей (*AutoCAD*, *T-Flex CAD*, *Solid Works*, *КОМПАС-3D*, *Visio* и др.);

кино- и шоу-бизнесе для создания теле- и видеофильмов, телепередач, мультфильмов (*Autodesk 3ds Max*, *Maya* и др.);

компьютерных играх и обучающих системах, например, в симуляторах авиapolетов (*Visual C++*, *DirectX*, *OpenGL* и др.);

моделировании физических, химических и других процессов для наглядного представления результатов моделирования (*MatLab*, *MathCAD* и др.);

сети Интернет для размещения иллюстраций, кнопок, баннеров и анимации на веб-сайтах (*Adobe Dreamweaver*, *WYSIWYG Web Builder* и др.).

Современным направлением развития компьютерной графики является трехмерная графика, изучающая приемы построения объемных моделей объектов в виртуальном трехмерном пространстве.

Редактирование графики требует множества математических вычислений и обработки больших объемов информации, поэтому для работы с ней необходимы мощные компьютеры. Компьютерная графика является одной из наиболее быстро развивающихся областей информатики и выступает одним из двигателей прогресса всей компьютерной индустрии.

Библиографический список, представленный в конце данного издания, содержит литературу для углубленного изучения материала по рассматриваемой тематике.

1. ПОНЯТИЕ И КЛАССИФИКАЦИЯ КОМПЬЮТЕРНОЙ ГРАФИКИ

В настоящее время визуализация данных находит широкое распространение в самых разных сферах человеческой деятельности: научных исследованиях, моделировании и проектировании, медицине, рекламе и дизайне, играх и обучении и др. В связи с этим широкое развитие получила **компьютерная графика** – область информатики, которая занимается созданием, вводом (оцифровкой), обработкой, хранением и выводом графической информации с использованием компьютера и специализированного программного обеспечения.

Компьютерная графика служит основой **анимации**, под которой понимается изменение вида, формы, размеров, расположения объектов на экране, создающее эффект движения.

Классификация компьютерной графики представлена на [рис. 1](#).



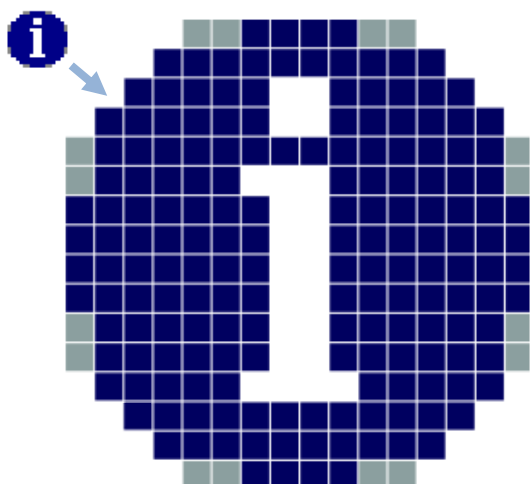
[Рис. 1](#). Виды компьютерной графики

1.1. Растровая графика

Растровое представление обычно используют для изображений фотографического типа с большим количеством деталей или оттенков. Растровые изображения редко создают вручную, чаще используют готовые рисунки или фотографии (с цифровой камеры, сканера и др.), которые затем обрабатывают с помощью графических редакторов растровых изображений. В сети Интернет применяют только растровые иллюстрации.

Основной элемент растрового изображения – **точка** (пиксель – от англ. *picture element*, т. е. элемент картины). **Пиксель** является наименьшим элементом растрового изображения, для которого могут быть заданы адрес, цвет и его интенсивность.

Растровое изображение напоминает мозаику и описывается конкретным расположением и цветом пикселей. Благодаря незначительному размеру точек они не видны или малозаметны, поэтому создается впечатление однородной картины. Пример растрового изображения приведен на [рис. 2](#).



[Рис. 2](#). Пример растрового изображения

При растривании изображение привязывается к прямоугольной сетке линий – **растру**. Каждая ячейка (**элемент растра**) отводится для одной **растровой точки**, представляющей собой совокупность микроточек (пикселей).

Частота сетки растра называется **линиатурой**, измеряется количеством линий на дюйм (*lines per inch – lpi*) или на сантиметр (см^{-1}) и обычно составляет 20 – 80 см^{-1} . Чем выше линиатура, тем мельче растровые точки, например, для растровых изображений принтеров линиатура составляет 12 – 24,

а для рекламных изданий она равна 80 – 120 см^{-1} .

Размер растровой точки рассчитывается для каждой ячейки растра и зависит от интенсивности тона в ней. Под **тоном** понимается оттенок цвета.

Интенсивность тона (яркость, оптическая плотность, светлота) в ячейке растра определяется процентом заполнения ячейки микроточками. Интенсивность тона подразделяют на 256 уровней (больше – не воспринимается зрением человека, меньше – ухудшается восприятие изображения). Для воспроизведения 256 уровней тона достаточно иметь размер ячейки растра $16 \times 16 = 256$ точек.

Размер растровой точки зависит также от примененного метода и параметров растривания оригинала изображения:

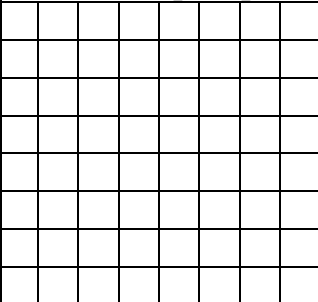
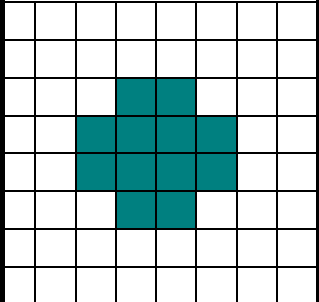
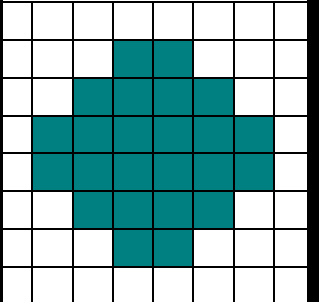
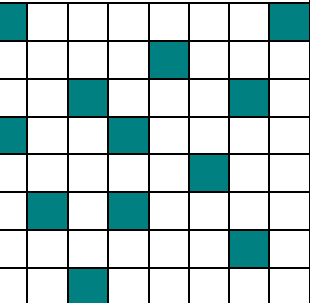
амплитудная модуляция (АМ) – метод растривания, при котором интенсивность тона в ячейках растра регулируется увеличением размера точек с соответствующим сокращением пробельного поля между ними при одинаковом расстоянии между центрами элементов растра;

частотная модуляция (ЧМ) – метод растривания, при котором интенсивность тона в ячейках растра регулируется расстоянием между соседними точками одинакового размера.

На [рис. 3](#) для сравнения приведены примеры амплитудной и частотной модуляций (ячейка растра имеет размер 8×8 точек).

Для обработки растровых изображений используются программные средства *Paint*, *Adobe Photoshop*, *GIMP* и др.

Расширения имен файлов растровой графики: *bmp*, *gif*, *tif*, *jpg*, *psx*, *png*, *heic* и др.

Ячейка растра	АМ 18,75 %	АМ 37,5 %	ЧМ 18,75 %
			

[Рис. 3](#). Пример использования амплитудной и частотной модуляций

1.1.1. Преимущества и недостатки растровой графики

Преимущества растровой графики:

возможность отображения мелких деталей, плавных переходов цвета, нечетких границ или размытых краев объектов;

высокое качество фотореалистичных изображений;

легкое изменение цветности, яркости, контрастности и других параметров любой части изображения;

простота оцифровки (ввода в компьютер) с помощью современных устройств: сканера, цифровой фото- или видеокамеры.

Недостатки растровой графики:

ухудшение качества при масштабировании. При увеличении изображения появляется эффект **пикселизации** (ступенчатости) ([рис. 4](#)), при уменьшении необратимо теряются некоторые детали изображения, так как несколько соседних пикселей сливаются в один;

редактирование растровой графики заключается в изменении цвета и яркости пикселей. Это удобно, когда необходимо изменить мелкие детали изображения или применить какой-либо визуальный эффект (например, эффект

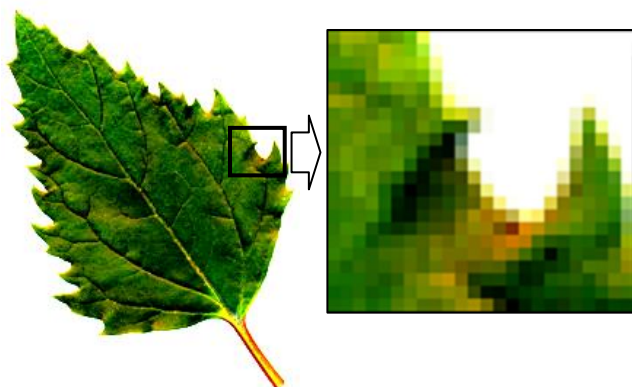


Рис. 4. Эффект пикселизации при увеличении растрового изображения

размытого изображения), однако изображение, представленное в растровом виде, не хранит информацию о форме объектов. Форма получается в сознании человека за счет разницы цветов соседних пикселей, поэтому редактировать форму объектов, представленных растровым способом, достаточно сложно;

в файлах растровой графики запоминается цвет каждого пикселя

на рисунке, поэтому такие файлы занимают, как правило, большой объем памяти. Одним из способов решения этой проблемы является **сжатие информации**, т. е. уменьшение размера файла растровой графики за счет изменения способа организации данных в нем.

1.2. Векторная графика

Векторная графика обычно используется не для создания художественных композиций, а для оформительских, чертежных и проектно-конструкторских работ. Элементарным объектом векторного изображения является **линия (контур)** (рис. 5).

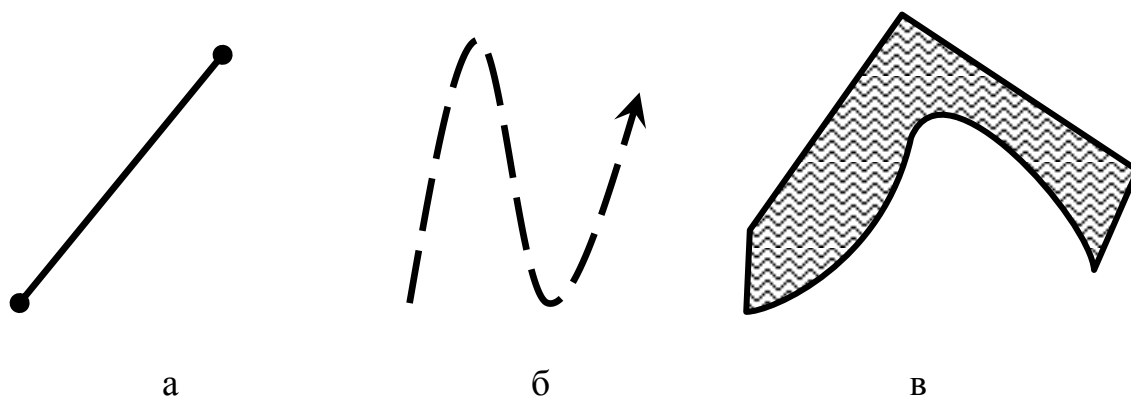


Рис. 5. Примеры элементов векторной графики

Каждый контур имеет две (или более) опорные точки, которые называются **узлами**. Элемент контура, заключенный между двумя смежными опорными точками, называют **сегментом** контура. Контур может быть открытым, когда последняя опорная точка не совпадает с первой, либо замкнутым, когда последний узел одновременно является и первым.

Как и все объекты, линии имеют определенные свойства: форму (прямая, кривая 2-го или 3-го порядка), толщину, цвет, начертание (сплошная, пунктирная) и др.

Замкнутые (в некоторых векторных редакторах и открытые) контуры имеют еще свойства заполнения (заливки), открытые контуры – свойства узлов, например, узлы могут быть точками, кружками, квадратиками, различными фигурами.

Линия описывается математической формулой, и потому объем данных для отображения объекта средствами векторной графики существенно меньше, чем в растровой графике. Простейшие объекты объединяются в более сложные, в результате изображение представляет собой совокупность простых элементов, называемых **графическими примитивами** (прямых линий, дуг, окружностей, эллипсов, прямоугольников и др.). Из-за такого подхода векторную графику часто называют объектно-ориентированной.

Положение и форма графических примитивов задаются в системе координат, связанных с виртуальным рабочим листом. Начало координат располагается в левом верхнем или правом нижнем углу этого листа.

Примеры фигур, полученных из графических примитивов, приведены на [рис. 6](#).

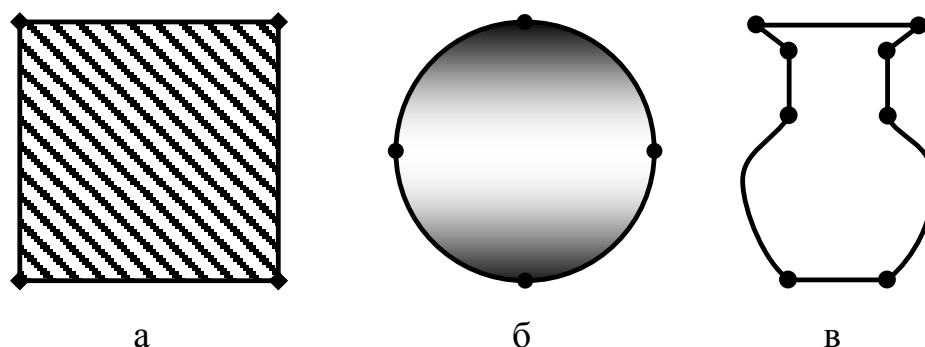


Рис. 6. Примеры фигур, полученных из графических примитивов

По сравнению с растровым векторное представление требует намного меньше памяти для хранения изображений, однако построение любого объекта

сопровождается непрерывным пересчетом его параметров, поэтому работа с векторной графикой требует более производительных вычислительных систем.

Для обработки векторных изображений используются программные средства *Microsoft Visio*, *Adobe Illustrator*, *CorelDraw*, *AutoCAD* и др.

Расширения имен файлов векторной графики: *wmf*, *ai*, *cdr*, *eps*, *fla*, *svg* и др.

1.2.1. Преимущества и недостатки векторной графики

Преимущества векторной графики:

отдельные объекты векторного изображения хорошо поддаются редактированию, не оказывая влияния на объекты, расположенные рядом на том же изображении;

при масштабировании и других трансформациях не возникает потери качества (рис. 7);

изображения, представленные в векторном виде, занимают намного меньший объем памяти, чем растровые, потому что нет необходимости сохранять информацию о каждом пикселе изображения, для построения изображения нужны только формы и характеристики графических объектов;

в векторном изображении содержится не только описание внешнего вида объектов, но и взаимосвязи между разными объектами, что позволяет автоматизировать процесс изменения объектов (анимации).



Рис. 7. Масштабирование векторного изображения без потери качества

Недостатки векторной графики:

прежде чем вывести векторное изображение на экран монитора или принтер, необходимо выполнить процесс **растеризации** (т. е. представить изображение в растровом виде), что обуславливает дополнительные затраты процессорного времени;

сложно автоматизировать ввод векторной графической информации в компьютер;

векторная графика не может применяться для хранения изображений фотореалистичного качества, так как они могут содержать множество мелких деталей, смазанных или расплывчатых границ и т. д.;

разработано большое количество форматов и стандартов векторной графики. Каждый разработчик имеет собственное мнение о том, какие возможности должны быть у формата или стандарта, вследствие чего разные форматы могут иметь индивидуальные функциональные возможности и разнообразные формы представления графической информации. По этой причине полноценный перевод из одного формата векторной графики в другой проблематичен, так как в процессе перевода неизбежны потери части информации.

1.3. Фрактальная графика

Фрактальная графика, как и векторная, – вычисляемая. Она представляет собой результат автоматической генерации изображений путем математических расчетов, поэтому основой создания фрактальных изображений является не рисование или оформление, а программирование. Фрактальная графика используется в научных и развлекательных программах, играх.

Базовым элементом фрактальной графики является *математическая формула*, изображение строится исключительно по уравнениям.

Простейшим фрактальным объектом является «Снежинка Коха», представляющая собой фрактальный треугольник (рис. 8).

Для создания изображения фрактальной графики используются программные средства *Art Dabbler* (наиболее простой графический редактор, особенно эффективен на начальном этапе освоения компьютерной графики), *ChaosPro* (один из лучших бесплатных генераторов фрактальных изображений), *Ultra Fractal*, *Fractal Explorer*.

Расширения имен файлов фрактальной графики: *frp, frs, fri, fro, fr3, fr4* и др.

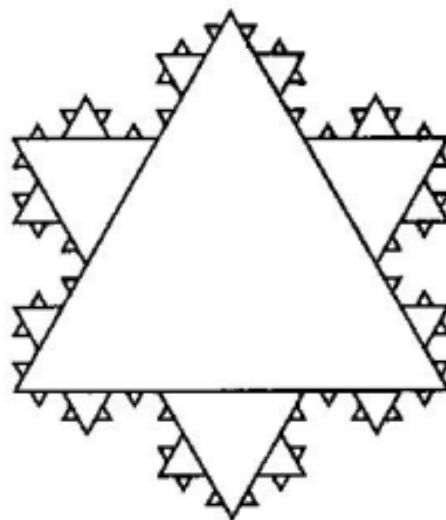


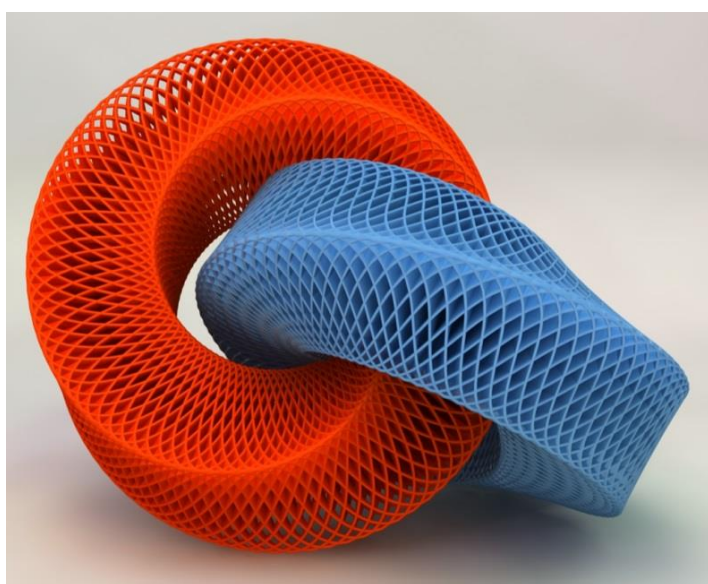
Рис. 8. Фрактальный треугольник

1.4. Трехмерная графика

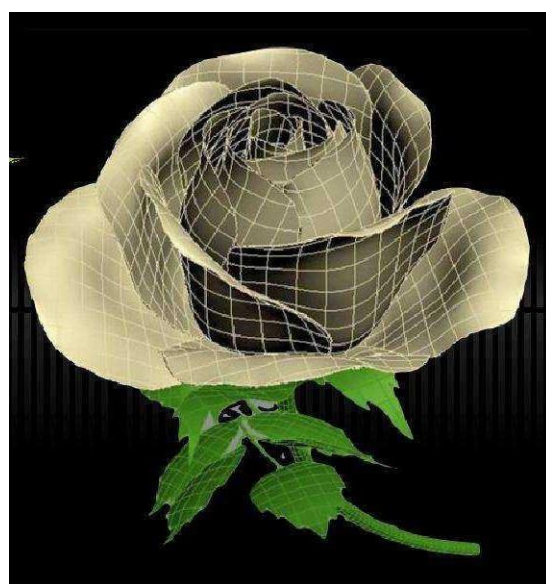
Трехмерная графика широко применяется в таких областях, как научные расчеты, инженерное проектирование, моделирование физических процессов и

технических объектов, а также в обучающих системах и игровых программах. Как правило, в ней сочетаются векторный и растровый способы формирования изображений. Для создания модели трехмерного объекта используются геометрические примитивы (куб, параллелепипед, шар, эллипсоид, конус и др.) и гладкие поверхности, описываемые кусочно-гладкими бикубическими полиномами. Вид поверхности задается сеткой расположенных в пространстве опорных точек. Участки поверхности между опорными точками – границы объекта, которые характеризуются различными свойствами и могут быть гладкими, шероховатыми, прозрачными, непрозрачными, зеркальными и т. п. В соответствии с этими свойствами поверхности закрашиваются тем или иным способом. Движение объектов и анимация воспроизводятся движением геометрических примитивов и опорных точек по заданным зависимостям. Для построения трехмерных изображений и анимации используется достаточно сложное алгоритмическое и программное обеспечение.

Примеры трехмерных изображений приведены на [рис. 9](#).



а



б

[Рис. 9](#). Примеры трехмерных изображений

Для обработки трехмерной графики используются программные средства *Autodesk 3ds Max*, *Maya*, *Blender* (бесплатный графический редактор).

Расширения имен файлов трехмерной графики: *3ds*, *mb*, *3dxml*, *blend* и др.

2. КОДИРОВАНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

2.1. Разрешение изображения, экрана и принтера

Чем больше точек, составляющих растровое изображение, и чем меньше их размер, тем визуально качественнее изображение (и больше размер файла). Качество изображения зависит от его разрешения. **Разрешение изображения** – величина, определяющая количество пикселей (элементов растрового изображения), приходящихся на единицу длины. Оно измеряется в точках на дюйм (*dots per inch – dpi*) или в пикселях на дюйм (*pixel per inch – ppi*). Значение разрешения изображения хранится в файле изображения и неразрывно связано с другим свойством изображения – его **физическим размером**, который может измеряться как в пикселях, так и в единицах длины (миллиметрах, сантиметрах, дюймах). Если изображение готовят для демонстрации на экране, то его ширину и высоту задают в пикселях, чтобы знать, какую часть экрана оно занимает, если для печати – то в единицах длины. Для отображения рисунка на экране монитора достаточно разрешения *72 ppi*, для распечатки на принтере – *300 – 600 ppi*.

Понятие разрешения изображения (разрешения оригинала) не следует путать с понятиями разрешения экрана и разрешения печатающего устройства, поскольку эти понятия характеризуют свойства разных объектов.

Разрешение экрана (экранное разрешение) – это свойство операционной системы, определяющее максимальный размер изображения, которое может поместиться на экране целиком. Самые популярные стандартные разрешения – 1024×768 , 1280×1024 , 1366×768 , 1920×1080 точек.

Разрешение принтера – это свойство принтера, выражающее количество отдельных точек, которые могут быть напечатаны на участке единичной длины. Измеряется в *dpi*. Стандартные разрешения – *300* и *600 dpi*.

2.2. Цветовое разрешение

Цветность изображения характеризуется цветовым разрешением и цветовой моделью. **Цветовое разрешение (глубина цвета)** определяется количеством бит для кодирования (хранения) цвета одного пикселя. Таблица данных, хранящая информацию о коде каждого цвета, называется **цветовой палитрой**.

От цветового разрешения зависит, сколько цветов на экране может воспроизводиться одновременно. Это количество может быть рассчитано по формуле:

$$N_{\text{цв}} = 2^{i_{\text{цв}}}, \quad (1)$$

где $i_{\text{цв}}$ – глубина цвета, бит.

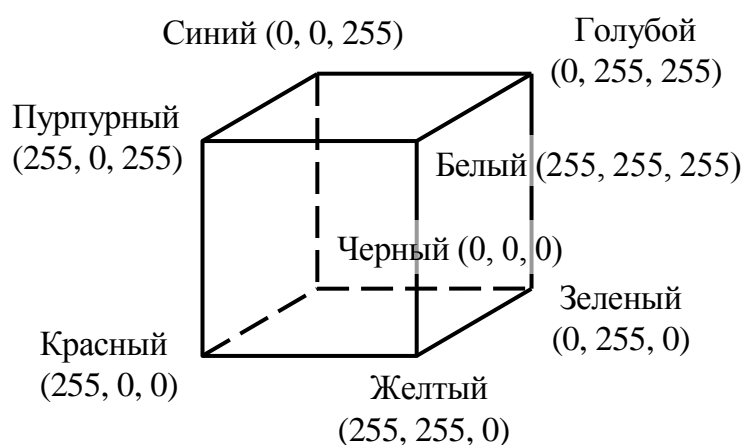
Если на кодирование цвета отводится 1 бит информации, изображение будет монохромным (черно-белым). Один байт информации позволяет закодировать 256 (2^8) цветов, два байта – 65 536 (2^{16}) цветов (режим *High Color*), три байта – 16 777 216 (2^{24}) цветов (режим *True Color*).

2.3. Цветовые модели

Под **цветовой моделью** понимают способ разделения цвета на основные компоненты. Наиболее популярные цветовые модели: *RGB*, *CMYK*, *HSL*.

В основе цветовой **модели RGB** лежит воспроизведение любого цвета путем сложения (смешивания) трех базовых цветов: красного (*Red*), зеленого (*Green*) и синего (*Blue*). Поскольку в модели *RGB* происходит сложение цветов, то она называется *аддитивной*. Именно на такой модели построено воспроизведение цвета современными мониторами и цифровыми экранами.

Этот режим наиболее подходит для передачи богатства красок окружающей природы, но он требует и больших затрат памяти, так как глубина цвета здесь наибольшая – 24 бита (по 8 бит на каждый канал цвета). Каждый базовый цвет характеризуется яркостью, которая может принимать 256 значений – от 0 до 255. При максимальной интенсивности всех трех базовых компонентов получается белый цвет, при отсутствии яркости – черный (табл. 1). Изменяя яркость каждой составляющей, можно смешивать цвета в различных пропорциях и таким образом получить $256 \times 256 \times 256 = 16\,777\,216$ цветовых оттенков. Например,



RGB(255, 117, 56) – цвет, содержащий сильный красный компонент, средней интенсивности зеленый и немного синего. Такая комбинация представляет собой один из оттенков оранжевого цвета.

Цветовым пространством модели *RGB* является цветовой куб (рис. 10).

Рис. 10. Цветовое пространство модели *RGB*

Недостатком модели *RGB* является то, что не все цвета могут быть выведены на печать.

Таблица 1

Кодирование цветов с помощью модели *RGB*

Цвет	Интенсивность тона					
	двоичный код			десятичный код		
	<i>R</i> красный	<i>G</i> зеленый	<i>B</i> синий	<i>R</i> красный	<i>G</i> зеленый	<i>B</i> синий
<i>Основные цвета (комбинации базовых цветов)</i>						
Черный	0000 0000	0000 0000	0000 0000	0	0	0
Синий	0000 0000	0000 0000	1111 1111	0	0	255
Зеленый	0000 0000	1111 1111	0000 0000	0	255	0
Голубой	0000 0000	1111 1111	1111 1111	0	255	255
Красный	1111 1111	0000 0000	0000 0000	255	0	0
Пурпурный	1111 1111	0000 0000	1111 1111	255	0	255
Желтый	1111 1111	1111 1111	0000 0000	255	255	0
Белый	1111 1111	1111 1111	1111 1111	255	255	255
<i>Некоторые оттенки цветов</i>						
Оранжевый	1111 1111	0111 0101	0011 1000	255	117	56
Оливковый	1000 0000	1000 0000	0000 0000	128	128	0
Бирюзовый	0011 0000	1101 0101	1100 1000	48	213	200

В модели *CMYK* (обычно произносится как «цмик») базовыми цветами являются *Cyan* – голубой, *Magenta* – пурпурный, *Yellow* – желтый, *Black* – черный. Эта модель является *субтрактивной* системой, т. е. построенной на вычитании цветов. Три первых цвета описывают отраженный от белой бумаги свет трех базовых цветов аддитивной модели, черный цвет используется для усиления черного при печати (при наложении цветов достичь чистого черного цвета не удастся). Принципиальное отличие модели *CMYK* от *RGB* заключается в том, что в ней максимальная яркость базовых компонентов дает черный цвет, а отсутствие яркости – белый.

Модель *CMYK* наиболее точно описывает цвета при выводе изображения на печать, поэтому используется в полиграфии (указывается количество единиц краски каждого цвета, например, для оранжевого цвета – *CMYK*(0, 54, 78, 1)). Однако типографские краски не могут передать все множество оттенков модели *RGB*, поэтому визуально насыщенность цветов у *CMYK* ниже, чем у *RGB*.

Модель HSL (или *HLS*) является представлением модели *RGB* в цилиндрической системе координат. В этой модели задаются следующие характеристики: *Hue* – оттенок цвета, *Saturation* – насыщенность, *Lightness* – светлота (яркость). Оттенок задается указанием градуса поворота на цветовом круге (от 0 до 360°), например, красный цвет – 0°, желтый – 60°. Насыщенность и светлота указываются в процентах (от 0 до 100 %). Эта модель наиболее приближена к человеческому восприятию и описанию цвета, она часто применяется в графических приложениях, в палитрах цветов и для анализа изображений.

Для сравнения на [рис. 11](#) приведены параметры оранжевого цвета в моделях *RGB* и *HSL* в десятичном коде с учетом особенностей их реализации в операционной системе *Windows* (в стандартном представлении модели *HSL* этот цвет указывается в виде *HSL*(18°, 100 %, 61 %)).

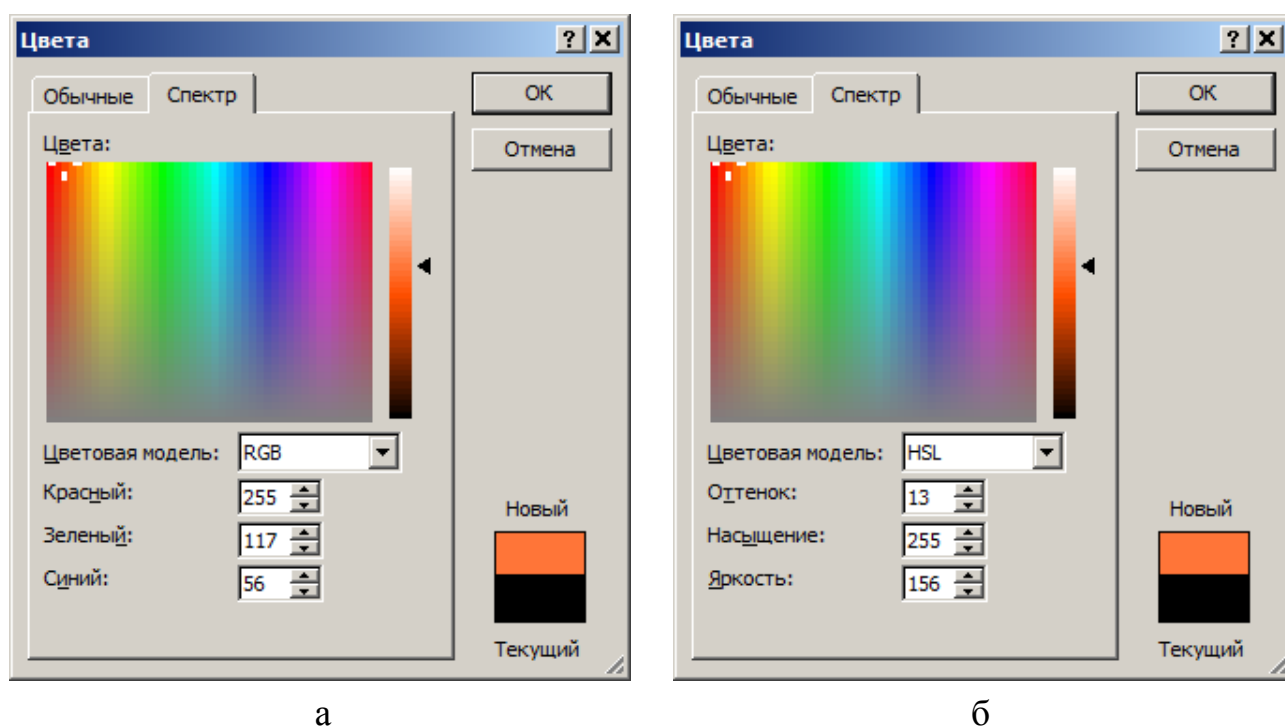


Рис. 11. Параметры оранжевого цвета в моделях *RGB* (а) и *HSL* (б)

2.4. Информационный объем изображения

Общий объем памяти, который требуется для хранения информации обо всех объектах и элементах, представленных на изображении, называется **информационным объемом изображения** и измеряется в единицах измерения информации: битах, байтах, килобайтах, мегабайтах.

2.4.1. Информационный объем растрового изображения

Количество пикселей растрового изображения определяется по формуле:

$$K_{\text{пикс}} = b \cdot h, \quad (2)$$

где b и h – количество пикселей по ширине и по высоте изображения.

Чтобы найти информационный объем растрового изображения в битах, необходимо умножить количество пикселей на глубину цвета:

$$I_{\text{р.и}} = K_{\text{пикс}} \cdot i_{\text{цв}}. \quad (3)$$

Информационный объем изображения обычно не равен размеру файла. Кроме информации о цвете пикселей в файл могут быть записаны различные параметры изображения, в том числе физический и экранный размеры, имя автора, название программы, с помощью которой создано изображение, и т. д., поэтому размер файла может быть больше, чем информационный объем изображения.

С другой стороны, многие форматы файлов используют алгоритмы сжатия графической информации, что позволяет хранить данные более компактно, поэтому размер файла может быть и меньше информационного объема. Иногда размеры файлов, содержащих одно и то же изображение, но имеющих разные форматы, могут отличаться в десятки раз.

Существует большое количество различных методов сжатия растровой графической информации, которые реализуются в различных форматах графических файлов. Одним из таких форматов является формат *JPEG* (*.jpg, *.jpeg, *.jpe, *.jif), сохраняющий данные с большой степенью сжатия за счет потери небольшой части информации. Формат *GIF* (*.gif) обеспечивает сжатие за счет использования ограниченного количества цветов в изображении (максимум 256 цветов). В полиграфии распространенным является специальный формат *TIFF* (*.tif, *.tiff), дающий достаточную степень сжатия и открывающий некоторые дополнительные возможности при печати изображений. Новый формат *HEIC* (*.heic) сохраняет не только растровое изображение, но и его изменения при редактировании (поворот, обрезку и пр.), которые при необходимости можно отменить и вернуться к нужной версии изображения. Изображение в формате *HEIC* имеет в два раза меньший размер файла по сравнению с таким же изображением в формате *JPEG*. В настоящее время этот формат используется только в устройствах компании *Apple*.

2.4.2. Информационный объем векторного изображения

Информационный объем двумерного векторного изображения зависит от количества геометрических объектов (графических примитивов) на изображении и их характеристик и может быть вычислен по формуле:

$$I_{в.и} = \sum_{n=1}^K \left(I_{\phi} + \sum_{i=1}^{K_{\pi}} I_{\pi i} + \sum_{j=1}^{K_{\text{в}}} I_{\text{в}j} \right)_n, \quad (4)$$

где n – номер объекта на изображении;

K – количество объектов на изображении;

I_{ϕ} – объем памяти для хранения информации о геометрической форме (овал, прямоугольник и т. п.) n -го объекта;

K_{π} – количество показателей, характеризующих размеры и положение n -го объекта;

$I_{\pi i}$ – объем памяти для хранения информации об i -м показателе, характеризующем размеры и положение n -го объекта;

$K_{\text{в}}$ – количество показателей, характеризующих цвет и внешний вид (тип и толщина линий, узор для заливки контура) n -го объекта;

$I_{\text{в}j}$ – объем памяти для хранения информации о j -м показателе, характеризующем цвет и внешний вид n -го объекта.

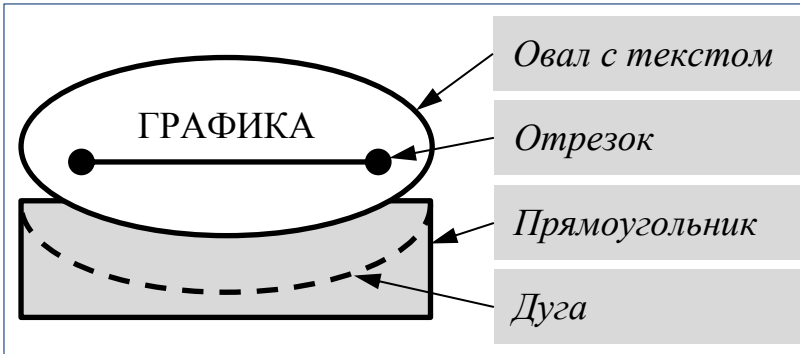


Рис. 12. Векторное изображение

Например, если на изображении представлена фигура, состоящая из четырех геометрических объектов – овала, отрезка, прямоугольника и дуги окружности (рис. 12), то информационный объем этой фигуры без учета текстовых надписей и выносных линий со стрелками определяется по формуле:

$$I_{в.и} = \left(I_{\phi} + \sum_{i=1}^{K_{\pi}} I_{\pi i} + \sum_{j=1}^{K_{\text{в}}} I_{\text{в}j} \right)_{\text{овал}} + \left(I_{\phi} + \sum_{i=1}^{K_{\pi}} I_{\pi i} + \sum_{j=1}^{K_{\text{в}}} I_{\text{в}j} \right)_{\text{отрезок}} + \\ + \left(I_{\phi} + \sum_{i=1}^{K_{\pi}} I_{\pi i} + \sum_{j=1}^{K_{\text{в}}} I_{\text{в}j} \right)_{\text{прямоуг}} + \left(I_{\phi} + \sum_{i=1}^{K_{\pi}} I_{\pi i} + \sum_{j=1}^{K_{\text{в}}} I_{\text{в}j} \right)_{\text{дуга}}. \quad (5)$$

Все параметры, характеризующие каждый геометрический объект в формуле (5), зависят от особенностей векторного графического редактора, с помощью которого создается изображение. **Информационный вес** одного параметра (объем информации для его кодирования) измеряется в битах и зависит от количества возможных значений этого параметра:

$$I = \log_2 N, \quad (6)$$

где N – количество возможных значений параметра.

Если в графическом редакторе предусмотрено использование до 256 различных геометрических фигур, то согласно выражению (6) для кодирования формы любой фигуры достаточно $I_{\phi} = 8$ бит.

Учитывая диапазон возможных значений каждого i -го показателя, характеризующего размеры и положение любого геометрического объекта, для его кодирования требуется $I_{\pi i} = 32$ бита. Исходя из этого определены и представлены в табл. 2 значения общего объема информации I_{π} , необходимого для характеристики размеров и положения базовых геометрических фигур.

Таблица 2

Объем информации для характеристики размеров и положения векторных объектов

Объект	Количество показателей (K_{π})	$I_{\pi} = K_{\pi} \cdot I_{\pi i}$, бит
Отрезок	4	128
Прямоугольник	5	160
Ромб	5	160
Овал	5	160
Параллелограмм	6	192
Прямоугольник со скругленными углами	6	192
Дуга	7	224

Значения объема информации I_{vj} для каждого j -го показателя, характеризующего цвет (с учетом использования модели *RGB*) и внешний вид базовых геометрических фигур, приведены в табл. 3.

Если внутри графического объекта имеется текст, то к информационному объему такого объекта нужно прибавить информационный объем текста, который определяется умножением информационного веса одного текстового символа i_t на количество текстовых символов K_t :

$$I_t = i_t \cdot K_t. \quad (7)$$

Информационный вес одного символа зависит от применяемой в графическом редакторе кодировки. В кодировке Юникод (UTF-8) информационный вес одного символа $i_T = 16$ бит (2 байта).

Формула (7) не учитывает параметры шрифта и параметры абзаца текста.

Таблица 3

Объем информации для характеристики цвета и внешнего вида векторных объектов

Показатель	Количество возможных значений показателя	Информационный вес показателя для различных объектов (I_{vj}), бит		
		отрезок	дуга	прочие фигуры
Толщина линии, пт	1584	32	32	32
Равномерный цвет линии (контура)	16 777 216	24	24	24
Тип начала отрезка	6	3	3	—
Тип конца отрезка	6	3	3	—
Размер начала отрезка	9	4	4	—
Размер конца отрезка	9	4	4	—
Тип штриха	8	3	3	3
Цвет равномерной заливки фигуры	16 777 216	—	24	24
Прозрачность, %	100	7	7	7
Итого (I_v), бит	—	80	104	90

В правой части рис. 12 расположены четыре графических объекта типа **надпись** – это содержащий текстовые символы прямоугольник с прозрачными контуром и заливкой (при необходимости их вид можно изменить).

Расчет полного информационного объема представленного на рис. 12 векторного изображения с учетом текста «ГРАФИКА» в овале, надписей и выносных линий приведен в табл. 4. При этом принято, что значение I_ϕ для каждого объекта равно 8, столбец 4 заполняется по данным табл. 2, столбец 5 – в соответствии с итоговой строкой табл. 3. Количество символов текста в столбце 6 для каждого объекта определяется простым подсчетом, значения в столбце 7 вычисляются по формуле (7) с учетом использования кодировки Юникод. Информационный объем каждого объекта в столбце 8 определяется на основании выражения (4) по формуле:

$$I_{\text{объект}} = I_\phi + I_\Pi + I_v + I_T. \quad (8)$$

По результатам расчетов получаем, что информационный объем векторного изображения (см. [рис. 12](#)) составляет 3684 бита (460,5 байта).

Таблица 4

Расчет информационного объема векторного изображения

Объект		Объем информации, бит			Количество символов текста K_T	I_T , бит	$I_{\text{объект}}$, бит
тип	количество	I_Φ	I_Π	I_B			
1	2	3	4	5	6	7	8
Овал с текстом «ГРАФИКА»	1	8	160	90	7	112	370
Отрезок	1	8	128	80	—	—	216
Прямоугольник с заливкой	1	8	160	90	—	—	258
Дуга	1	8	224	104	—	—	336
Надпись «Овал с текстом»	1	8	160	90	14	224	482
Надпись «Отрезок»	1	8	160	90	7	112	370
Надпись «Прямоугольник»	1	8	160	90	13	208	466
Надпись «Дуга»	1	8	160	90	4	64	322
Выносные линии со стрелками	4	32	512	320	—	—	864
Итого	бит	—	96	1824	1044	—	720
	байт	—	12	228	130,5	—	90

3. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Какие виды компьютерной графики существуют?
- 2) Что является наименьшим элементом растрового изображения?
- 3) Что такое разрешение изображения? В чем оно измеряется?
- 4) Что такое растр? Как соотносятся понятия «ячейка растра» и «пиксель»?
- 5) Что такое глубина цвета? На что она влияет?
- 6) Какие основные цветовые модели существуют? Каким образом формируется цвет в этих цветовых моделях?
- 7) Что является элементарным объектом векторного изображения?

- 8) Как формируется векторное изображение?
- 9) В чем заключаются преимущества и недостатки растровых и векторных изображений?
- 10) От чего зависит информационный объем растрового и векторного изображений?
- 11) Что является базовым элементом фрактальной графики?
- 12) Каким образом строится модель трехмерного изображения?

4. ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Задание 1. Растровая графика (РГ).

Задание выполняется в растровом графическом редакторе *Paint*, входящем в состав операционной системы *Windows*.

1. Запустите редактор *Paint*: **Пуск** → Все программы → Стандартные → *Paint*¹. Сохраните созданное пустое изображение в личной папке в файле с именем *Фамилия_РГ_Экр1*². При сохранении выберите тип файла «24-разрядный рисунок» (расширение имени файла *bmp*).




2. Сверните окно редактора *Paint* (не закрывайте его) и в центре рабочего стола создайте папку с именем *Фамилия_Группа*. При помощи контекстного меню откройте свойства созданной папки, на вкладке *Настройка* нажмите кнопку *Сменить значок...* и выберите понравившийся вам значок папки, отличающийся от стандартного значка. Нажмите клавишу *Print Screen*, в результате чего изображение экрана с видом рабочего стола и значком созданной папки (далее – изображение «Снимок экрана») будет скопировано в буфер обмена.

3. Разверните окно редактора *Paint* и вставьте в него полученное изображение, выполнив команду **Главная** → Буфер обмена → *Вставить*. Затем с помощью инструмента **Главная** → Изображение → *Выделить* выделите изображение значка созданной папки с ее названием. Выполнив команду **Главная** → Изображение → *Изменить размер*, увеличьте размер изображения значка до 300 × 300 пикселей. Для этого в появившемся окне *Изменение размеров и наклона* выберите опцию *Пиксели*, отключите флажок *Сохранить пропорции* и введите требуемый размер

¹ Здесь и далее путь для запуска программы указывается в виде: **Пуск** → Название группы программ → *Название программы*, путь доступа к рассматриваемому элементу управления (команде) – **Вкладка ленты** → Группа → *Элемент управления*.

² Здесь и далее вместо слова «Фамилия» указывается фамилия студента, выполнившего работу, вместо слова «Группа» указывается группа, в которой обучается студент.

в пикселях по горизонтали и по вертикали. После завершения операции убедитесь в наличии эффекта пикселизации увеличенного значка.

4. Сверните все открытые окна с помощью комбинации клавиш **Windows**  + **D**. Запустите программу *Ножницы*: **Пуск** → Все программы → Стандартные → *Ножницы* (при этом изображение на экране должно стать прозрачным, если этого не произошло, нажмите в окне *Ножницы* кнопку *Создать*).левой кнопкой мыши выделите значок корзины на рабочем столе так, чтобы вошла также надпись «Корзина». Далее в появившемся окне *Ножницы* с помощью инструмента  *Маркер* выделите цветом слово «Корзина» и нажмите кнопку  *Копировать*. Откройте окно редактора *Paint*, вставьте скопированное изображение корзины, увеличьте его до размера 200 × 200 пикселей и переместите в правый нижний угол изображения «Снимок экрана».

5. Сохраните полученное изображение в текущем файле **Фамилия_РГ_Экр1** (тип файла – «24-разрядный рисунок»).

6. Выполнив команду **Файл** → Сохранить как, сохраните полученное изображение в файле с именем **Фамилия_РГ_Экр2** как рисунок формата *JPEG*, а затем в файле с именем **Фамилия_РГ_Экр3** как рисунок формата *GIF*, указав при сохранении соответствующие типы файлов.

7. Создайте в редакторе *Paint* новое изображение, установите его размер 800 × 800 пикселей и с помощью графических инструментов редактора изобразите блок-схему по образцу, представленному на [рис. 13](#) (далее – изображение «Блок-схема»).

В верхнем блоке вместо слова «Начало» впишите свою фамилию и сделайте заливку светло-зеленым цветом, в нижнем – вместо слова «Конец» впишите свою группу и примените заливку светло-оранжевым цветом. Полученное изображение сохраните с именем **Фамилия_РГ_Сх1** как «24-разрядный рисунок».

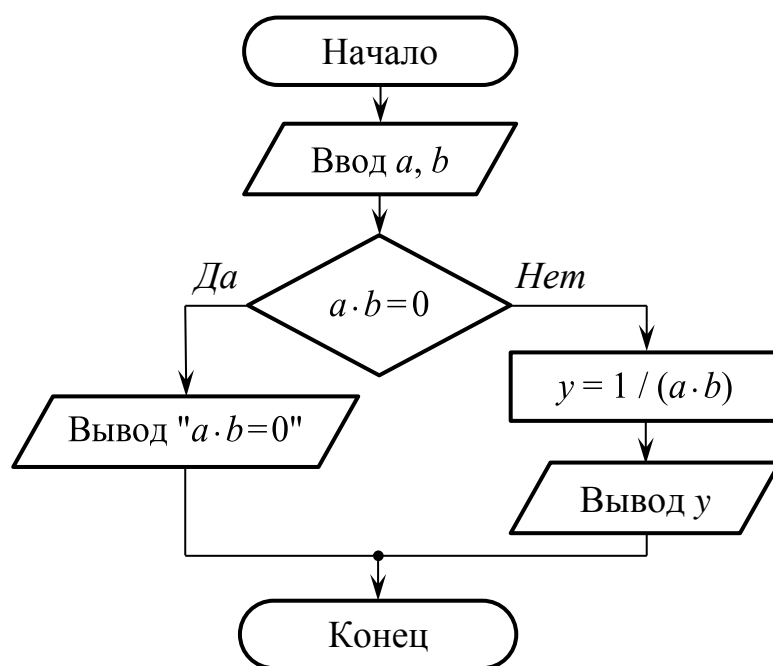


Рис. 13. Блок-схема

8. Выполнив команду **Файл** → **Сохранить как**, сохраните полученное изображение в файле с именем **Фамилия_РГ_Сх2** как рисунок формата *JPEG*, а затем в файле с именем **Фамилия_РГ_Сх3** как рисунок формата *GIF*, выбрав при сохранении соответствующие типы файлов.

9. Откройте табличный процессор *Microsoft Excel* и сохраните созданную пустую рабочую книгу³ с именем **Фамилия_Графика** в личной папке.

10. Создайте в рабочей книге *Excel* **Фамилия_Графика** новый рабочий лист, назовите его «**Фамилия_РГ**» и заполните на нем таблицу по образцу **табл. 5** в соответствии с представленными ниже рекомендациями.

Таблица 5

Сравнение характеристик файлов с растровыми изображениями

Показатели			Изображение	
			Снимок экрана	Блок-схема
Количество пикселей	по ширине			
	по высоте			
	всего			
Характеристики файлов	тип файла – 24-разрядный рисунок ($i_{\text{цв}} = 24$ бита)	Имя файла	Фамилия_РГ_Экр1	Фамилия_РГ_Сх1
		Качество изображения (+/–)	+	+
		Размер файла, байт		
		Информационный объем изображения, байт		
	тип файла – <i>JPEG</i> ($i_{\text{цв}} = 24$ бита)	Имя файла	Фамилия_РГ_Экр2	Фамилия_РГ_Сх2
		Качество изображения (+/–)		
		Размер файла, байт		
		Информационный объем изображения, байт		
	тип файла – <i>GIF</i> ($i_{\text{цв}} = 8$ бит)	Имя файла	Фамилия_РГ_Экр3	Фамилия_РГ_Сх3
		Качество изображения (+/–)		
		Размер файла, байт		
		Информационный объем изображения, байт		

³ По указанию преподавателя в качестве исходной рабочей книги может быть использован специальный файл *Excel-шаблон.xlsm*.

Размер изображения по ширине и по высоте в пикселях можно узнать в свойствах соответствующего файла на вкладке *Подробно*, размер файла – на вкладке *Общие* в пункте *Размер* (в байтах).

В ячейки таблицы по строкам «Качество изображения (+/–)» впишите символ «+» в случае полного соответствия изображения в файле исходному изображению в формате 24-разрядного рисунка. В случае визуального отличия изображения от исходного (искажение цветов, зернистость и др.) впишите в ячейки символ «–».

В ячейки, отмеченные серой заливкой, введите формулы *Excel* для расчета требуемых показателей: общее количество пикселей определяется по формуле (2), информационный объем изображения – по формуле (3).

11. Сохраните содержимое рабочей книги *Excel* *Фамилия_Графика*.

Задание 2. Векторная графика (ВГ).

Задание выполняется в текстовом процессоре *Microsoft Word*, который содержит в себе встроенный векторный графический редактор.

1. Откройте текстовый процессор *Microsoft Word* и с помощью его графических инструментов в новом документе изобразите блок-схему по образцу, представленному на [рис. 13](#). В верхнем блоке вместо слова «Начало» впишите свою фамилию и сделайте заливку светло-зеленым цветом, в нижнем вместо слова «Конец» впишите свою группу и примените заливку светло-оранжевым цветом. С помощью инструментов в группе **Формат** → **Размер** увеличьте размеры блоков «Начало» и «Конец» в два раза, так же увеличьте размер шрифта в них. Убедитесь в отсутствии эффекта пикселизации увеличенных блоков.

2. С помощью команды **Главная** → **Редактирование** → **Выделить** → **Область выделения...** откройте список имеющихся фигур на странице, при нажатой клавише *Ctrl* выделите их все, а затем объедините в одну группу, выполнив команду **Формат** → **Упорядочить** → **Группировать**.

3. С помощью пункта меню **Файл** → **Сохранить как** сохраните полученное изображение в трех разных форматах:

в файле с именем *Фамилия_ВГ_Сх1* – в формате *Word* (тип файла – «Документ Word»);

в файле с именем *Фамилия_ВГ_Сх2* – в формате *Word 97-2003* (тип файла – «Документ Word 97-2003»);

в файле с именем *Фамилия_ВГ_Сх3* – в формате *RTF* (тип файла – «Текст в формате RTF»).

4. Откройте рабочую книгу *Excel* *Фамилия_Графика*. Создайте в ней новый рабочий лист с именем «Фамилия_ВГ_Расчет», организуйте на нем таблицу по образцу табл. 4, в которой перечислите все графические элементы блок-схемы и выполните расчет ее информационного объема. При этом данные в ячейках столбцов 7, 8 и строк «Итого» определите с помощью формул *Excel*.

5. Создайте в рабочей книге *Excel* *Фамилия_Графика* новый рабочий лист с именем «Фамилия_ВГ» и заполните на нем таблицу по образцу табл. 6:

данные в столбцах «Качество изображения (+/–)» и «Размер файла, байт» определите аналогично табл. 5;

в последнем столбце введите ссылку на ячейку рабочего листа «Фамилия_ВГ_Расчет», содержащую результат расчета информационного объема векторного изображения блок-схемы (он не зависит от типа файла).

6. Сохраните содержимое рабочей книги *Excel* *Фамилия_Графика*.

Таблица 6

Сравнение характеристик файлов с векторным изображением

Тип файла	Имя файла	Качество изображения (+/–)	Размер файла, байт	Информационный объем изображения, байт
<i>Word (*.docx)</i>	Фамилия_ВГ_Сх1	+		
<i>Word 97-2003 (*.doc)</i>	Фамилия_ВГ_Сх2			
<i>RTF (*.rtf)</i>	Фамилия_ВГ_Сх3			

7. Проанализируйте данные в рабочей книге *Excel* *Фамилия_Графика*, полученные по итогам выполнения работы, и запишите в лабораторную тетрадь вывод, в котором обоснуйте выбор наиболее подходящего формата файла для хранения фотореалистичных изображений (снимок экрана, фотография и др.) и схем (блок-схема, инженерный чертеж и др.).

Библиографический список

1. Трофимов В. В. Информатика: В 2 т.: Учебник / В. В. Трофимов, О. П. Новожилов. М.: Юрайт, 2017. Т. 1. 553 с.

2. Гаврилов М. В. Информатика и информационные технологии: Учебник / М. В. Гаврилов, В. А. Климов. М.: Юрайт, 2017. 383 с.

3. Кудинов Ю. И. Основы современной информатики: Учебное пособие / Ю. И. Кудинов, Ф. Ф. Пашенко. СПб: Лань, 2017. 256 с.

Учебное издание

СИДОРОВА Елена Анатольевна,
САЛЯ Илья Леонидович

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ ГРАФИКИ

Учебно-методическое пособие

Редактор Н. А. Майорова

Подписано в печать 16.05.2018. Формат $60 \times 84 \frac{1}{16}$.
Офсетная печать. Бумага офсетная. Усл. печ. л. 1,8. Уч.-изд. л. 2,0.
Тираж 500 экз. Заказ .

**

Редакционно-издательский отдел ОмГУПСа
Типография ОмГУПСа

*

644046, г. Омск, пр. Маркса, 35