

Simple Restaurant Simulation

Симулятор простого ресторана

C# Training Exercises

Задания C# тренинга

Overview (Обзор)

- ▶ Goals
- ▶ Problem Domain
- ▶ Requirements
- ▶ Design Guidelines

- ▶ Цели
- ▶ Описание задачи
- ▶ Требования
- ▶ Рекомендации к дизайну

Goals (Цели)

- ▶ Concepts in Visual C# Step by Step chapters 6, 7, 8
 - ▶ Compound statements
 - ▶ Exceptions
 - ▶ Nullable types
 - ▶ Simple classes and fields
 - ▶ Static members
 - ▶ Copying objects
- ▶ Основные понятия глав 6, 7, 8 из книги Visual C# Step by Step
 - ▶ Операции сравнения
 - ▶ Исключения
 - ▶ Нулевые типы (типы с возможными нулевыми значениями)
 - ▶ Простые классы и поля
 - ▶ Статические члены
 - ▶ Копирование объектов

Problem Domain: A Simple Restaurant (Постановка задачи: Простой ресторан)

- ▶ You are the owner of a restaurant
 - ▶ The restaurant serves chicken (meat) and eggs
 - ▶ You have one employee, who obtains the food materials, inspects the eggs, and prepares the food
 - ▶ This computer program will simulate:
 - ▶ Submitting requests from customers
 - ▶ Servicing the requests by preparing the food
- ▶ Вы владелец ресторана
 - ▶ Ресторан подает яйца и курицу (мясо)
 - ▶ У вас есть один сотрудник, который получает продукты питания, проверяет яйца и готовит еду.
 - ▶ Эта компьютерная программа должна симулировать:
 - ▶ Прием заказов клиентов
 - ▶ Накрывание на стол приготовленных блюд

Requirements

(Требования)

Sample User Interface

(Пример пользовательского интерфейса)

The screenshot shows a web application window titled "Resta". The interface includes the following elements:

- A form with a label "Menu" and two radio button options: "Chicken" (selected) and "Egg".
- A "Quantity" input field containing the value "3".
- A "Submit new request" button.
- A "Copy the previous request" button.
- A text label "Egg Quality: 65".
- A "Prepare Food" button.
- A "Results" section with an empty rectangular box below it.

Simulated Employee Actions

(Симуляция поведения сотрудника)

1. Submit new request
2. Copy the previous request
3. Prepare food

1. Прием нового заказа
2. Копирование предыдущего заказа
3. Приготовление еды

Action 1: Submit New Request

Действие 1. Прием нового заказа

- ▶ The employee is requested to obtain food materials
 - ▶ The request contains:
 - ▶ Menu item (chicken or egg)
 - ▶ Quantity (numeric value)
 - ▶ The employee reports back to you what he obtained
 - ▶ Your employee is forgetful!
1/3 of the time he obtains a different menu item than you requested
 - ▶ He always obtains the correct requested quantity
 - ▶ If he obtains eggs, he also inspects their quality and reports them to you.
However, he forgets to do so $\frac{1}{2}$ of the time.
- ▶ Сотруднику предлагается получить продукты питания
 - ▶ Запрос содержит:
 - ▶ Элемент меню (курица или яйцо)
 - ▶ Количество (числовое значение)
 - ▶ Сотрудник сообщает обратно вам, что он подготовил
 - ▶ У вас забывчивый работник! В 1/3 случаев подготавливает не то блюдо, которое было заказано
 - ▶ Он всегда подготавливает правильное запрошенное количество.
 - ▶ Если он получает яйца, он также проверяет их качество и сообщает вам. К сожалению, он забудет сделать это в половине случаев.

Action 2: Copy the Previous Request

(Действие 2. Копирование предыдущего заказа)

- ▶ The employee will obtain the same food materials that he obtained the last time
- ▶ If he has never been given a request, he will get upset at you...
- ▶ Работник готовит такое же количество еды, что и в прошлый раз
- ▶ Если у него никогда не было заказов, он на вас обидется.

Action 3: Prepare Food

(Действие 3. Приготовление еды)

CHICKEN

- ▶ For each chicken ordered:
 - ▶ Cut up the meat
- ▶ Cook the entire chicken request at once
- ▶ The employee reports back the results

КУРИЦА

- ▶ Для каждой заказанной курицы:
 - ▶ Порезать мясо
- ▶ Готовить все заказанные курицы вместе
- ▶ Работник сообщает результат

EGGS

- ▶ For each egg ordered:
 - ▶ Crack the egg into a bowl
 - ▶ Discard the egg shell
- ▶ Cook the entire egg request at once
- ▶ The employee reports back the results

ЯЙЦА

- ▶ Для каждого заказанного яйца:
 - ▶ Разбить яйцо в миску
 - ▶ Очистить яйцо от скорлупы
- ▶ Готовить все заказанные яйца вместе
- ▶ Работник сообщает результат

Design Guidelines

(Рекомендации к дизайну)

Classes to be Created

Классы, которые должны быть созданы

1. ChickenOrder
2. EggOrder
3. Employee
4. User Interface Dialog (Пользовательский интерфейс)

ChickenOrder Class

Класс ChickenOrder

- ▶ Constructor argument is quantity requested
 - ▶ GetQuantity method
 - ▶ Returns same value as passed in constructor
 - ▶ CutUp method
 - ▶ No arguments or return value
 - ▶ Should be called the number of times requested in quantity
 - ▶ Cook method
 - ▶ No arguments or return value
 - ▶ Should be called once only
- ▶ Аргумент конструктора - количество
 - ▶ Метод GetQuantity
 - ▶ Возвращает то значение, которые передалось в конструкторе
 - ▶ Метод CutUp
 - ▶ Нет аргументов и возвращаемого значения
 - ▶ Должен вызываться столько раз, сколько требуется в зависимости от количества курицы
 - ▶ Метод Cook
 - ▶ Нет аргументов и возвращаемого значения
 - ▶ Должен быть вызван один раз

EggOrder Class

Класс EggOrder

- ▶ Constructor argument is quantity requested
 - ▶ GetQuantity method
 - ▶ Returns same value as passed in constructor
 - ▶ GetQuality method
 - ▶ Quality is a random numeric value between 1 and 100, calculated once per instance of the class
 - ▶ The method should return a nullable value type
 - ▶ To simulate the employee forgetting $\frac{1}{2}$ of the time, the method should return a *null* value on the 2nd, 4th, 6th, etc., instances of the class.
 - ▶ Crack, DiscardShell, Cook methods
 - ▶ No arguments
 - ▶ No return value
 - ▶ Crack method should throw an exception if the egg quality is less than 25
 - ▶ This simulates a rotten egg
- ▶ Аргумент конструктора - количество
 - ▶ Метод GetQuantity
 - ▶ Возвращает то значение, которые передалось в конструкторе
 - ▶ Метод GetQuality
 - ▶ Качество - случайное значение в диапазоне от 1 до 100, вычисляемое для каждого экземпляра класса
 - ▶ Метод должен вернуть числовое значение, допускающее значение null
 - ▶ Для того, чтобы симитировать забывчивость работника, метод должен возвращать значение null для каждого 2-го, 4-го, 6-го и т.д. экземпляра класса
 - ▶ Методы Crack, DiscardShell, Cook
 - ▶ Без аргументов
 - ▶ Без возвращаемых значений
 - ▶ Метод Crack должен возвращать исключение, если качество яйца меньше 25
 - ▶ Это означает, что яйцо испорченное

Random Number Generation

(Генерация случайного числа)

- ▶ Can be used to calculate the EggOrder quality value
- ▶ In C#, this can be easily accomplished with code like the following:
- ▶ Можно использовать генератор случайных чисел для расчета значения качества EggOrder
- ▶ В C # это легко сделать с помощью следующего кода:

```
// Instantiate random number generator,  
// using system-supplied value as seed.  
Random rand = new Random();  
// Generate and display a random integer between 0 and 100.  
Console.WriteLine("{0,8:N0}", rand.Next(101));
```

Employee Class - Method Overview

Класс Employee - обзор методов

- ▶ Constructor with no arguments
- ▶ NewRequest method
 - ▶ Quantity argument
 - ▶ Menu item argument
 - ▶ Returns object (either ChickenOrder or EggOrder class)
- ▶ CopyRequest method
 - ▶ No arguments
 - ▶ Returns object (either ChickenOrder or EggOrder class)

- ▶ Конструктор без аргументов
- ▶ Метод NewRequest
 - ▶ Аргумент Quantity
 - ▶ Аргумент Menu item
 - ▶ Возвращает объект (либо класса ChickenOrder, либо EggOrder)
- ▶ Метод CopyRequest
 - ▶ Без аргументов
 - ▶ Возвращает объект (либо класса ChickenOrder, либо EggOrder)

- ▶ Inspect method
 - ▶ Object argument (as returned from NewRequest or CopyRequest methods)
 - ▶ Returns results of inspection (string)
- ▶ PrepareFood method
 - ▶ Object argument (as returned from NewRequest or CopyRequest methods)
 - ▶ Returns results of food preparation (string)

- ▶ Метод Inspect
 - ▶ Аргумент - объект, возвращаемый методами NewRequest или CopyRequest
 - ▶ Возвращает результат проверки (строка)
- ▶ Метод PrepareFood
 - ▶ Аргумент - объект, возвращаемый методами NewRequest или CopyRequest
 - ▶ Возвращает строку-результат о приготовленной еде

Employee Class - Method Details (1)

Класс Employee - детали методов (1)

Method NewRequest

- ▶ Create new instance of ChickenOrder or EggOrder class
- ▶ To simulate the employee forgetting 1/3 of the time, the method should create an instance of the wrong class on the 3rd, 6th, 9th, etc., new requests.
- ▶ Save reference to the new instance so it can be copied
- ▶ Return the new instance of the class

CopyRequest method

- ▶ Create a new class instance with same type, quantity and quality (if EggOrder) as previously created instance
- ▶ Return the new instance of the class

Inspect method

- ▶ If argument is a ChickenOrder, return string that specifies no inspection is required
- ▶ If argument is an EggOrder, return the quality value

Метод NewRequest

- ▶ Создаёт новый экземпляр класса ChickenOrder или EggOrder
- ▶ Чтобы смоделировать забывчивость работника 1/3 случаев, в 3-м, 6-м, 9м и т. д. новых запросах, метод должен создавать экземпляр неправильного класса
- ▶ Сохраните ссылку на новый экземпляр, чтобы его можно было скопировать
- ▶ Возвращает новый экземпляр класса

Метод CopyRequest

- ▶ Создаёт новый экземпляр класса с тем же типом, количеством и качеством (если EggOrder), как и ранее созданный экземпляр
- ▶ Возвращает новый экземпляр класса

Метод Inspect

- ▶ Если аргумент - ChickenOrder, возвращает строку, проверка не требуется.
- ▶ Если аргумент - EggOrder, возвращает значения качества

Employee Class - Method Details (2)

Класс Employee - детали методов (2)

▶ PrepareFood method

- ▶ If argument is a ChickenOrder
 - ▶ Call CutUp method the number of times specified by ChickenOrder.GetQuantity
 - ▶ Call Cook method once
 - ▶ Return string indicating preparation has been completed
- ▶ If argument is an EggOrder
 - ▶ Call Crack method the number of times specified by EggOrder.GetQuantity
 - ▶ Call DiscardShell method after each call to Crack, even if Crack throws an exception
 - ▶ Call Cook method once
 - ▶ Return string indicating preparation has been completed, including the status of an egg. Whether it is rotten or not.

▶ Метод PrepareFood

- ▶ Если аргумент ChickenOrder
 - ▶ Вызывает метод CutUp столько раз, сколько указано в ChickenOrder.GetQuantity.
 - ▶ Вызывает метод Cook один раз
 - ▶ Возвращает строку, с информацией о том, что курица приготовлена.
- ▶ Если аргумент EggOrder
 - ▶ Вызывает метод Crack столько раз, сколько указано в EggOrder.GetQuantity
 - ▶ Вызывает метод DiscardShell после каждого вызова Crack, даже если Crack выдает исключение.
 - ▶ Вызывает метод Cook один раз
 - ▶ Возвращает строку, с информацией о том, что яйцо приготовлено. Также выводит статус яйца: тухлый или не.

Employee Class - Exceptions Thrown

Класс Employee - Исключения

- ▶ CopyRequest method
 - ▶ Throws exception if NewRequest has never been called before (because the employee has no request to copy)
 - ▶ PrepareFood method
 - ▶ Throws exception if called twice in a row (once the food is prepared, the employee cannot be prepare it again)
- ▶ Метод CopyRequest
 - ▶ Выдает исключение, если NewRequest никогда ранее не вызывался (потому что у employee нет запроса на копирование)
 - ▶ Метод PrepareFood
 - ▶ Выдает исключение, если вызывается дважды подряд (после того, как еда будет приготовлена, сотрудник не сможет приготовить ее снова)

User Interface Dialog Class (1)

Класс диалогового окна пользовательского интерфейса (1)

- ▶ Submit New Request button
 - ▶ Calls Employee.NewRequest
 - ▶ Calls Employee.Inspect
 - ▶ Displays results of inspection
- ▶ Copy Previous Request button
 - ▶ Calls Employee.CopyRequest
 - ▶ Calls Employee.Inspect
 - ▶ Displays results of inspection
- ▶ Кнопка Submit New Request
 - ▶ Вызывает Employee.NewRequest
 - ▶ Вызывает Employee.Inspect
 - ▶ Отображает результаты проверки
- ▶ Кнопка Copy Previous Request
 - ▶ Вызывает Employee.CopyRequest
 - ▶ Вызывает Employee.Inspect
 - ▶ Отображает результаты проверки

User Interface Dialog Class (2)

Класс диалогового окна пользовательского интерфейса (2)

- ▶ Prepare Food button
 - ▶ Calls Employee.PrepareFood method
 - ▶ Displays returned results
 - ▶ All exceptions should be caught and displayed in the results area
 - ▶ Be sure to choose an interesting name for your restaurant!
- ▶ Кнопка Prepare Food
 - ▶ Вызывает метод Employee.PrepareFood
 - ▶ Отображает возвращенные результаты
 - ▶ Все исключения должны быть отловлены и отображены в области результатов
 - ▶ Обязательно выберите интересное название для своего ресторана! 

Design Restrictions

Ограничения дизайна

- ▶ Avoid concepts that have not yet been introduced in the training. Specifically, do not use:
 - ▶ Member properties
 - ▶ Object Inheritance
 - ▶ Collections
- ▶ Why these restrictions?
 - ▶ To give you practice with the concepts that have just been taught, rather than alternate solutions you might already know.
- ▶ Избегайте концепций, которые еще не были представлены на тренинге. В частности, не используйте:
 - ▶ Свойства члена
 - ▶ Наследование объекта
 - ▶ Коллекции
- ▶ Зачем эти ограничения?
 - ▶ Чтобы дать Вам возможность попрактиковаться в использовании только что изученных концепций, а не альтернативных решений, которые вы, возможно, уже знаете.