

Отзывы о книге

«Раньше создать компьютерную программу было столь же просто, как включить компьютер и набрать одну или две строки кода, чтобы на экране появилось слово «Привет». Теперь для этого требуется руководство в 500 страниц и целая команда программистов. Или не все так плохо? Благодаря этой небольшой книге Бена и Кейси вы сможете рисовать линии, треугольники и круги при помощи компьютерной программы спустя несколько минут после включения компьютера. Они снова сделали компьютерные программы простыми и понятными для обычного человека – а это немалое достижение».

– *Джон Маэда, президент школы дизайна Род-Айленда*

«Это не только простое введение в основы программирования – это еще и весело! Книга похожа на рабочую тетрадь для взрослых. Вам захочется купить ее, даже если вы никогда не думали, что заинтересуетесь программированием, – потому что вы обязательно заинтересуетесь».

– *Марк Аллен, основатель и директор Machine Project*

«Это отличный учебник для тех, кто хочет окунуться в программирование графики. Методика “обучение через практику” делает его особенно подходящим для художников и дизайнеров, которых часто отталкивают более традиционные подходы, основанные на теории. Доступная стоимость книги и тот факт, что среда Processing имеет открытый исходный код, делает ее отличным выбором для студентов».

– *Джиллиан Крэмpton Смит,
профессор дизайна Fondazione Venezia,
Венецианский университет IUAV*

«Processing радикально изменил способ обучения программированию, и это один из основных факторов успеха Arduino».

– *Массимо Банци, соучредитель Фонда Arduino*

«Кейси Риас и Бен Фрай в своей книге раскрыли захватывающую силу программирования для творческих людей. Риас и Фрай ясны и прямолинейны, но как художники они не боятся быть немного эксцентричными и нестандартными. Это делает их способ обучения уникальным и эффективным».

– *Холли Уиллис, директор академических программ,
Институт мультимедийного образования,
Школа кинематографических искусств, USC*

Содержание

Отзывы о книге	5
От издательства	12
Предисловие	13
Глава 1. Знакомство с языком Processing	17
1.1. Скетчи и прототипы	18
1.2. Гибкость	19
1.3. Гиганты прошлого.....	20
1.4. Генеалогическое древо языков	21
1.5. Присоединяйтесь к сообществу!.....	21
Глава 2. Начинаем программировать	22
2.1. Ваша первая программа.....	23
Пример 2.1. Рисование эллипса.....	23
Пример 2.2. Рисование кругов	24
2.2. Режим отображения рабочего окна.....	25
2.3. Создание и сохранение нового скетча.....	25
2.4. Распространение программы	26
2.5. Примеры и ссылки	26
Глава 2. Рисование	28
3.1. Рабочее окно	28
Пример 3.1. Рисование рабочего окна.....	29
Пример 3.2. Рисование точки.....	29
3.2. Основные фигуры	29
Пример 3.3. Рисование линии.....	31
Пример 3.4. Рисование основных фигур.....	31
Пример 3.5. Рисование прямоугольника.....	31
Пример 3.6. Рисование эллипса.....	32
Пример 3.7. Рисование сегмента	33
Пример 3.8. Использование градусов	33
3.3. Порядок рисования.....	34
Пример 3.9. Управление порядком рисования	35
Пример 3.10. Обратный порядок рисования	35
3.4. Свойства фигуры	35
Пример 3.11. Толщина обводки	36
Пример 3.12. Разные законцовки линий.....	36

Пример 3.13. Разные соединения линий.....	37
3.5. Режимы рисования	37
Пример 3.14. Отсчет координат от угла.....	37
3.6. Использование цвета.....	38
Пример 3.15. Рисование в градациях серого	38
Пример 3.16. Управление заливкой и обводкой	39
Пример 3.17. Рисование в цвете	39
Пример 3.18. Использование прозрачности.....	42
3.7. Пользовательские фигуры.....	42
Пример 3.19. Рисование стрелки.....	42
Пример 3.20. Устранение разрыва.....	43
Пример 3.21. Небольшое развлечение.....	44
3.8. Комментарии.....	45
3.9. Робот 1: рисование.....	46
Глава 4. Переменные	48
4.1. Первые переменные	48
Пример 4.1. Повторное использование одних и тех же значений	48
Пример 4.2. Изменение значений.....	49
4.2. Создание переменных.....	49
4.3. Специальные переменные Processing	50
Пример 4.3. Рисование фигур относительно границ окна	51
4.4. Немного математики.....	51
Пример 4.4. Основные арифметические операции.....	51
4.5. Повторение	53
Пример 4.5. Повторение однотипного действия	53
Пример 4.6. Использование цикла for	54
Пример 4.7. Преимущество использования цикла for	55
Пример 4.8. Линии с разным наклоном.....	56
Пример 4.9. Ломаные линии	56
Пример 4.10. Вложенные циклы.....	56
Пример 4.11. Строки и столбцы.....	57
Пример 4.12. Точки и линии.....	58
Пример 4.13. Полутоновые точки.....	58
4.6. Робот 2: переменные	59
Глава 5. Отклик на внешние события.....	61
5.1. Порядок выполнения программы	61
Пример 5.1. Функция draw().....	61
Пример 5.2. Функция setup().....	62
Пример 5.3. Глобальная переменная	63
5.2. Отслеживание действий пользователя.....	63
Пример 5.4. Отслеживание указателя мыши.....	63
Пример 5.5. Точка следует за указателем мыши.....	64
Пример 5.6. Непрерывное рисование.....	64
Пример 5.7. Изменяемая толщина линии	65

Пример 5.8. Реакция с отставанием.....	66
Пример 5.9. Плавные линии с отставанием.....	67
5.3. Нажатие	68
Пример 5.10. Отслеживание нажатия на кнопку мыши	68
Пример 5.11. Реакция на отсутствие нажатия мыши.....	69
Пример 5.12. Распознавание кнопок мыши	70
5.4. Расположение указателя мыши	71
Пример 5.13. Движение объекта к курсору	72
Пример 5.14. Границы круга	73
Пример 5.15. Границы прямоугольника.....	74
5.5. События клавиатуры	76
Пример 5.16. Нажатие клавиши	76
Пример 5.17. Рисование букв	77
Пример 5.18. Проверка нажатия заданных клавиш	78
Пример 5.19. Перемещение с помощью клавиш со стрелками	79
5.6. Сопоставление диапазонов	79
Пример 5.20. Сопоставление значений с диапазоном.....	79
Пример 5.21. Сопоставление значений при помощи функции <code>map()</code>	80
5.7. Робот 3: отклик на воздействие	81
Глава 6. Перемещение, вращение и масштабирование	83
6.1. Перемещение.....	83
Пример 6.1. Перемещение объекта на экране.....	84
Пример 6.2. Множественные перемещения	84
6.2. Вращение.....	85
Пример 6.3. Вращение на переменный угол	85
Пример 6.4. Вращение вокруг собственного центра	86
Пример 6.5. Смещение, затем вращение	87
Пример 6.6. Вращение, затем смещение.....	87
Пример 6.7. Рычаг с шарнирными сочленениями	88
6.3. Масштабирование.....	89
Пример 6.8. Масштабирование изображения	89
Пример 6.9. Сохранение постоянства штрихов	90
6.4. Сохранение и восстановление системы координат.....	91
Пример 6.10. Изолированные преобразования	91
6.5. Робот 4: смещение, вращение, масштабирование.....	92
Глава 7. Медиафайлы.....	94
7.1. Изображения	95
Пример 7.1. Загрузка изображения	95
Пример 7.2. Загрузка нескольких изображений.....	95
Пример 7.3. Наведение курсора на изображение.....	96
Пример 7.4. Прозрачность в изображениях GIF	97
Пример 7.5. Прозрачность в изображениях PNG	98
7.2. Шрифты.....	98
Пример 7.6. Использование шрифтов.....	99

Пример 7.7. Рисование текста в рамке	100
Пример 7.8. Сохранение текста в строке	100
7.3. Векторные фигуры	101
Пример 7.9. Рисование готовыми фигурами	101
Пример 7.10. Масштабирование фигур	102
Пример 7.11. Создание новой векторной фигуры	103
7.4. Робот 5: медиафайлы	104
Глава 8. Движение	106
8.1. Кадры	106
Пример 8.1. Проверка частоты кадров	106
Пример 8.2. Установка частоты кадров	106
8.2. Скорость и направление	107
Пример 8.3. Перемещение фигуры	107
Пример 8.4. Замкнутое движение	108
Пример 8.5. Отскок от стены.....	109
8.3. Анимация	110
Пример 8.6. Расчет позиций анимации	110
8.4. Случайное движение	111
Пример 8.7. Генерация случайных значений.....	111
Пример 8.8. Рисование случайных объектов.....	111
Пример 8.9. Произвольное перемещение фигур	112
8.5. Таймеры.....	113
Пример 8.10. Течение времени.....	113
Пример 8.11. Запуск событий по времени	114
8.6. Круговое движение	114
Пример 8.12. Значения синусоидальной функции.....	116
Пример 8.13. Движение синусоидальной волны	116
Пример 8.14. Круговое движение	117
Пример 8.15. Движение по спирали.....	117
8.7. Робот 6: движение	118
Глава 9. Функции.....	120
9.1. Основы работы с функциями	120
Пример 9.1. Бросание игровых кубиков.....	121
Пример 9.2. Другой способ бросить кубик.....	121
9.2. Создание функции	122
Пример 9.3. Рисование совы	122
Пример 9.4. Компания из двух сов.....	123
Пример 9.5. Функция рисования совы	125
Пример 9.6. Увеличение популяции сов.....	126
Пример 9.7. Совы разного размера	127
9.3. Возвращаемые значения	128
Пример 9.8. Вычисления в функции	128
9.4. Робот 7: функции.....	129

Глава 10. Объекты	131
10.1. Поля и методы	131
10.2. Определение класса	132
10.3. Создание объектов	136
Пример 10.1. Создание объекта	136
Пример 10.2. Создание нескольких объектов	138
10.4. Вкладки	139
10.5. Робот 8: объекты	140
Глава 11. Массивы	142
11.1. От переменных к массивам	142
Пример 11.1. Множество переменных	142
Пример 11.2. Когда переменных слишком много	143
Пример 11.3. Массивы вместо переменных	144
11.2. Создание массива	144
Пример 11.4. Объявление массива и присвоение значений	146
Пример 11.5. Более компактное создание массива	146
Пример 11.6. Создание массива одной операцией	146
Пример 11.7. Возвращаясь к первому примеру	147
11.3. Повторение и массивы	147
Пример 11.8. Заполнение массива при помощи цикла for	147
Пример 11.9. Отслеживание перемещений мыши	148
11.4. Массивы объектов	150
Пример 11.10. Работа с множеством объектов	150
Пример 11.11. Новый способ управления объектами	151
Пример 11.12. Последовательности изображений	151
11.5. Робот 9: массивы	153
Глава 12. Данные	155
12.1. Что мы знаем о данных	155
12.2. Таблицы	156
Пример 12.1. Чтение таблицы	157
Пример 12.2. Визуализация табличных данных	158
Пример 12.3. 29 740 городов	159
12.3. Данные в формате JSON	160
Пример 12.4. Чтение файла JSON	162
Пример 12.5. Визуализация данных из файла JSON	162
12.4. Сетевые данные и API	164
Пример 12.6. Анализ погодных данных	166
Пример 12.7. Объединение методов JSON	167
12.5. Робот 10: данные	168
Глава 13. Дополнительные возможности языка Processing	171
13.1. Звук	172
Пример 13.1. Воспроизведение звукового фрагмента	172
Пример 13.2. Прослушивание микрофона	173

Пример 13.3. Генерирование синусоидальной волны.....	175
13.2. Экспорт изображений и PDF	176
Пример 13.4. Сохранение изображений.....	177
Пример 13.5. Рисование в PDF	178
13.3. Привет, Ардуино.....	179
Пример 13.6. Считывание показаний датчика.....	180
Пример 13.7. Чтение данных из последовательного порта.....	181
Пример 13.8. Визуализация потока данных	182
Пример 13.9. Еще один способ визуализации данных.....	184
Приложение 1. Рекомендации по программированию.....	186
Приложение 2. Типы данных.....	190
Приложение 3. Порядок операций	191
Приложение 4. Область видимости переменных.....	192
Предметный указатель.....	193

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com на странице с описанием соответствующей книги.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Maker Media очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Предисловие

Мы создали Processing, чтобы упростить программирование интерактивной графики. Мы были разочарованы тем, насколько сложно было написать этот тип программного обеспечения на языках программирования, которые мы обычно использовали (Java и C++), и были вдохновлены тем, насколько просто было писать интересные программы на языках нашего детства (Logo и BASIC). На нас больше всего повлиял созданный нашим научным руководителем Джоном Маэдой¹ язык Design By Numbers (DBN), который мы поддерживали и преподавали в то время.

Processing родился весной 2001 года как результат мозгового штурма на листке бумаги. Нашей целью было создание прототипа такого типа программного обеспечения, которое почти всегда было полноэкранным и интерактивным. Мы искали лучший способ легко проверить наши идеи в коде, а не просто говорить о них или тратить слишком много времени на их программирование на C++. Другой нашей целью было создать язык для обучения студентов-дизайнеров искусству программирования и в то же время дать большему количеству технических студентов простой способ работы с графикой. Эта комбинация отличается от того, как обычно преподают программирование. Мы решили, что нужно начинать с графики и взаимодействия, а не структур данных и вывода текстовой консоли.

Processing пережил долгое детство; это была альфа-версия с августа 2002 года по апрель 2005 года, а затем публичная бета-версия до ноября 2008 года. В течение этого времени его постоянно использовали в учебных классах тысячи людей по всему миру. При этом сам язык, среда разработки и учебные программы постоянно пересматривались. Многие из наших первоначальных решений относительно языка были подтверждены, а многие изменены или отвергнуты. Мы разработали систему программных расширений, называемых библиотеками, которые позволили людям применить Processing во многих неожиданных и удивительных проектах. (Сейчас существует более 100 библиотек.)

Осенью 2008 года мы запустили версию 1.0 программы. После семи лет работы запуск 1.0 означал долгожданную стабильность. Весной 2013 года мы выпустили версию 2.0, работающую заметно быстрее. В этой версии улучшена интеграция с OpenGL, шейдерами GLSL и ускорено воспроизведение видео с помощью GStreamer. Выпущенная в 2015 году версия 3.0 облегчает программирование в Processing благодаря новому интерфейсу и проверке ошибок.

Теперь, спустя четырнадцать лет после своего появления, Processing вышел за рамки своего первоначального предназначения, и мы узнали, как он может быть полезен в других контекстах. Соответственно, эта книга написа-

¹ Тот самый Джон Маэда – всемирно известный гуру дизайна, идеолог максимальной простоты, лаконичности и удобства в дизайне и технологиях. – *Прим. перев.*

на для новой аудитории – специалистов в других областях, которым нужно программировать время от времени, любителей и всех, кто хочет изучить, на что способен Processing, не теряясь в деталях огромного учебника. Мы надеемся, что вам понравится и вы решите продолжить занятия программированием. Эта книга – только начало.

Хотя мы (Кейси и Бен) ведем корабль Processing по водам последние два десятка лет, мы не можем переоценить тот факт, что Processing – это плод коллективных усилий сообщества. От написания библиотек, расширяющих возможности языка, до публикации кода в интернете и помощи другим в обучении сообщество пользователей Processing продвинуло его далеко за пределы первоначальной концепции. Без этих коллективных усилий Processing не был бы тем, чем он является сегодня.

КРАТКОЕ СОДЕРЖАНИЕ КНИГИ

В главах этой книги раскрыты следующие темы:

- глава 1: знакомство с языком Processing;
- глава 2: первая программа на языке Processing;
- глава 3: определение и рисование простых фигур;
- глава 4: хранение, изменение и повторное использование данных;
- глава 5: управление выполнением программы с помощью мыши и клавиатуры;
- глава 6: преобразование координат;
- глава 7: загрузка и отображение мультимедийных файлов, включая шрифты, растровые и векторные изображения;
- глава 8: перемещение и взаимосвязанное движение фигур;
- глава 9: создание новых модулей кода;
- глава 10: создание модулей кода, сочетающих переменные и функции;
- глава 11: упрощение работы со списками переменных;
- глава 12: загрузка и визуализация данных;
- глава 13: работа с 3D, экспорт PDF, компьютерное зрение и чтение данных с платы Arduino.

КОМУ АДРЕСОВАНА ЭТА КНИГА

Эта книга написана для людей, которые нуждаются в кратком и понятном введении в программирование, чтобы научиться создавать компьютерную графику и простые интерактивные программы. Она адресована читателям, которые хотят разобраться в тысячах примеров бесплатного кода на языке Processing и справочных материалов, доступных в интернете. «Знакомство с программированием на языке Processing» – это не учебник по программированию; это именно первое знакомство с ним. Эта книга предназначена для подростков, любителей технического творчества, их бабушек и дедушек и всех остальных.

Книга также подходит для читателей с опытом программирования, желающих изучить основы интерактивной компьютерной графики. В этой книге они найдут описание методов, которые можно применять для создания игр, анимации и интерфейсов.

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ И СОГЛАШЕНИЯ, ПРИНЯТЫЕ В КНИГЕ

В книге используются следующие типографские соглашения.

Курсив – используется для смыслового выделения важных положений, новых терминов, имен команд и утилит, а также слов и предложений на естественном языке.

Моноширинный шрифт применяется для листингов программ, а также в обычном тексте для обозначения имен переменных, функций, типов, объектов, баз данных, переменных среды, операторов, ключевых слов и других программных конструкций и элементов исходного кода.

Моноширинный курсив применяется для обозначения в исходном коде или в командах шаблонных меток-заполнителей, которые должны быть заменены соответствующими контексту реальными значениями.



Этот элемент обозначает совет, предложение или общее примечание.



Этот элемент указывает на предупреждение или предостережение.

БЛАГОДАРНОСТИ

За первое и второе издания этой книги мы благодарим Брайана Джепсона за его огромную энергию, поддержку и понимание. В первом издании Нэнси Котари, Рэйчел Монаган и Сумита Мукхерджи умело довели книгу до финиша. Том Сгоурос тщательно отредактировал книгу, а Дэвид Хамфри представил содержательный технический обзор.

Мы не можем представить эту книгу без книги Массимо Банци *Getting Started With Arduino*. (В России издана под названием «Arduino для начинающих волшебников».) Превосходная книга Массимо – наш образец для подражания.

Небольшая группа людей годами вкладывала время и энергию в Processing. Дэн Шиффман – наш партнер в Processing Foundation, организации, которая поддерживает программное обеспечение Processing. Большая часть основного кода для Processing 2.0 и 3.0 рождена острыми умами Андреса Колубри и Маниндры Мохараны. Скотт Мюррей, Джейми Косой и Джон Гакник создали замечательную веб-инфраструктуру для этого проекта. Джеймс Грэди развивает пользовательский интерфейс версии 3.0. Мы благодарим Флориана Дженетта за его годы разнообразной работы над проектом, включая

форумы, веб-сайт и дизайн. Эли Зананири и Андреас Шлегель разработали инфраструктуру для подключения и документирования сторонних библиотек и потратили бесчисленные часы на составление списков. Многие пользователи внесли значительный вклад в проект; точные данные доступны по адресу <https://github.com/processing>.

Релиз Processing 1.0 был поддержан Университетом Майами и Oblong Industries. Институт интерактивных медиаисследований Армстронга при Университете Майами профинансировал Оксфордский проект – серию семинаров по разработке программ на языке Processing. Эти семинары стали возможными благодаря упорному труду Иры Гринберг. Четырехдневные конференции в Оксфорде, штат Огайо, и Питтсбурге, штат Пенсильвания, позволили в ноябре 2008 года запустить Processing 1.0. Летом 2008 года компания Oblong Industries профинансировала Бену Фрау разработку проекта Processing; это было важно для выпуска стабильной версии.

Выпуску Processing 2.0 способствовал семинар по развитию, спонсируемый Программой интерактивных телекоммуникаций Нью-Йоркского университета. Работа над Processing 3.0 была щедро спонсирована программой Emergent Digital Practices в Университете Денвера. Мы благодарим Кристофера Колемена и Лале Мехран за неоценимую поддержку.

Эта книга выросла из лекций по языку Processing в Калифорнийском университете в Лос-Анджелесе. Чендлер МакВильямс сыграл важную роль в создании учебного курса. Кейси благодарит студентов факультета дизайна и медиаискусства Калифорнийского университета в Лос-Анджелесе за их энергию и энтузиазм. Его ассистенты были отличными соавторами в разработке учебного плана. Снимаю шляпу перед Тацуей Сайто, Джоном Хоуком, Тайлером Адамсом, Аароном Сигелом, Кейси Альт, Андресом Колубри, Майклом Контопулосом, Дэвидом Эллиотом, Христо Аллегрой, Питом Хоуксом и Лорен Маккарти.

Все это стало возможным благодаря тому, что Джон Маэда основал группу компьютерной эстетики (1996–2002) в MIT Media Lab.

Глава 1

Знакомство с языком Processing

Processing предназначен для написания программ, которые позволяют создать анимированное изображение и взаимодействовать с ним. Напишите одну строку кода – и на экране появится кружок. Добавьте еще несколько строк кода – и кружок будет следовать за курсором мыши. Еще одна строка кода – и кружок меняет цвет при нажатии на кнопку мыши. Мы называем эти короткие наброски кода *скетчами*¹. Вы пишете одну строку, к ней добавляете другую и смотрите, что получилось. Затем добавляете новые строки и так далее. В результате программа создается по частям.

Курсы программирования традиционно сосредоточены на структуре и теории выбранного языка. Все, что касается визуального оформления, – интерфейс, анимация – считается десертом, которым можно наслаждаться только после того, как вы съели отварные овощи, обычно спустя несколько недель изучения алгоритмов и методов. На протяжении многих лет мы наблюдали, как наши друзья пытались пройти такие курсы и бросали учебу после первой же лекции или после долгой бесплодной ночи перед крайним сроком выполнения первого задания. Их стремление заставить компьютер выполнять полезную работу быстро угасало, потому что они не видели прямой связи между тем, чему их учили, и тем, что они хотели получить.

Processing предлагает изучать программирование посредством создания интерактивной графики. Существуют разные методики обучения программированию, но для начинающих программистов наилучшей мотивацией чаще всего служит возможность немедленно увидеть на экране эффектный результат. Именно способность языка Processing предоставлять визуальную обратную связь буквально с первой строки кода сделала его столь популярным во всем мире средством обучения программированию. Давайте кратко обсудим основные подходы к программированию на языке Processing.

¹ Sketch – набросок, эскиз. Все, кому доводилось программировать для Arduino, уже знакомы с этим термином. Более того, Arduino и Processing – родные братья, и у них одинаковая среда разработки кода. Поэтому мы будем использовать в этой книге привычный многим читателям термин «скетч», хотя в интерфейсе среды разработки Processing он переведен как «набросок». – *Прим. перев.*

1.1. СКЕТЧИ И ПРОТОТИПЫ

Скетчи – это особый способ мышления; с ними можно программировать весело и быстро. Основная цель – изучить множество идей за короткий промежуток времени. Приступая к разработке программы, мы обычно начинаем с набросков алгоритма на бумаге, а затем переносим свои идеи в код. Идеи анимации и взаимодействия с пользователем обычно набрасываются в виде раскадровки с примечаниями. После создания набросков программного обеспечения лучшие идеи отбираются и объединяются в *прототипы* (рис. 1.1). Это циклический процесс разработки, тестирования и улучшения, когда вы перемещаетесь между бумагой и экраном.

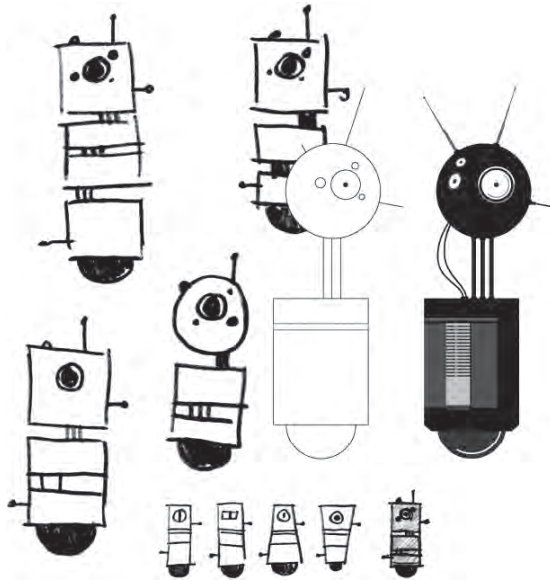


Рис. 1.1 ❖ По мере того как эскизы перемещаются из блокнота на экран, их облик становится все более детализованным

1.2. ГИБКОСТЬ

Подобно ящику со слесарными приспособлениями, Processing состоит из множества инструментов, которые можно использовать в разных комбинациях. В результате его можно использовать как для быстрых набросков кода, так и для серьезного сложного программирования. Программа на языке Processing может состоять из одной строки или тысяч строк, поэтому у вас есть простор для роста и выбора вариантов. Более 100 специализированных библиотек расширяют возможности Processing в таких областях, как звук, компьютерное зрение и генерация цифровых объектов (рис. 1.2).



Рис. 1.2 ❖ Processing может оперировать разнообразными типами данных

1.3. ГИГАНТЫ ПРОШЛОГО

Люди создают изображения с помощью компьютеров с 1960-х годов, и в истории компьютерной графики можно найти много интересного. Например, до того, как компьютеры смогли отображать изображения на ЭЛТ или ЖК-экранах, для рисования изображений применялись огромные плоттеры (рис. 1.3). В жизни мы все в том или ином смысле стоим на плечах гигантов, а среди гигантов обработки данных есть мыслители из области дизайна, компьютерной графики, искусства, архитектуры, статистики и смежных областей. Взгляните, например, на альбом Ивана Сазерленда *Sketchpad* (1963), *Dynabook* Алана Кея (1968) и работы многих художников, представленных в книге Рут Левитт «Художник и компьютер» (*Artist and Computer*, Harmony Books, 1976). Архивы ACM SIGGRAPH и Ars Electronica позволяют заглянуть в историю графики и программного обеспечения.

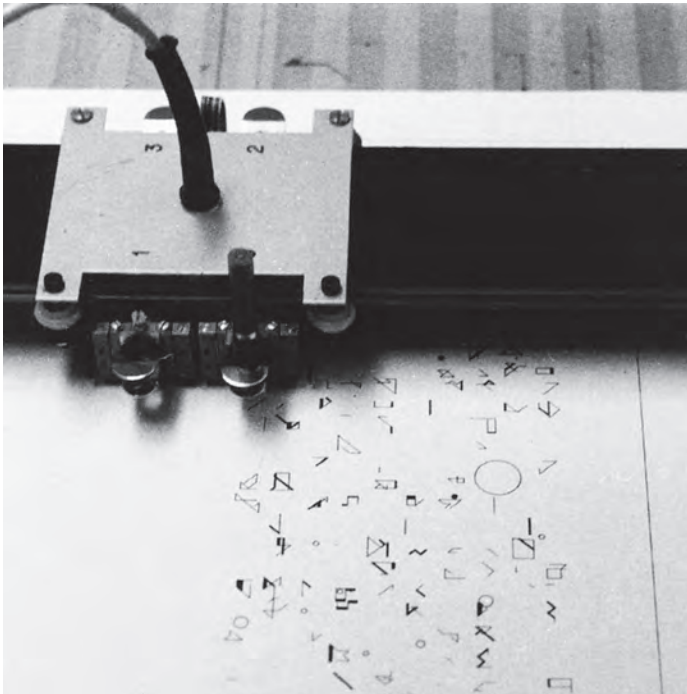


Рис. 1.3 ❖ Демонстрационный рисунок Манфреда Мора, представленный в Музее современного искусства в Париже, выполнен с использованием плоттера Venson и компьютера 11 мая 1971 г. (фото Райнера Мюрле любезно предоставлено галереей *bitforms*, Нью-Йорк)

1.4. ГЕНЕАЛОГИЧЕСКОЕ ДРЕВО ЯЗЫКОВ

Языки программирования, как и человеческие языки, образуют семейства родственных языков. Processing – это диалект языка программирования Java; синтаксис этих языков почти идентичен, но Processing добавляет специальные функции, связанные с графикой и визуальным взаимодействием (рис. 1.4). Графические элементы Processing связаны с PostScript (основа PDF) и OpenGL (спецификация трехмерной графики). Благодаря этим родственным связям изучение языка Processing – это первый шаг к программированию на других языках и с использованием различных программных инструментов.

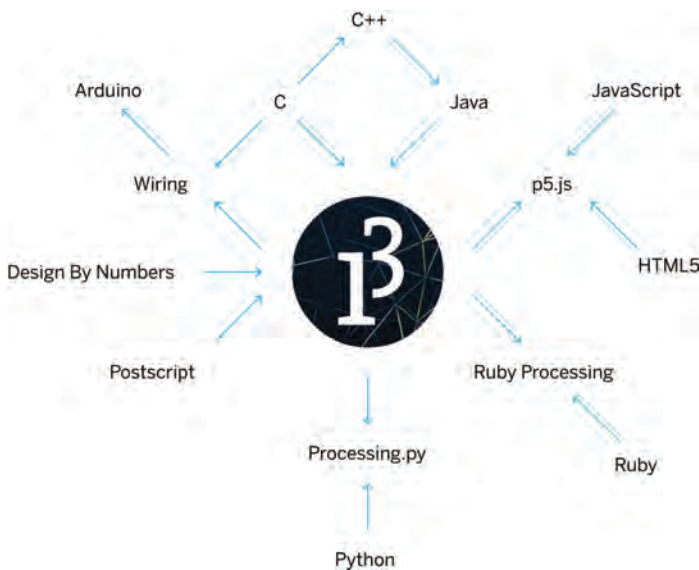


Рис. 1.4 ❖ Демонстрация рисунка Манфреда Мора в Musée d'Art Moderne de la Ville de Paris с использованием плоттера Benson и цифрового компьютера 11 мая 1971 г. (фото Райнера Мюрле, любезно предоставлено галереей bitforms, Нью-Йорк)

1.5. ПРИСОЕДИНЯЙТЕСЬ К СООБЩЕСТВУ!

Тысячи людей по всему миру используют Processing каждый день. Как и они, вы можете бесплатно скачать Processing. У вас даже есть возможность изменить исходный код Processing в соответствии с вашими потребностями. Processing – это бесплатное программное обеспечение с открытым исходным кодом, и мы призываем вас участвовать в жизни сообщества, делаясь своими проектами и знаниями в интернете на сайте Processing.org и на различных сайтах социальных сетей, где есть группы, посвященные языку Processing. Ссылки можно найти на сайте Processing.org.

Глава 2

Начинаем программировать

Чтобы извлечь максимальную пользу из этой книги, недостаточно просто ее прочитать. Вам нужно экспериментировать и практиковаться. Вы не можете научиться программировать, просто читая об этом, – вам нужно это делать. Для начала скачайте Processing и напишите свой первый скетч.

Первым делом зайдите на сайт <http://processing.org/download> и выберите версию для Mac, Windows или Linux, в зависимости от того, какой у вас компьютер¹. Процедура установки среды разработки Processing чрезвычайно проста:

- версия для Windows распространяется в виде архивного файла *.zip*. Дважды щелкните по нему и перетащите расположенную внутри папку в любое место на жестком диске. Это может быть папка Program Files или просто рабочий стол, важно только, чтобы папка с файлами Processing была извлечена из скачанного архива. Затем дважды щелкните по файлу *processing.exe*, чтобы запустить Processing;
- версия для Mac OS X – это тоже файл *.zip*. Дважды щелкните по нему и перетащите значок *Processing* в папку *Applications* (Приложения). Если вы используете чужой компьютер и не можете изменить эту папку, просто перетащите приложение на рабочий стол. Затем дважды щелкните по иконке *Processing*, чтобы запустить Processing;
- версия для Linux представляет собой файл *.tar.gz*, который должен быть знаком большинству пользователей Linux. Загрузите файл в домашний каталог, затем откройте окно терминала и введите:

```
tar xvfz processing-xxxx.tgz
```

(Замените символы xxxx номером версии.) Будет создан каталог с именем `processing-3.5.4` или наподобие этого. Затем перейдите в этот каталог:

```
cd processing-xxxx
```

и запустите Processing:

```
./processing
```

¹ На момент работы над переводом была доступна версия Processing 3.5.4, поэтому все снимки экрана и пункты меню будут представлены для этой версии. – *Прим. перев.*

Если вы все сделали правильно, то увидите главное окно среды разработки Processing (рис. 2.1). Если же программа не запустилась или возникла какая-то другая проблема, прочитайте об устранении неполадок по адресу <http://bit.ly/process-wiki/>.

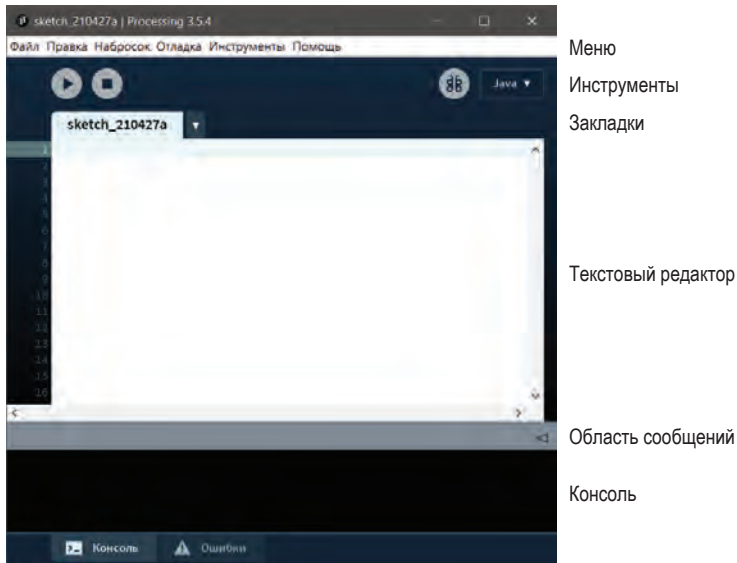


Рис. 2.1 ❖ Среда разработки Processing

2.1. ВАША ПЕРВАЯ ПРОГРАММА

Итак, у вас запущена среда разработки Processing (или Processing development environment, PDE). В ней нет ничего особенного; большая область – это *текстовый редактор*, а в верхней части расположены две кнопки; это *панель инструментов*. Под редактором находится *область сообщений*, а еще ниже – *консоль*. Область сообщений используется для вывода однострочных сообщений, а консоль используется для многострочного текстового вывода.

Пример 2.1. Рисование эллипса

В редакторе введите следующий текст:

```
ellipse(50, 50, 80, 80);
```

Эта строка кода означает «нарисовать эллипс с центром, расположенным на 50 пикселей вправо и на 50 пикселей вниз относительно верхнего левого угла, с шириной и высотой 80 пикселей». Нажмите кнопку **Запустить** (кнопка с треугольником на панели инструментов).

Если вы ввели код правильно, на экране появится кружок. Если вы ввели его неправильно, область сообщений станет красной и появится сообщение об ошибке. В этом случае убедитесь, что вы точно скопировали пример кода: числа должны быть заключены в круглые скобки и разделены запятыми, а строка должна заканчиваться точкой с запятой.

Одна из самых сложных вещей, связанных с началом программирования, – это то, что вы должны очень точно соблюдать требования синтаксиса. Processing, как и другие инструменты программирования, не настолько умный, чтобы догадываться, что вы имеете в виду, и может быть очень придирчивым к расстановке знаков препинания. Вы быстро привыкнете к этому после небольшой тренировки.

Далее мы перейдем к более интересному скетчу.

Пример 2.2. Рисование кругов

Удалите из окна редактора текст предыдущего примера и введите новый:



```
void setup() {
  size(480, 120);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Эта программа создает рабочее окно скетча шириной 480 пикселей и высотой 120 пикселей, а затем начинает рисовать в нем белые круги с центром в том месте, куда указывает курсор мыши. При нажатии кнопки мыши цвет круга меняется на черный. Подробнее об этой программе мы расскажем позже. А пока запустите код, перемещайте указатель мыши над рабочим окном и периодически нажимайте левую кнопку мыши, чтобы увидеть, что будет происходить. Справа от кнопки **Запустить** на панели инструментов находится кнопка **Остановить** со значком в виде квадрата, на который вы можете на-

жать, чтобы остановить выполнение скетча. Вы можете также закрыть рабочее окно скетча, как вы это обычно делаете с любым другим приложением.

2.2. РЕЖИМ ОТОБРАЖЕНИЯ РАБОЧЕГО ОКНА

Если вы не хотите использовать кнопки на панели инструментов, вы всегда можете обратиться к меню **Набросок** (Скетч), в котором есть сочетание клавиш **Ctrl-R** (или **Cmd-R** на Mac) для запуска скетча. Опция **Режим презентации** перед запуском скетча очищает экран, чтобы на нем осталось только рабочее окно скетча. Вы также можете перейти в **Режим презентации**, удерживая нажатой клавишу **Shift** при нажатии кнопки **Запустить**.

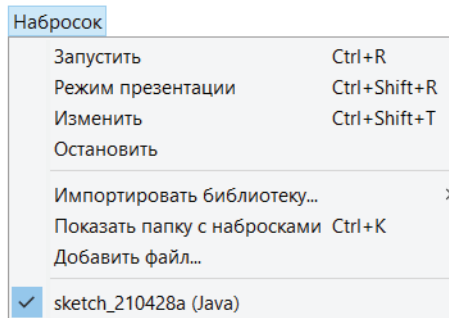


Рис. 2.2 ❖ Скетч можно выполнить при помощи опций **Запустить** и **Режим презентации**.
В режиме презентации экран полностью очищается перед запуском скетча

2.3. СОЗДАНИЕ И СОХРАНЕНИЕ НОВОГО СКЕТЧА

Следующая важная команда – **Сохранить**. Вы можете найти ее в меню **Файл**. По умолчанию ваши программы сохраняются в папке Processing, внутри которой при сохранении скетча автоматически создается вложенная папка, название которой совпадает с названием скетча. Выберите пункт **Рабочая папка** в меню **Файл**, чтобы открыть список всех скетчей в вашей рабочей папке.

Всегда полезно почаще сохранять свои программы. Пробуя разные изменения кода, сохраняйте их под разными именами, чтобы всегда можно было вернуться к более ранней версии. Это особенно полезно, если... нет, когда что-то пойдет не так. Вы также можете увидеть, где находится скетч на вашем компьютере, с помощью команды **Показать папку с набросками** в меню **Наброски**.

Вы можете создать новый скетч, выбрав опцию **Создать** в меню **Файл**. Будет создан новый пустой скетч в отдельном окне.

2.4. РАСПРОСТРАНЕНИЕ ПРОГРАММЫ

Написав программу, вы можете поделиться ей с другими пользователями. Опция **Экспорт приложения** в меню **Файл** объединит ваш код в одну папку, которой вы сможете поделиться с другими пользователями. Можно создать приложения для Mac OS X, Windows и/или Linux по вашему выбору. Это простой способ сделать автономные версии ваших проектов, которые можно запускать двойным щелчком по имени файла в полноэкранный режиме или в окне. Если вы не уверены, что на компьютерах пользователей, с которыми вы делитесь приложением, установлена среда выполнения Java SE, установите в окне **Настройка экспорта** галочку **Встроить Java для Windows**.

Папки приложения удаляются и создаются заново каждый раз, когда вы используете команду **Экспорт приложения**, поэтому обязательно переместите готовую папку в другое место, если вы не хотите, чтобы она была удалена при следующем экспорте.

2.5. ПРИМЕРЫ И ССЫЛКИ

Чтобы научиться программировать, нужно пропустить через свои руки большой объем кода: запускать, изменять, портить и исправлять его до тех пор, пока у вас не получится что-то новое. Именно поэтому доступные для скачивания файлы готовых скетчей Processing включают в себя десятки примеров, демонстрирующих различные опции этого языка.

Чтобы открыть пример, выберите опцию **Примеры** в меню **Файл** и дважды щелкните по имени примера. Примеры сгруппированы по категориям в зависимости от их функции, например **Форма**, **Движение** и **Изображение**. Найдите в списке интересную тему и попробуйте запустить скетч.



Все примеры из этой книги можно загрузить и запустить непосредственно из среды разработки Processing. Выберите опцию **Примеры** в меню **Файл**, затем нажмите кнопку **Добавить...**, чтобы открыть список пакетов с примерами, доступных для загрузки. Выберите пакет **Getting Started with Processing, 2nd Edition** и нажмите кнопку **Install** (Установить).

Взглянув на код в редакторе, вы увидите, что такие функции, как `ellipse()` и `fill()`, окрашены в другой цвет на фоне остального текста. Если вы видите незнакомую функцию, выделите окрашенный текст и нажмите опцию **Найти в документации** в меню **Помощь**. Вы также можете щелкнуть по тексту правой кнопкой мыши (или при нажатой клавише **Ctrl** на Mac) и выбрать опцию **Найти в документации** в появившемся контекстном меню. После этого в веб-браузере откроется страница с описанием функции. Кроме того, вы можете просмотреть полную документацию, выбрав опцию **Документация** в меню **Помощь**.

Справочник по языку Processing объясняет каждый элемент кода с описанием и примерами. Программы в справочнике очень короткие (обычно

четыре или пять строк), и их намного проще понять, чем более длинный код примеров. Мы рекомендуем держать справочник открытым во время чтения этой книги и программирования. Вы можете перемещаться в справочнике по темам или в алфавитном порядке; иногда быстрее всего выполнить текстовый поиск в окне браузера.

Справочник был написан для новичков; мы надеемся, что сделали его ясным и понятным. Мы благодарны множеству людей, которые все прошедшие годы замечали ошибки и сообщали о них. Если вы думаете, что можете улучшить статью в справочнике или обнаружили ошибку, сообщите нам об этом, перейдя по ссылке вверху каждой справочной страницы.