**Глава 3. ЭЛЕМЕНТ CANVAS**

**3.1. Назначение элемента CANVAS и принципы его использования**

Элемент <canvas> – новый элемент языка HTML5, предназначенный для растрового двухмерного изображения при помощи JavaScript. Впервые элемент был представлен в движке WebKit для Mac OS X.

С помощью <canvas> на холсте (англ. canvas – холст) можно разместить линии, геометрические фигуры, текст, картинки (рис. 3.1). Все изображения можно обвести контуром, залить сплошным цветом или добавить цветовой градиент.

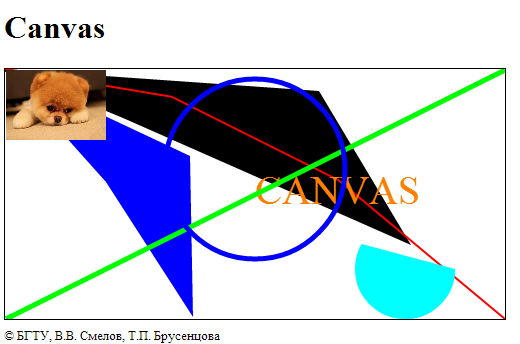


Рис. 3.1. Пример холста элемента <canvas> c построенными изображениями

Собственно сам элемент <canvas> только задает в окне браузера поверхность (холст) указанных размеров (атрибуты width и height). Каждый пиксель холста имеет свою координату. Начало координат находится в левом верхнем углу холста. Для того, чтобы на холсте появилось изображение, следует разработать сценарий на JavaScript, использующий программный интерфейс Canvas API.

На сегодняшний день <canvas> чаще всего используется для построения графиков, простой анимации и игр в браузерах.

**3.2. Применение элемента CANVAS**

Рассмотрим простое одностраничное web-приложение демонстрирующее применение элемента <canvas>.

На рис. 3.2 представлен исходный код html-страницы, содержащей элемент <canvas>. Обратите внимание на следующее: размеры холста задаются атрибутами width и height; между открывающим и закрывающим тегами элемента <canvas> располагается сообщение, которое будет выводиться, если элемент <canvas> не может быть отображен браузером.



Рис. 3.2. Применение элемента <canvas> в html-странице

Весь JavaScript-код, который будет использоваться далее для формирования изображения на холсте располагается в файлe Canvas.js.

На рис. 3.3 приведен фрагмент файла Canvas.js. В этом фрагменте объявлены три глобальные переменные, а также анонимная функция, выполняющаяся при каждой загрузке html-страницы. В функции формируются значения глобальных переменных: размеры холста (canvasW и canvasH) и ссылка на объект (context), полученный с помощью метода getContext и служащий для управления холстом элемента <canvas>.

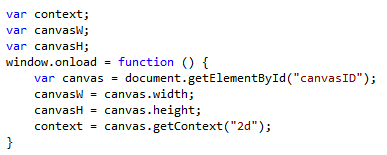


Рис. 3.3. Фрагмент файла Canvas.js: объявление и инициализация глобальных переменных

Обратите внимание на содержимое переменной глобальной переменной context: методы объекта, ссылку на который содержит эта переменная, позволяют наносить изображение на холст элемента <canvas>.

**3.2.1. Отображение прямоугольника и очистка холста**

На рис. 3.4 представлен содержимое двух элементов <section> html-страницы (рис. 3.2).



Рис. 3.4. Фрагмент html-страницы (рис. 3.2) отвечающий за управление изображением прямоугольников на холсте

Первый элемент <section> (рис. 3.4) содержит закрашенную клавишу «Прямоугольник» (рис. 3.5), нажатие которой приводит к вызову функции exe\_fillRect, расположенной в файле Canvas.js. Четыре поля, следующие за клавишей позволяют установить координаты верхнего левого угла изображаемого прямоугольника, его ширину, высоту и цвет заливки.

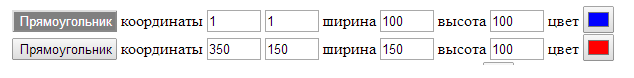


Рис. 3.5. Изображение фрагмента html-страницы (рис. 3.4) в браузере

Содержимое второго <section> (рис. 3.4) практически совпадает с содержимым первого, но при нажатии клавиши будет вызвана функция exe\_strokeRect, которая тоже расположена в файле Canvas.js.

На рис. 3.6 представлен фрагмент содержимого файла Canvas.js, содержащий функции exe\_fillRect и exe\_strokeRect.

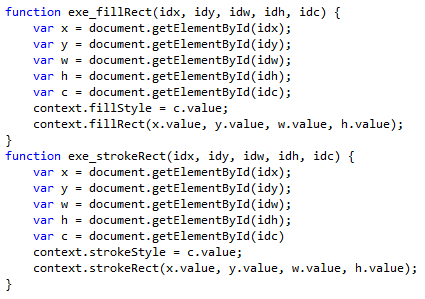


Рис. 3.6. Фрагмент содержимого файла Canvas.js: функции exe\_fillRect и exe\_strokeRect

Функции exe\_fillRect и exe\_strokeRect (рис.3.6) принимают одинаковый набор параметров – идентификаторы элементов, содержащие: координаты левого верхнего угла прямоугольника (idx и idy), ширину (idw), высоту (idh) и цвет (idc).

Отображение прямоугольника в функции exe\_fillRect, осуществляется с помощью метода fillRect объекта управления холстом, хранящегося в глобальной переменной context (рис. 3.3). Функция принимает четыре параметра: два первых – координаты левого верхнего угла прямоугольника, два следующих – ширина и высота. Метод fillRect отображает закрашенный прямоугольник, причем цвет заливки определяется значением свойства fillStyle, которое должно быть установлено до вызова функции.

Функция exe\_strokeRect имеет структуру, аналогичную функции exe\_fillRect, но для отображения прямоугольника применяется метод strokeRect, а цвет устанавливается с помощью свойства strokeStyle. Метод strokeRect отображает контур прямоугольника.

На рис. 3.7 приведен внешний вид холста, после последовательного нажатия двух клавиш «Прямоугольник» (рис. 3.4, 3.5).

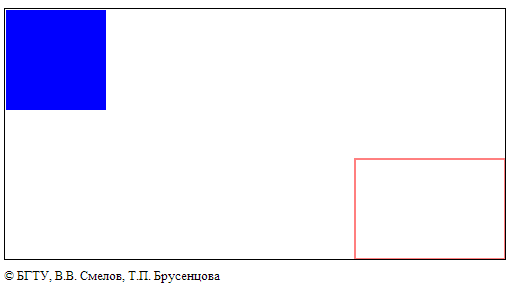


Рис. 3.7. Холст, с изображенными прямоугольниками

Очистка фрагментов холста осуществляется прямоугольными областями. На рис. 3.8 представлено содержимое элемент <section> html-страницы на рис. 3.2. Элемент содержит клавишу «Очистить» и четыре поля для ввода координат, ширины и высоты области очистки (рис. 3.9).

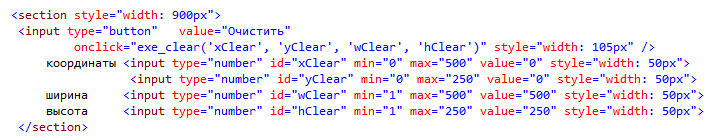


Рис. 3.8. Фрагмент html-страницы (рис. 3.2) отвечающий за управление изображением прямоугольников на холсте



Рис. 3.9. Изображение фрагмента html-страницы (рис. 3.8) в браузере

При нажатии клавиши «Очистить» вызывается функция exe\_clear, расположенная в файле Canvas.js (рис. 3.10). Функция принимает четыре параметра – идентификаторы элементов, содержащие: координаты левого верхнего угла (idx, idy), ширину (idw) и высоту (idh), прямоугольной очищаемой области. Очистка осуществляется с помощью метода clearRect. Область холста, подлежащая очистке, задается параметрами метода: координатами верхнего левого угла, шириной и высотой.

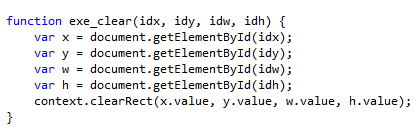


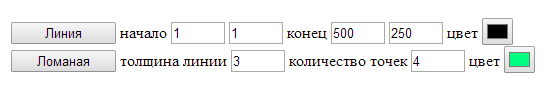
Рис. 3.10. Фрагмент содержимого файла Canvas.js: функции exe\_clear

**3.2.2. Отображение прямых линий**

На рис. 3.11 представлено содержимое двух элементов <section> html-страницы (рис. 3.2). Браузер отобразит эти элементы примерно так, как это изображено на рис. 3.12.



Рис. 3.11. Фрагмент html-страницы (рис. 3.2) отвечающий за управление изображением на холсте линий



3.12. Изображение фрагмента html-страницы (рис. 3.11) в браузере

Клавиша, расположенная в первом элементе <section>, предназначена для вывода на холст элемента <canvas> прямой линии, заданной координатами двух точек (элементы с идентификаторами). При нажатии клавиши, вызывается функция exe\_line (рис. 3.13), принимающая пять параметров: идентификаторы элементов <input>, предназначенных для ввода координат точек и цвета.

Клавиша, расположенная во втором элементе <section>, предназначена для вывода на холст ломаной линии, для которой задается толщина линии, количество звеньев и цвет. При нажатии клавиши вызывается функция exe\_polyline, принимающая три параметра: идентификаторы элементов <input>, предназначенных для ввода толщины линии, количества точек (звеньев) и цвета линии.

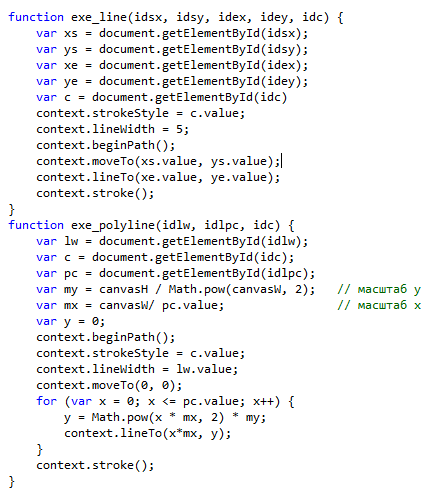


Рис. 3.13. Фрагмент содержимого файла Canvas.js: функции exe\_line и exe\_polyline

Для отображения прямой линии в функции, последовательно выполняются четыре метода объекта управления холстом (ссылка в переменной canvas): beginPath, moveTo, lineTo и stroke.

Метод beginPath не принимает никаких параметров и служит для обозначения старта изображения нового контура.

Метод moveTo принимает два параметра (координаты начальной точки) и задает текущее положение точки на холсте.

Метод lineTo принимает два параметра (координаты конечной точки) и рисует (но не отображает на холсте) прямую линию, соединяющую текущую и конечную точки. Кроме того, lineTo устанавливает новое текущее положение точки на холсте.

Собственно отображение линии осуществляется методом stroke. При этом принимаются по внимание установленные предварительно свойства: цвет (strokeStyle) и толщина (lineWidth) линии.

Ломаная линия отображается функцией exe\_polyline. Принцип ее отображения заключается в следующем. После прорисовки первого звена (как это было сделано в первом случае), следует снова вызвать метод lineTo, который будет рисовать очередное звено. В данном примере, значение первой координаты вычисляется исходя из ширины холста и количества заданных точек ломаной, а значение второй, как квадрат значения первой с коэффициентом, позволяющим отмасштабировать на всю высоту холста.

На рис. 3.14 приведен внешний вид холста, после последовательного нажатия двух клавиш «Линия» и «Ломаная» (рис. 3.14, 3.12).

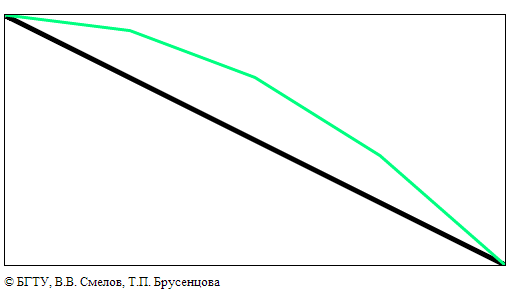


Рис. 3.14. Холст, с изображенными прямой и ломаной линий

**3.2.3. Отображение дуг, окружностей и кругов**

На рис. 3.15 представлено содержимое двух элементов <section> html-страницы (рис. 3.2). Браузер отобразит эти элементы примерно так, как это изображено на рис. 3.16.



Рис. 3.15. Фрагмент html-страницы (рис. 3.2) отвечающий за управление изображением на холсте окружностей и кругов

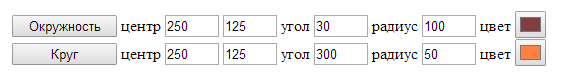


Рис. 3.16. Изображение фрагмента html-страницы (рис. 3.15) в браузере

Нажатие клавиш (рис. 3.15, 3.16) привет вызову функций exe\_arc и exe\_carc, расположенных в файле Canvas.js (рис. 3.17), Обе функции принимают одинаковый набор из четырех параметров: идентификаторы элементов <input>, предназначенных для ввода координат центра, величины угла и цвета.

В обеих функциях применяется метод arc, принимающий шесть параметров: координаты центра, радиус, начальный и конечный углы дуги и направление дуги. Метод arc рисует дугу с заданным центром и радиусом, начало и конец дуги задаются величиной угла в радианах. Дуга рисуется в зависимости от последнего параметра по часовой стрелке (значение false) или против часовой стрелки (значение true). Следует помнить, что углы откладываются всегда против часовой стрелки от горизонтальной оси.

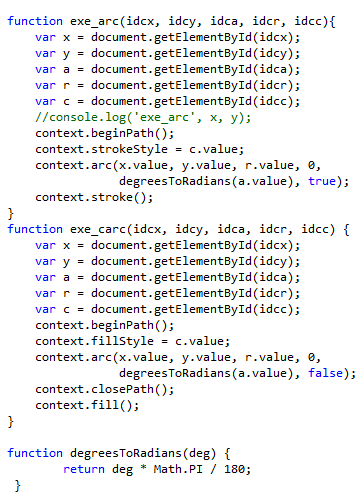


Рис. 3.17. Фрагмент содержимого файла Canvas.js: функции exe\_arc и exe\_carc

Функция exe\_arc отображает контур дуги, а функция exe\_сarc замкнутую закрашенную часть круга. Обратите внимание: в обеих функциях величина угла начала дуги равна нулю; в функции exe\_arc дуга рисуется против часовой стрелки, в функции exe\_сarc – против; в обеих функциях для старта контура применяется метод beginPath; только в функции exe\_сarc применяется функция closePath, предназначенная за замыкания контура (соединения первой и последней точки); в функции exe\_arc для указания цвета контура применяется свойство strokeStyle, а для отображения дуги метод stroke; в функции exe\_carc для указания цвета заливки применяется свойство fillStyle, а для отображения части круга метод fill;

На рис. 3.18 приведен внешний вид холста, после последовательного нажатия двух клавиш «Окружность» и «Круг» (рис. 3.15, 3.16).

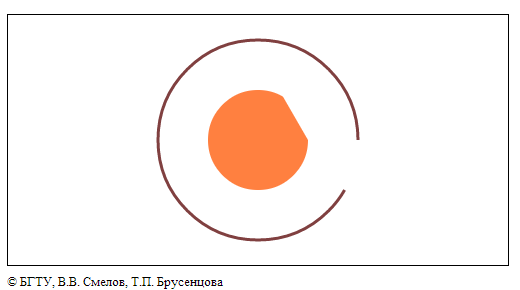


Рис. 3.18. Холст, с изображенными окружности и круга

**3.2.4. Отображение полигона**

Полигон – залитая цветом плоскость, ограниченная замкнутой ломаной линией. На рис. 3.19 представлено содержимое элемента <section> html-страницы (рис. 3.2). Браузер отобразит этот элемент примерно так, как это изображено на рис. 3.20.

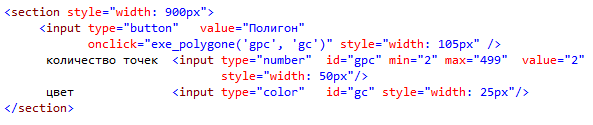


Рис. 3.19. Фрагмент html-страницы (рис. 3.2) отвечающий за управление изображением полигона



Рис. 3.20. Изображение фрагмента html-страницы (рис. 3.19) в браузере

Нажатие клавиши (рис. 3.19, 3.20) привет вызову функции exe\_polygone, расположенной в файле Canvas.js (рис. 3.21). Функция принимает два параметра: идентификаторы элементов <input>, предназначенных для ввода количества точек и цвета заливки.

По сути, для построения полигона необходимо построить замкнутую ломаную линию и залить, ограниченную контуром поверхность холста необходимым цветом.

В функции polygone координаты ломаной линии, ограничивающей полигон, формируются случайным образом. Для этого применяется стандартная функция Math.random, результат выполнения которой масштабируется функцией getRandomInt по размерам холста.

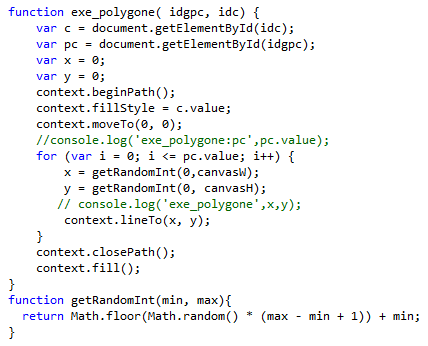


Рис. 3.21. Фрагмент содержимого файла Canvas.js: функция exe\_polygone

Обратите внимание на следующее: построение ломаной завершается выполнением метода closePath, осуществляющим замыкание; для отображения полигона применяется метод fill, который использует для заливки цвет, установленный в свойстве fillStyle.

На рис. 3.22 приведен примерный внешний вид холста, после нажатия клавиши «Полигон» (рис. 3.19, 3.20).

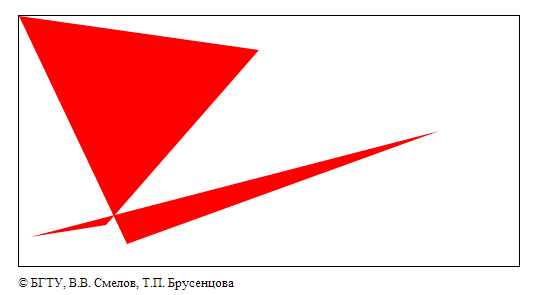


Рис. 3.22. Холст, с изображенными случайного полигона

**3.2.5. Отображение текста**

На холсте элемента <canvas> можно отобразить текст. На рис. 3.23 представлено содержимое элемента <section> html-страницы (рис. 3.2). Браузер отобразит этот элемент примерно так, как это выглядит на рис. 3.23. Нажатие клавиши «Текст» позволяет отобразить текст, содержащийся в элементе <input id=”tt”>, с заданными другими элементами <input> характеристиками.

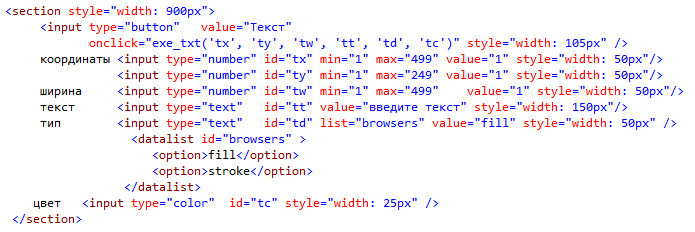


Рис. 3.23. Фрагмент html-страницы (рис. 3.2) отвечающий за управление изображением текста



Рис. 3.24. Изображение фрагмента html-страницы (рис. 3.22) в браузере

При нажатии клавиши «Текст», вызывается функция exe\_txt, расположенная в файле Canvas.js (рис. 3.25). Функция принимает шесть параметров: идентификаторы элементов <input>, предназначенных для ввода

координат отображаемого текста, ширины поля, собственно текста, способа отображения и цвета.

Текст на холсте может отображаться двумя способами: заливкой (fill) или обрисовкой контуров символов (stroke). В первом случае для вывода текста применяется метод strokeText во втором fillText (рис. 3.25). В зависимости от значения, которое содержится в элементе <input id=”td”> (рис. 3.23, 3.24), в функции exe\_txt (рис. 3.25) выполняется та или иная функция.

Текст на холсте отображается вдоль условной неотображаемой линии, называемой базовой линией. Координаты начала ее заданы вторым и третьим параметрами функций strokeText и fillText. Свойство textBaselile определяет расположение базовой линии, относительно текста. Возможны следующие варианты: alphabet или bottom (внизу текста), middle (по середине текста), top (сверху текста).

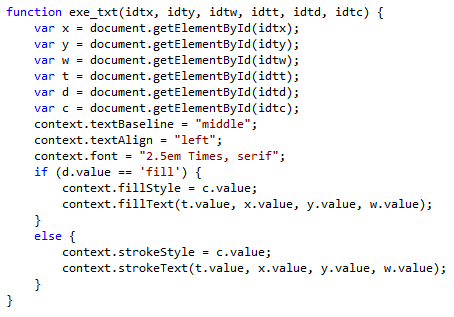


Рис. 3.25. Фрагмент содержимого файла Canvas.js: функция exe\_txt

Тип выравнивания текста задается с помощью свойства textAlign. Допускаются следующие значения этого свойства: left (слева), right (справа), center (по центру), start (на начало текста), end (на конец текста). Значения right и start отличаются в том случае, если его слова в нем пишутся слева на правою

Обе функции (fillText и strokeText) имеют одинаковый набор параметров: собственно текст, координаты начала базовой линии, максимальная ширина поля для вывода текста Обратите внимание: цвет заливки для функции fillText задается свойством fillStyle, а для функции strokeText – свойством strokeStyle;

Тип шрифта, его размер и прочие характеристики отображаемого текста могут быть заданы свойством font.

На рис. 3.26 приведен примерный внешний вид холста, после размещения на нем двух строк.

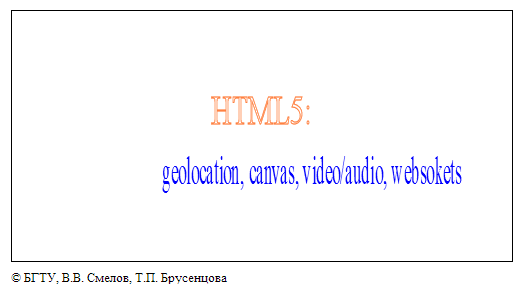


Рис. 3.26. Холст, с изображенным текстом

Обратите внимание: первая строка изображена обводкой контура (функция strokeText), вторая – заливкой (функция fillText).

**3.2.6. Отображение картинки**

На хост элемента <canvas> можно поместить картинку. Формат картинки в общем случае зависит от браузера, но, как правило, поддерживаются форматы PNG, GIF и JPEG.

На рис. 3.27 представлено содержимое элемента <section> html-страницы (рис. 3.2). Браузер отобразит этот элемент примерно так, как это выглядит на рис. 3.28. Нажатие клавиши «Картинка» позволяет отобразить картинку, выбранную с помощью элемента <input id=”if”>, с заданными другими элементами <input> координатами левого верхнего угла, шириной и высотой.

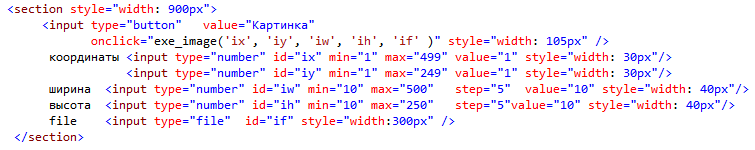


Рис. 3.27. Фрагмент html-страницы (рис. 3.2) отвечающий за управление изображением картинки



Рис. 3.28. Изображение фрагмента html-страницы (рис. 3.27) в браузере

При нажатии клавиши «Картинка» (рис. 3.26, 3.37) вызывается функция exe\_image (рис. 3.29). Функция принимает пять параметров: идентификаторы элементов <input>, предназначенных для ввода координат отображаемой картинки текста, ширины, высоты и файла картинки.

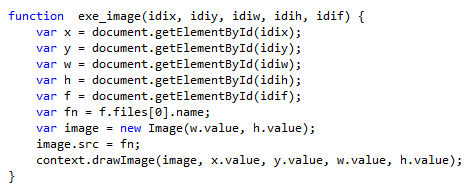


Рис. 3.29. Фрагмент содержимого файла Canvas.js: функция exe\_image

Вывод картинки на холст элемента <canvas> осуществляется методом drawImage (рис. 3.39). Функция принимает пять параметров: объект Image, координаты и размеры картинки. Объект Image, создается динамически до вызова функции drawImage, при вызове конструктора указываются размеры картинки. Свойству src объекта присваивается имя файла картинки, которое извлекается из коллекции свойства files элемента <input type=”file”>.

Задаваемые размеры картинки в функции drawImage не обязательно должны совпадать с исходными размерами которые можно увидеть в свойствах файла.

На рис. 3.30 приведен примерный внешний вид элемента <canvas> (рис. 3.2), после вывода на него картинки формата JPEG, исходные размеры которой 1536×2048 пикселей.

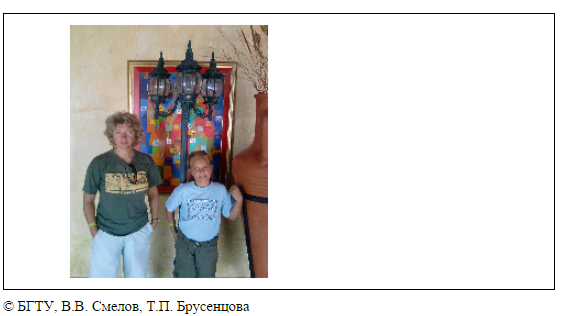


Рис. 3.30. Холст, с изображенной картинки