

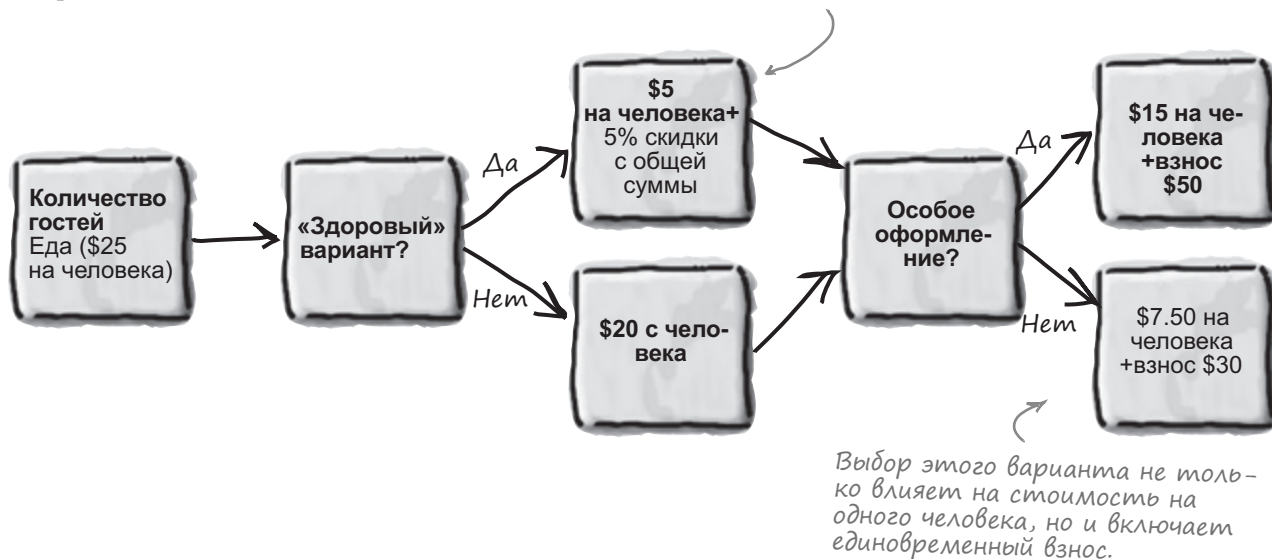
Как происходит оценка

Вот фрагмент записей Кэтлин, касающийся расчетов стоимости мероприятия:

Оценка стоимости обеда

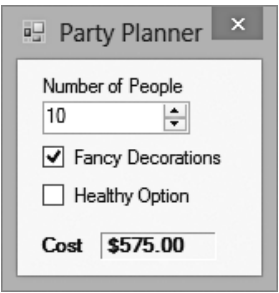
- За каждого гостя — \$25.
- Большинство обедов сервируется с алкогольными напитками — это дополнительные \$20 на человека. Можно выбрать «здоровый» вариант — это стоит всего \$5 на человека, вместо алкоголя подаются соки и газировка. «Здоровый» вариант намного проще в организации, поэтому скидка составит 5% на всю стоимость мероприятия.
- Обычное оформление стоит \$7.50 на человека плюс первоначальный взнос \$30. Стоимость особого оформления увеличивается до \$15 на человека, а первоначальный взнос составит \$50.

Для наглядности представим эти правила в виде диаграммы:



Вы создадите для Кэтлин программу

Перевернув страницу, вы обнаружите упражнение, в процессе выполнения которого вам предстоит создать для Кэтлин планировщик мероприятий. Вот предварительный план работ.



Вы постройте форму, в которой Кэтлин будет указывать параметры вечеринки. Достаточно указать количество человек и поставить или снять флажки, касающиеся особого оформления и здорового питания, и в нижней части формы появится стоимость будущего праздника.

DinnerParty
NumberOfPeople CostOfBeveragesPerPerson CostOfDecorations
SetHealthyOption() CalculateCostOfDecorations() CalculateCost()

Логика программы будет встроена в класс **DinnerParty**. Форма создаст объект **DinnerParty**, сохранит ссылки на него в поле и использует поля и методы для вычислений стоимости.

Вот как должна выглядеть верхняя часть формы.

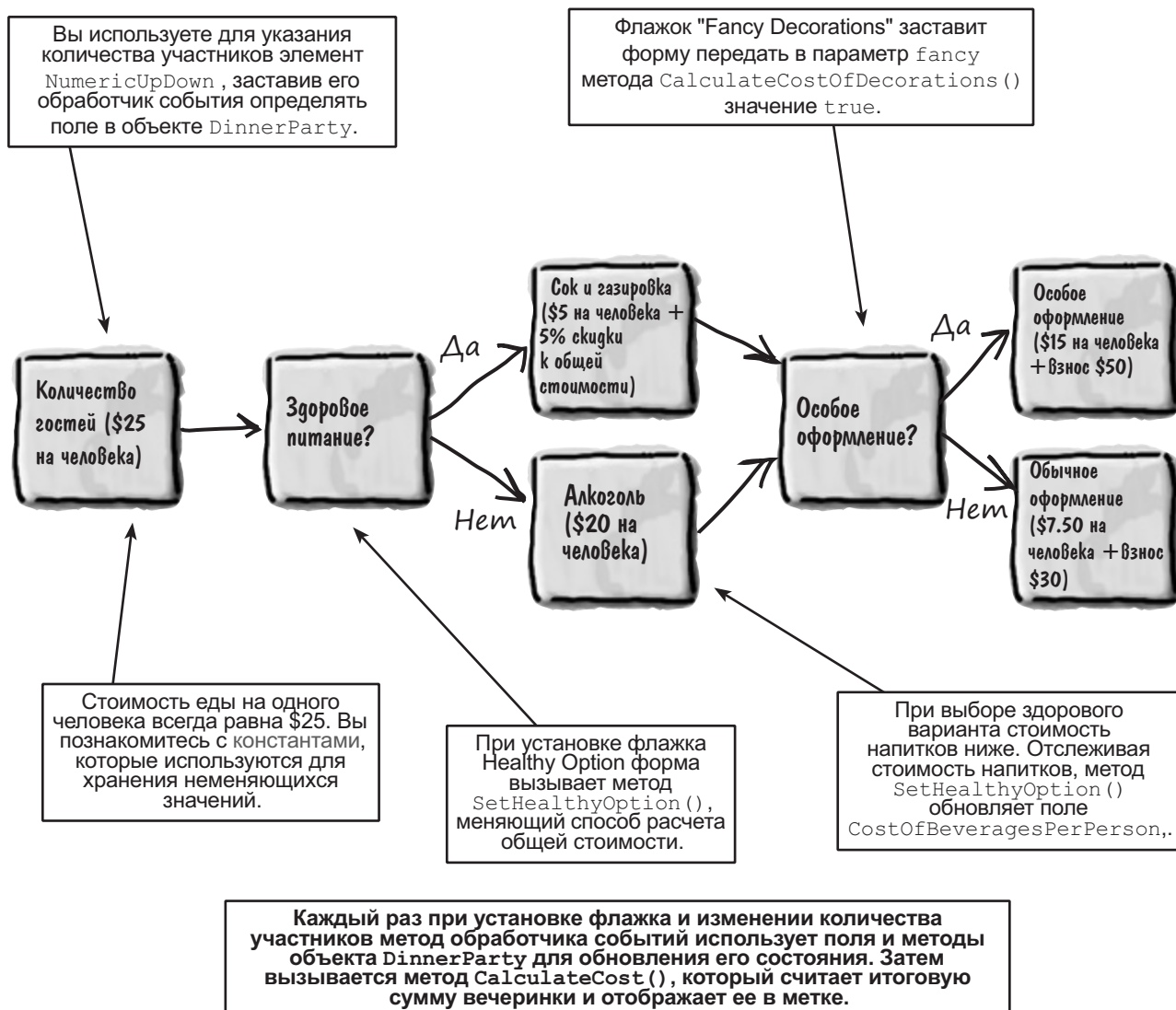
Для вычисления стоимости применяется поле

dinnerParty. Первым делом форма инициализируется значениями по умолчанию, а затем вычисляет стоимость `DisplayDinnerPartyCost()`, который вызывается при каждом изменении параметров пользователем.

```
public partial class Form1 : Form
{
    DinnerParty dinnerParty;

    public Form1()
    {
        InitializeComponent();
        dinnerParty = new DinnerParty() { NumberOfPeople = 5 };
        dinnerParty.SetHealthyOption(false);
        dinnerParty.CalculateCostOfDecorations(true);
        DisplayDinnerPartyCost();
    }
    ...
}
```

Вот как будет работать класс `DinnerParty`. Текущее состояние объекта `DinnerParty` — хранящееся в полях значения — определяет способ **вычисления стоимости**. Выбор здорового питания и особого оформления, а также изменение количества участников влияет на состояние объекта, что заставляет метод `CalculateCost()` пересчитать результат.



Разобрались? Тогда начнем работу! →



Упражнение

Создадим программу, оценивающую стоимость обедов.

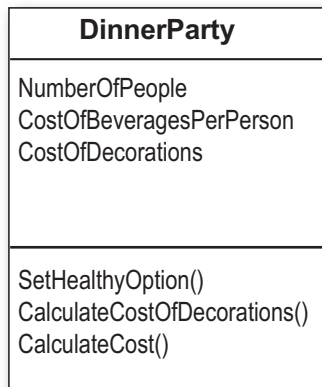


Диаграмма класса DinnerParty, который вам нужно создать.

Метод SetHealthyOption() использует логический параметр (healthyOption) для обновления значений поля CostOfBeveragesPerPerson.

Флажки называются fancyBox и healthyBox. Для элемента управления NumericUpDown можно сохранить имя по умолчанию.

1 Создайте проект Windows Application, добавьте класс DinnerParty.cs и оформите его в соответствии с приведенной слева диаграммой. Методы затрагивают расчеты стоимости «здорового» варианта, стоимости оформления, а также общей стоимости мероприятия. Два последних поля должны относиться к типу decimal, а первое — к типу int. Убедитесь, что в конце каждой константы типа decimal **стоит буква М** (например, 10.0М).

2 Так как стоимость еды на одного человека не будет меняться, ее можно объявить как **константу**. Вот как это делается:

```
public const int CostOfFoodPerPerson = 25;
```

3 Вернитесь на предыдущую страницу, чтобы убедиться, что вы работаете с нужными методами. Один из методов возвращает значение типа decimal, два других — нет. Метод CalculateCostOfDecorations() вычисляет стоимость оформления в зависимости от количества приглашенных. Метод CalculateCost() вычисляет общую стоимость, складывая цену оформления, еды и напитков в зависимости от количества приглашенных. При выборе «здорового варианта» делается скидка с общей стоимости.

4 Добавьте к форме код:

```
DinnerParty dinnerParty;

public Form1() {
    InitializeComponent();
    dinnerParty = new DinnerParty() { NumberOfPeople = 5 };
    dinnerParty.SetHealthyOption(false);
    dinnerParty.CalculateCostOfDecorations(true);
    DisplayDinnerPartyCost();
}
```

Вы объявляете поле dinnerParty и добавляете четыре строчки.

5 Форма должна выглядеть так. Используйте элемент NumericUpDown, чтобы задать максимальное количество людей равным 20, а минимальное — 1. По умолчанию количество приглашенных равно 5. Избавьтесь от кнопок управления размерами окна.

По умолчанию число приглашенных равно пяти.

Свойство Checked флажка Fancy Decorations должно иметь значение True.

Имя этой метки costLabel. Свойство Text оставлено пустым, свойство BorderStyle имеет значение Fixed3D, а свойство AutoSize — значение false.

6

Сделаем так, чтобы итоговая сумма автоматически менялась при изменении флажков и показаний счетчика NumericUpDown. Для этого вам потребуется метод, отображающий сумму.

Этот метод вызывается всеми методами, связанными с формой. Именно так обновляется значение метки Cost.

Добавим его к классу Form1. Он будет вызываться при щелчке на элементе NumericUpDown:

```
private void DisplayDinnerPartyCost()
{
    decimal Cost = dinnerParty.CalculateCost(checkBox2.Checked);
    costLabel.Text = Cost.ToString("c");
}
```

Метод рассчитывает стоимость обеда и передает результат метке Cost.

Присвойте метке, отображающей цену, имя costLabel.

Аргумент "с" метода ToString() отображает валюту с использованием принятого по соглашению символа.

Значение true появляется при установке флажка Healthy Option.

7

Соединим поле NumericUpDown с переменной NumberOfPeople, принадлежащей классу DinnerParty, чтобы в форме начала отображаться сумма расходов. Дважды щелкните на элементе NumericUpDown, в код будет добавлен **обработчик событий (event handler)**. Так называется метод, запускаемый при каждом изменении элемента управления. Он сбросит количество гостей. Впишите следующий код:

При двойном щелчке на кнопке IDE добавляет обработчик события Click.

```
private void numericUpDown1_ValueChanged(
    object sender, EventArgs e)
{
    dinnerParty.NumberOfPeople = (int) numericUpDown1.Value;
    DisplayDinnerPartyCost();
}
```

Ой, ошибка в коде. Можете указать, где именно? Не волнуйтесь, если вы этого пока не видите.

Значение numericUpDown1.Value принадлежит типу Decimal, поэтому требуется операция приведения типов.

Ой... код содержит ошибку. Вы ее видите? Если нет, не волнуйтесь. Скоро мы все объясним!

Из формы методу передается логическое значение fancyBox.Checked.

Первый метод рассчитывает стоимость мероприятия, а второй отображает конечную сумму в форме.

8

Дважды щелкните на флажке **Fancy Decorations** и убедитесь, что сначала он вызывает метод CalculateCostOfDecorations(), а потом метод DisplayDinnerPartyCost(). Затем дважды щелкните на флажке **Healthy Option** и убедитесь, что он сначала вызывает метод SetHealthyOption() класса DinnerParty, а затем метод DisplayDinnerPartyCost().