



Microsoft Partner
Silver Learning

C# Стартовый

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Ветвление



ITVDN
IT VIDEO DEVELOPERS NETWORK

Introduction



Александр Шевчук



MCID: 9230440

Тема урока

Ветвление



ГЛАВА 15

Условные операторы

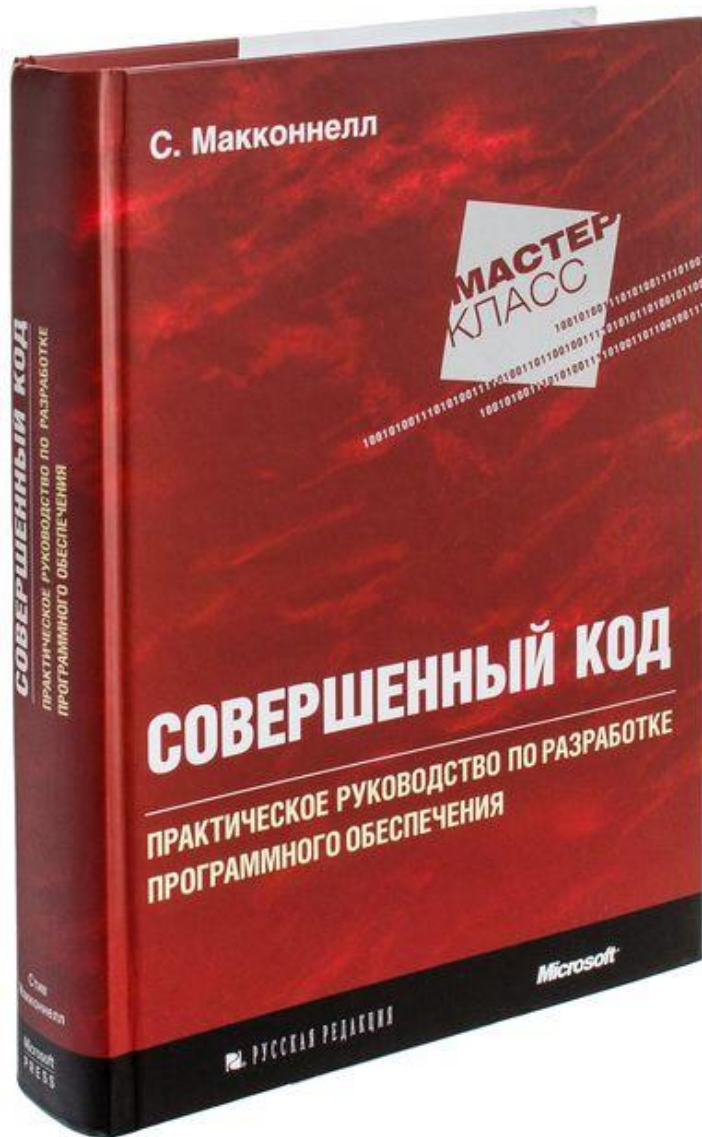
15.1. Операторы *if*

Стр. 346

Задание

Чтобы привлечь больше клиентов, руководство компании решило каждому покупателю (безоговорочно) предоставлять скидку в 10% от общей стоимости покупки. Требуется написать программу, которая должна:

1. Запросить у кассира ввести количество плитки в метрах квадратных.
2. Посчитать общую стоимость плитки.
3. Посчитать сумму скидки в 10%.
4. Посчитать сумму к оплате с учетом скидки.
5. Вывести на экран: общую стоимость, сумму скидки и сумму к оплате.



9.2. Псевдокод для профи

Стр. 211

Псевдокодом называют неформальную нотацию на естественном языке, описывающую работу алгоритма, метода, класса или программы. «Процесс программирования с псевдокодом» относится к конкретной методике применения псевдокода для эффективного создания кода методов.

Пример хорошего псевдокода

Стр. 212

Отслеживать текущее число используемых ресурсов

Если другой ресурс доступен

- Выделить структуру для диалогового окна

- Если структура для диалогового окна может быть выделена

 - Учесть, что используется еще один ресурс

 - Инициализировать ресурс

 - Хранить номер ресурса в вызывающей программе

- Конец «если»

Конец «если»

Вернуть true, если новый ресурс был создан; иначе вернуть false



24.3. Отдельные виды рефакторинга

Рефакторинг на уровне данных

Стр. 559

Введение промежуточной переменной

Присвойте результат вычисления выражения промежуточной переменной, имя которой резюмирует суть выражения.



12.5. Логические переменные

Используйте логические переменные для документирования программы

Вместо простой проверки логического выражения вы можете присвоить его значение переменной, которая сделает, его* смысл очевидным.

**перепаразировано*

Стр. 292

Используйте логические переменные для упрощения сложных условий

Чтобы правильно закодировать сложное условие, часто приходится делать несколько попыток. Когда через некоторое время нужно модифицировать *некоторое условие, бывает сложно разобраться, что же оно проверяет. Логические переменные могут упростить проверку.

**перепаразировано*

Стр. 293



19.1. Логические выражения

Использование true и false в логических проверках

Стр. 426

Используйте неявное сравнение логических величин с true или false

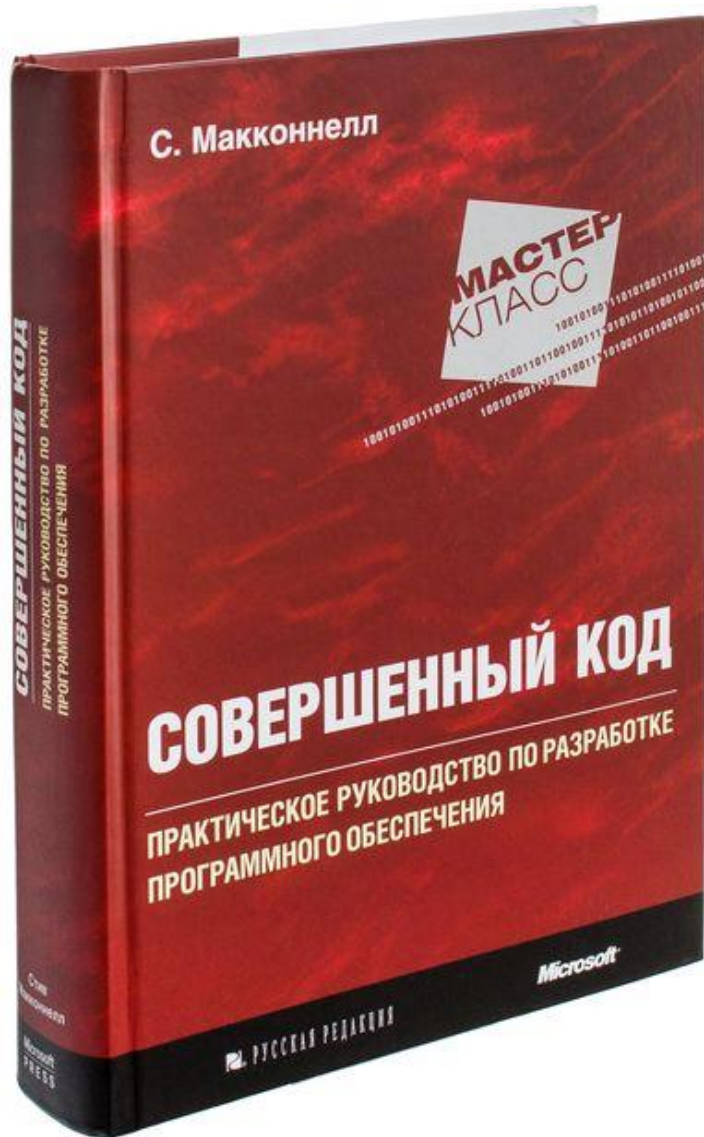
Вы сможете сделать проверку условия более понятной, если будете рассматривать проверяемые выражения как логические. Например, пишите:

```
if (discountAvailable) ← Неявное сравнение
```

вместо:

```
if (discountAvailable == true) ← Явное сравнение
```

Использование неявных сравнений уменьшает число элементов, которые придется помнить при чтении кода, и приближает получаемое выражение к разговорному английскому.



12.1. Числа в общем

Стр. 283

Избегайте «магических чисел»

Магические числа — это обычные числа, такие как 100 или 47524, которые появляются в программе без объяснений. Если вы программируете на языке, поддерживающем именованные константы, используйте их вместо магических чисел.

24.3. Отдельные виды рефакторинга

Стр. 559

Рефакторинг на уровне данных

Замена магического числа на именованную константу

Если вы используете численный или строковый литерал, скажем, 3.14, замените его именованной константой, такой как PI.

19.1. Логические выражения

...

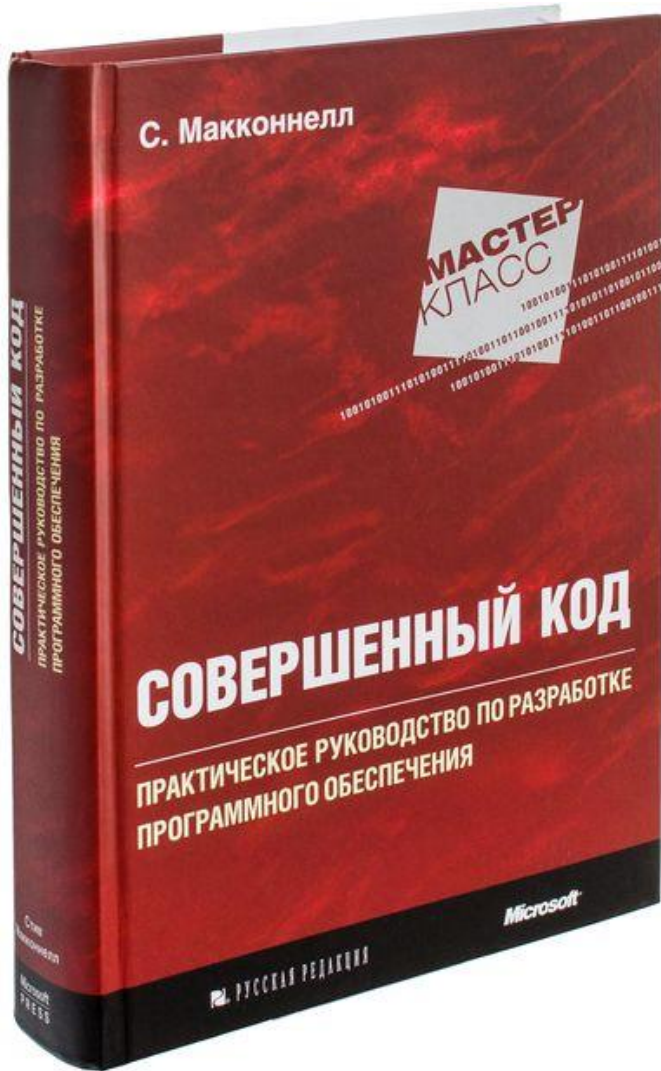
Общие проблемы с логическими выражениями

...

Стр. 435

В Си-подобных языках, помещайте константы с левой стороны сравнений

В Си-подобных языках возникают некоторые специфические проблемы с логическими выражениями. **Если время от времени вы набираете = вместо ==**, подумайте об использовании соглашения по размещению констант и литералов с левой стороны выражений ...



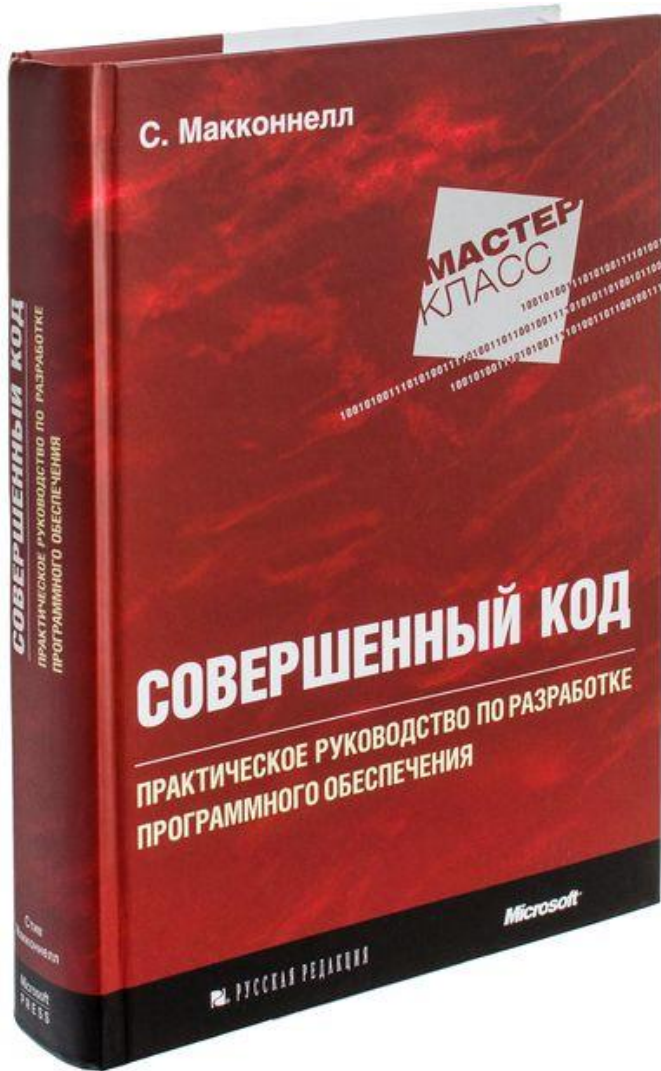


12.1. Числа в общем

Стр. 284

Выполняйте преобразования типов понятно

Убедитесь, что кто-нибудь, читая вашу программу, поймет преобразования между разными типами данных, которые в ней встречаются.



11.1. Общие принципы выбора имен переменных

...

Оптимальная длина имени переменной

Стр. 255

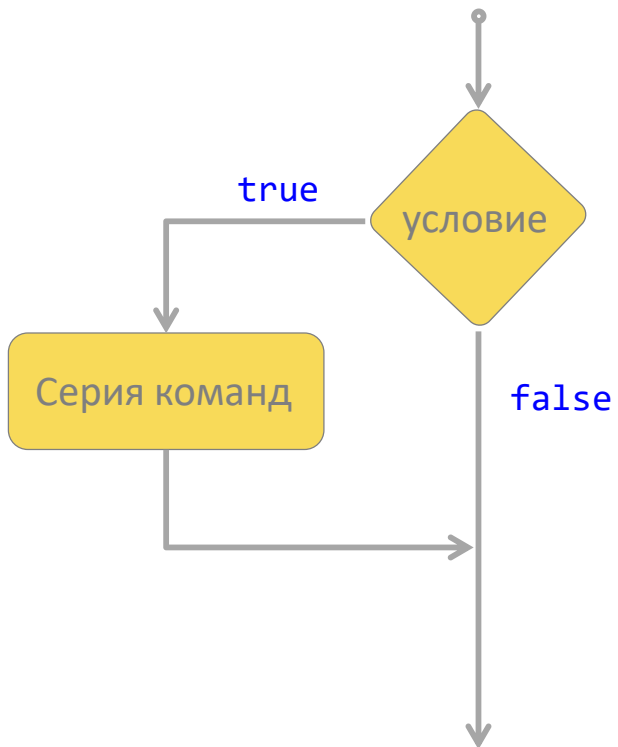
Оптимальная длина имени, наверное, лежит где-то между длинами имен `x` и `maximumNumberOfPointsInModernOlympics`. Слишком короткие страдают от недостатка смысла. Проблема с именами вроде `x1` и `x2` в том, что, даже узнав, что такое `x`, вы ничего не сможете сказать об отношении между `x1` и `x2`. Слишком длинные имена надоедает печатать, к тому же они могут сделать неясной визуальную структуру программы.

Горла, Бенандер и Бенандер обнаружили, что отладка программы требовала меньше всего усилий, если имена переменных состояли в среднем из 10–16 символов ...

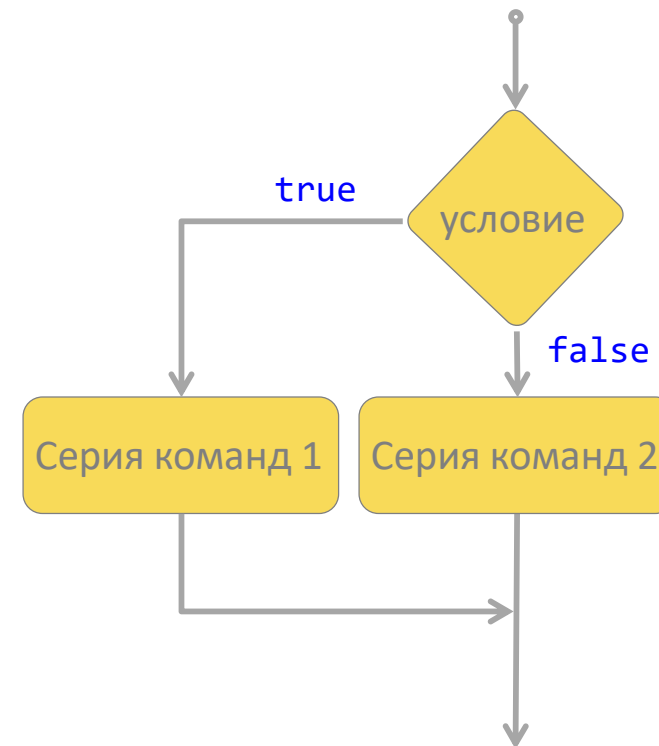
...

Условный оператор if

Условная конструкция **if**



Условная конструкция **if-else**



ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Спасибо за внимание! До новых встреч!



Александр Шевчук



MCID: 9230440

Информационный видеоресурс для разработчиков программного обеспечения

