O'REILLY®

Четвертое издание

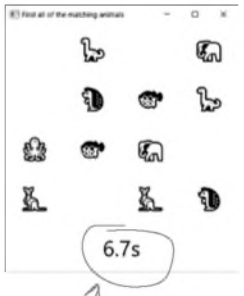
Head First

Изучаем C#

Эндрю Стиллмен
Дженнифер Грин

ПОДАРОК ДЛЯ МОЗГА

Содержание (сводка)

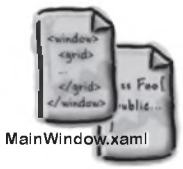


Сделаем игру чуть более азартной! В нижней части окна выводится время, прошедшее с момента запуска игры. Показания таймера постоянно увеличиваются, а останавливается таймер только после нахождения последней пары.

Содержание (настоящее)

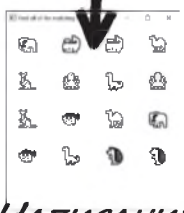
Введение

Ваш мозг и C#. Вы учитесь — готовитесь к экзамену. Или пытаетесь освоить сложную техническую тему. Ваш мозг хочет оказать вам услугу. Он старается сделать так, чтобы на эту очевидно несущественную информацию не тратились драгоценные ресурсы. Их лучше потратить на что-нибудь важное. Так как же заставить его изучить C#?

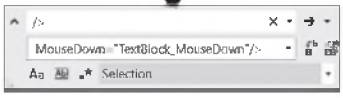


Создание проекта

Конструирование окна



Написание кода C#



Обработка щелчков

Добавление таймера

1 Начало работы с C#

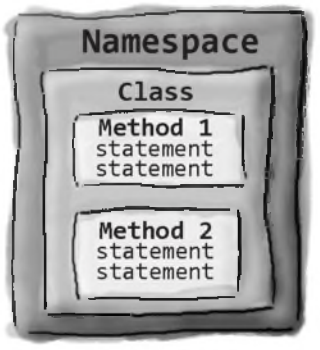
Быстро сделать что-то классное!

Хотите программировать быстро? C# — это мощный язык программирования. Благодаря Visual Studio вам не потребуется писать непонятный код, чтобы заставить кнопку работать. Вместо того чтобы запоминать параметры метода для имени и для ярлыка кнопки, вы сможете создать действительно классное приложение. Звучит заманчиво? Тогда переверните страницу и приступим к делу.

2 Погружение в C#

Команды, классы и код

Вы не просто пользователь IDE. Вы — разработчик. IDE может сделать за вас очень многое, и все же ее возможности небезграничны. Visual Studio — одна из самых совершенных систем разработки программного обеспечения, однако мощная IDE — только начало. Пришло время заняться углубленным изучением кода C#: какую структуру он имеет, как он работает, как управлять им… Потому что нет предела тому, что вы можете делать в ваших приложениях.

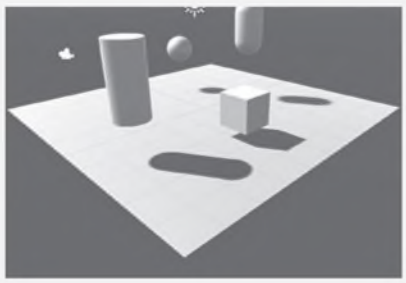


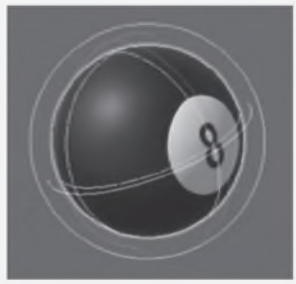
Лабораторный курс Unity № 1

Исследование C# с Unity

Добро пожаловать на первый урок **«Лабораторный курс Unity»**. Написание кода — навык, и, как и любой другой навык, он развивается за счет **практики** и **экспериментирования**. И в этом отношении Unity может стать очень полезным инструментом.

В первой лабораторной работе мы введем вас в курс дела. Вы начнете ориентироваться в редакторе Unity, а также создавать 3D-объекты и оперировать ими.

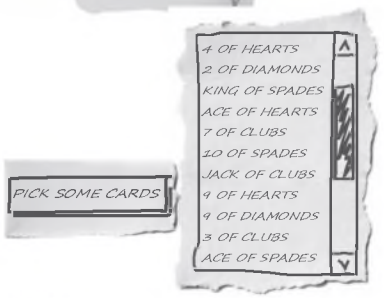


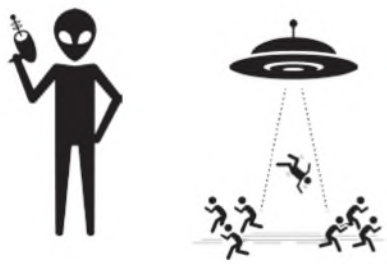


3 Ориентируемся на объекты

Написание осмысленного кода

Каждая написанная вами программа решает некоторую задачу. Когда вы пишете программу, всегда желательно заранее подумать, какую задачу должна решать ваша программа. Вот почему объекты приносят такую пользу. Они позволяют сформировать структуру кода в соответствии с решаемой задачей, чтобы вы могли тратить время на задачу, над которой работаете, не отвлекаясь на механику написания кода. Если вы правильно используете объекты (и действительно хорошо продумали их при проектировании), получившийся код будет интуитивно понятным, будет легко читаться и изменяться.



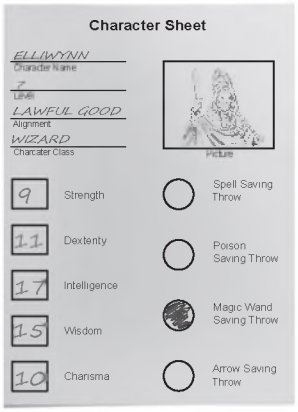


4

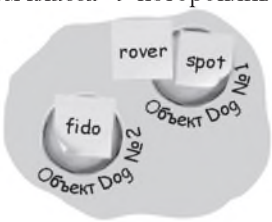
Типы и ссылки

Данные и ссылки

Чем были бы наши приложения без данных? Задумайтесь на минуту. Без данных наши программы... в общем, трудно представить, что кто-то станет писать код без данных. Вы запрашиваете информацию у ваших пользователей; эта информация используется для поиска данных или генерирования новой информации, которая возвращается пользователю. Собственно, практически все, что вы делаете в программировании, требует работы с данными в той или иной форме. В этой главе вы узнаете все тонкости типов данных и ссылок C#, узнаете, как работать с данными в программах, и даже узнаете кое-что новое об объектах (представьте, объекты — тоже данные!).



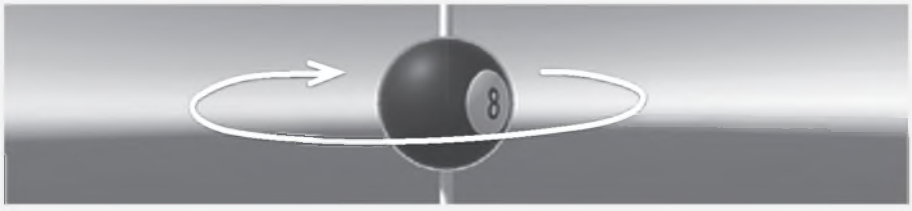
Создание ссылки выглядит так, словно вы пишете имя на наклейке и прикрепляете ее к объекту. Надпись становится своего рода «меткой», по которой вы можете обращаться к объекту в будущем.



Лабораторный курс Unity № 2

Написание кода C# для Unity

Unity — не только мощный кросс-платформенный движок и редактор для построения 2D- и 3D-игр и моделирования. Также это **отличный способ потренироваться в написании кода C#.** В этой лабораторной работе мы начнем писать код для управления объектами GameObject.



5 Инкапсуляция

Умейте хранить секреты

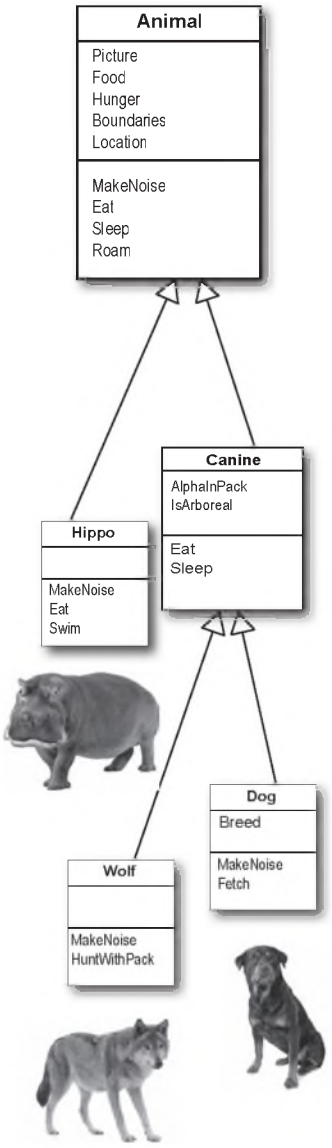
Вам когда-нибудь хотелось, чтобы посторонние не лезли в ваши личные дела? Вот и вашим объектам этого иногда хочется. И если вы не желаете, чтобы чужие люди читали ваш дневник или просматривали банковские выписки, хорошие объекты не позволяют другим объектам копаться в их полях. В этой главе вы узнаете о мощи инкапсуляции — приеме программирования, который делает ваш код более гибким. Такой код проще использовать и его труднее использовать некорректно. Данные вашего объекта объявляются приватными, и к ним добавляются свойства, защищающие обращения к этим данным.


```
RealName: "Herb Jones"
Alias: "Dash Martin"
Password: "the crow flies at midnight"
```

6 Наследование

Генеалогическое древо объектов

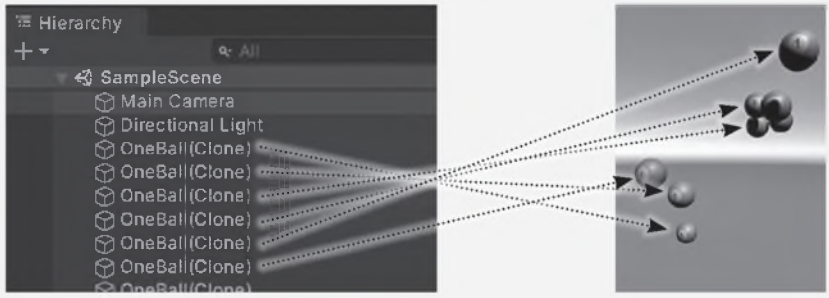
Иногда люди ХОТЯТ быть похожими на своих родителей. Вы встречали объект, который действует почти так, как нужно? Думали ли вы о том, что при изменении всего нескольких элементов класс стал бы идеальным? Наследование позволяет расширять существующие классы, чтобы новый класс получал все поведение существующего — сохраняя при этом гибкость для внесения изменений, чтобы класс можно было адаптировать под любые конкретные требования. Наследование является одним из самых мощных инструментов C#: в частности, оно помогает избегать дублирования кода, более адекватно моделировать реальный мир и в конечном итоге упрощает их сопровождение и снижает риск ошибок.



Лабораторный курс Unity № 3

Экземпляры GameObject

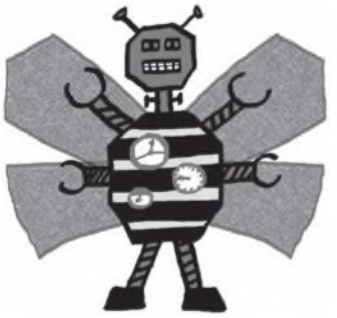
Unity — не только мощный кросс-платформенный движок и редактор для построения 2D- и 3D-игр и моделирования. Также это **отличный способ потренироваться в написании кода C#.** В этой лабораторной работе мы начнем писать код для управления объектами GameObject.



7 Интерфейсы, приведение типов и is

Классы должны держать обещания

Вам нужен объект для выполнения конкретной задачи? Используйте интерфейс. Иногда возникает необходимость сгруппировать объекты по выполняемым ими функциям, а не по классам, от которых они наследуют. На помощь приходят интерфейсы. Интерфейсы могут использоваться для определения конкретных задач. Любой экземпляр класса, реализующего интерфейс, гарантированно выполняет эту задачу независимо от того, с какими другими классами он связан. Чтобы эта схема работала, каждый класс, реализующий интерфейс, должен гарантировать выполнение всех своих обязательств… иначе программа компилироваться не будет.



8

Перечисления и коллекции

Организация данных

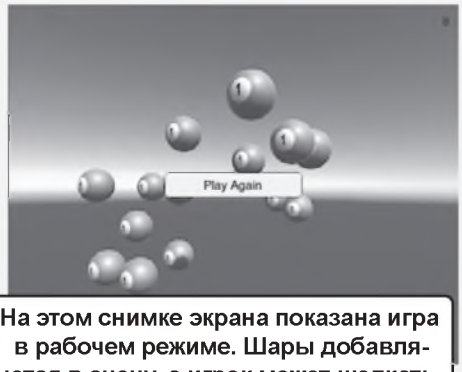
Данные не всегда бывают такими аккуратными и ухоженными, как нам хотелось бы. В реальном мире данные, как правило, не хранятся маленькими аккуратными кусочками. Нет, данные поступают вагонами, штабелями и кучами. Для их систематизации нужны мощные инструменты, и тут вам на помощь приходят перечисления и коллекции. Перечисления — типы, позволяющие определять значения для классификации ваших данных. Коллекции — специальные объекты, способные хранить и сортировать данные, которые обрабатывает программа, и управлять ими. В результате вы можете сосредоточиться на основной идее программирования, оставив задачу управления данных коллекциям.

Карта «Герцог быков». В природе не встречается.

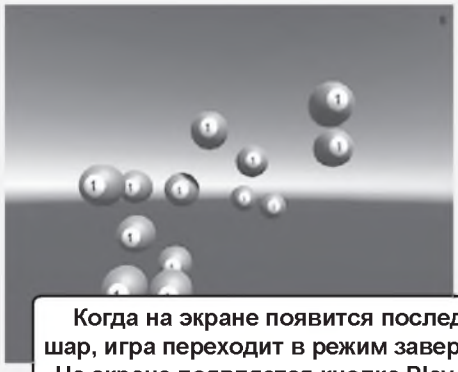
Лабораторный курс Unity № 4

Пользовательские интерфейсы

В предыдущей лабораторной работе Unity вы начали строить игру. Мы использовали заготовку для создания экземпляров GameObject, которые появлялись в случайных точках трехмерного пространства игры и летали по кругу. В этой лабораторной работе мы продолжим с того места, на котором остановились в предыдущей главе; в ней вы сможете применить то, что узнали об интерфейсах C#, и многое другое.



На этом снимке экрана показана игра в рабочем режиме. Шары добавляются в сцену, а игрок может щелкать на них, чтобы получать очки.



Когда на экране появится последний шар, игра переходит в режим завершения. На экране появляется кнопка Play Again, и новые шары перестают появляться.

9 LINQ и лямбда-выражения

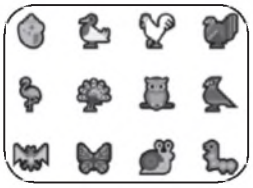
Контроль над данными

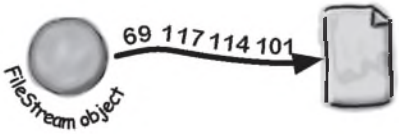
Этим миром правят данные... И нам нужно знать, как в нем жить. Прошли те времена, когда можно было программировать днями и даже неделями, не имея дела с огромными объемами данных. В наши дни данные стали сутью любой программы. LINQ – технология C# и .NET, которая позволяет не только обращаться с запросами к данным в коллекциях .NET на интуитивно понятном уровне, но и группировать данные и выполнять слияние данных из разных источников. Модульные тесты помогут убедиться в том, что ваш код работает так, как предполагалось. А когда вы освоитесь с задачей разбиения данных на блоки, с которыми удобно работать, вы можете воспользоваться лямбда-выражениями, провести рефакторинг кода C# и сделать его еще более выразительным.

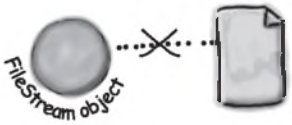
10 Чтение и запись файлов

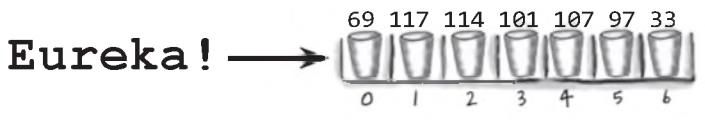
Прибереги последний байт для меня

Иногда настойчивость окупается. Пока что все ваши программы жили недолго. Они запускались, некоторое время работали и закрывались. Но этого недостаточно, когда имеешь дело с важной информацией. Вы должны уметь сохранять свою работу. В этой главе мы поговорим о том, как записать данные в файл, а затем о том, как прочитать эту информацию. Вы познакомитесь с потоками данных, узнаете о сохранении объектов в файлах с использованием сериализации, а также освоите работу с шестнадцатеричными и двоичными данными и кодировку Юникод.









Лабораторный курс Unity № 5

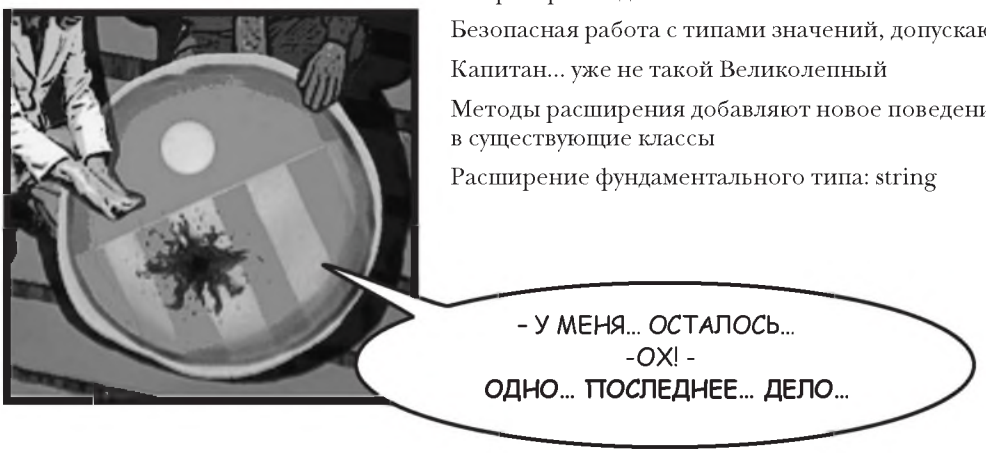
Отслеживание лучей

Создавая сцену в Unity, вы создаете виртуальный 3D-мир, в котором перемещаются персонажи вашей игры. Но в большинстве игр объекты окружающей обстановки не контролируются игроком напрямую. Как же эти объекты определяют свое место в сцене? В этой лабораторной работе мы построим сцену из объектов GameObject и используем навигацию для перемещения персонажей по сцене.

Капитан Великолепный

Смерть объекта

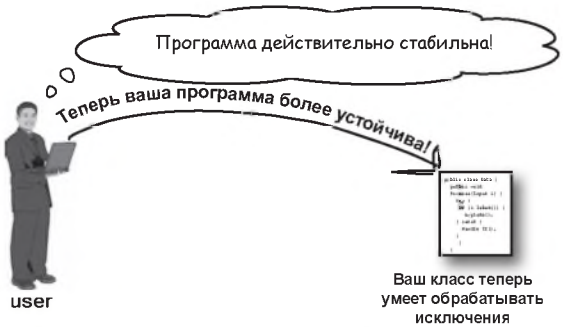
Head First C#	
Четыре доллара	Глава 11



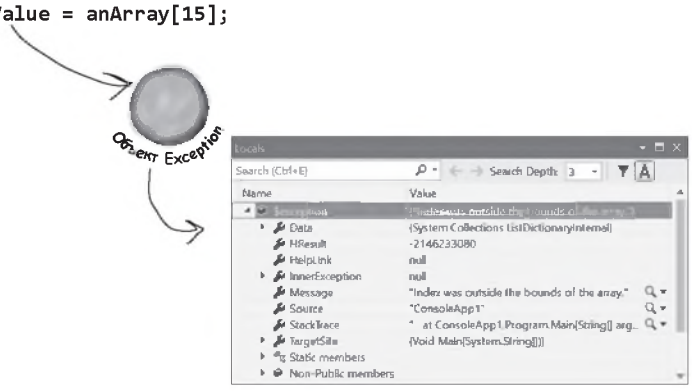
12 Обработка исключений

Борьба с огнем надоедает

Программисты не должны уподобляться пожарным. Вы усердно работали, штудировали справочники и руководства и, наконец, достигли вершины. Но вам до сих пор продолжают звонить с работы по ночам, потому что программа упала или работает не так, как должна работать. Ничто так не выбивает из колеи, как необходимость устранять странные ошибки... но благодаря обработке исключений вы сможете написать код, который сам будет разбираться с возможными проблемами. А еще лучше, что вы можете планировать такие проблемы и восстанавливать работоспособность программы при их возникновении.




```
int[] anArray = {3, 4, 1, 11};
int aValue = anArray[15];
```



Лабораторный курс Unity № 6

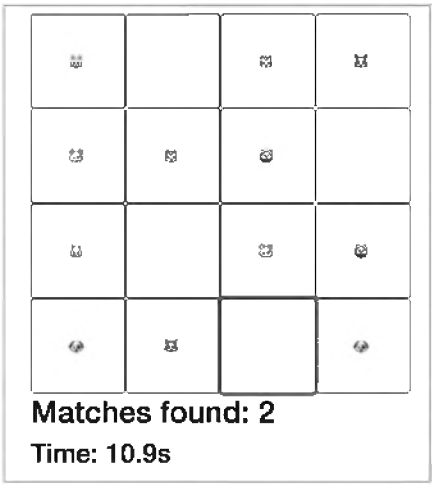
Перемещение по сцене

В последней лабораторной работе Unity была создана сцена с полом (плоскость) и игроком (сфера с цилиндром). При этом использовался объект NavMesh, NavMesh Agent и отслеживание лучей, чтобы игрок следовал за щелчками в сцене. В этой работе мы воспользуемся навигационной системой Unity, чтобы объекты GameObject сами перемещались по сцене.

Это препятствие NavMesh создает движущийся проем в NavMesh, который мешает игроку перемещаться вверх по наклонной плоскости. Мы добавим сценарий, который позволяет игроку перетаскивать его мышью, чтобы блокировать и освобождать наклонную плоскость.

I Приложение 1. Проекты ASP.NET Core Blazor

Visual Studio для пользователей Mac



II Приложение II. Ката программирования

Ката программирования для опытных и/или нетерпеливых