

Лекция №4

Рефакторинг

Рефакторинг

Рефакторинг (англ. refactoring) — процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и имеющий целью облегчить понимание её работы.

Цель

Сделать код программы легче для понимания, сопровождения

Причины

- дублирование кода;
- длинный метод;
- большой класс;
- длинный список параметров;
- «завистливые» функции — это метод, который чрезмерно обращается к данным другого объекта;
- избыточные временные переменные;
- классы данных;
- несгруппированные данные.

Не рефакторинг

Оптимизация алгоритма

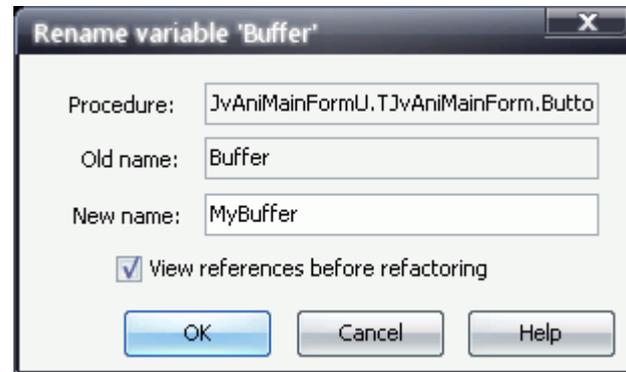
Форматирование кода

Исправление ошибок

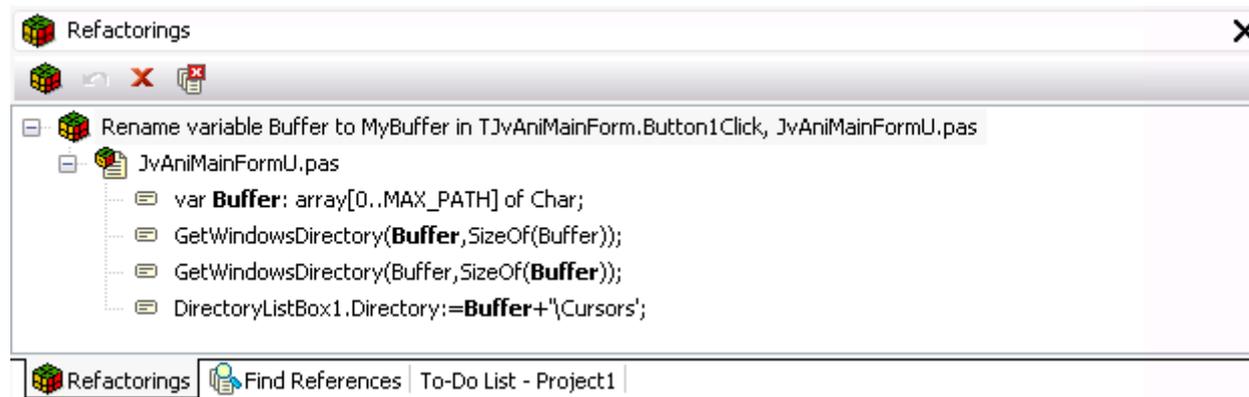
Методы

- Изменение сигнатуры метода (Change Method Signature)
- Инкапсуляция поля (Encapsulate Field)
- Выделение класса (Extract Class)
- Выделение интерфейса (Extract Interface)
- Выделение локальной переменной (Extract Local Variable)
- Выделение метода (Extract Method)
- Генерализация типа (Generalize Type)
- Встраивание (Inline)
- Введение фабрики (Introduce Factory)
- Введение параметра (Introduce Parameter)
- Подъём поля/метода (Pull Up)
- Спуск поля/метода (Push Down)
- Замена условного оператора полиморфизмом (Replace Conditional with Polymorphism)

Переименование



Переименование переменной, метода как локально, так и глобально.



Изменение сигнатуры метода (Change Method Signature)

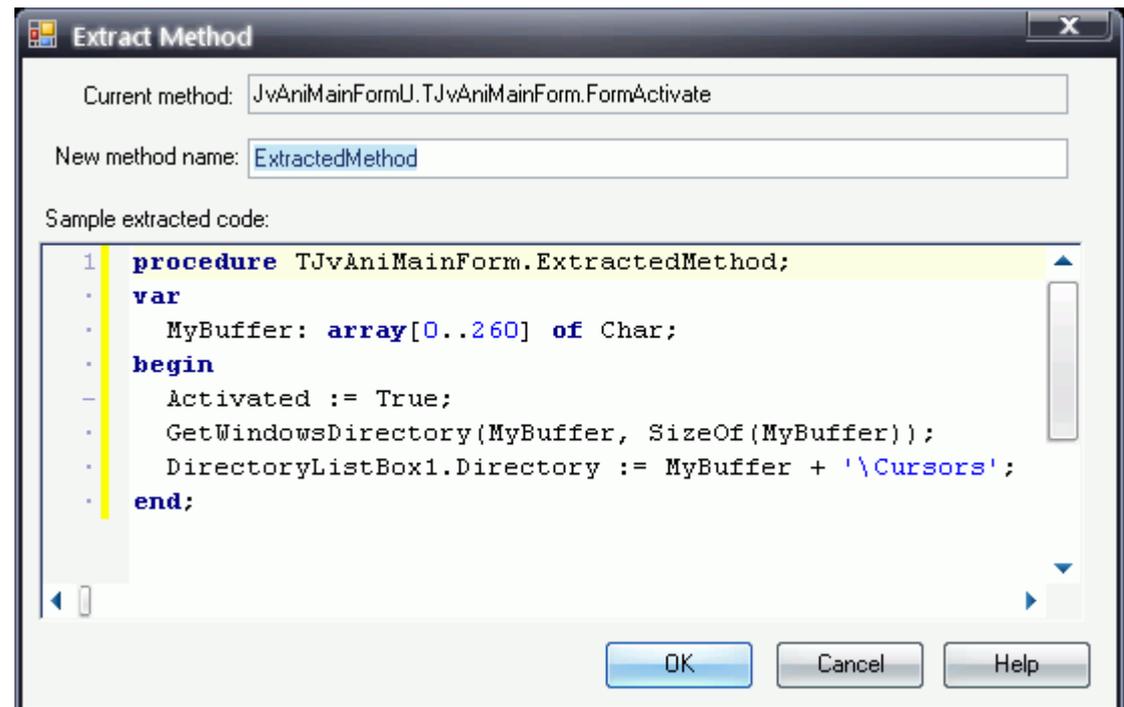
Суть изменения сигнатуры метода заключается в добавлении, изменении или удалении параметра метода. Изменив сигнатуру метода, необходимо скорректировать обращения к нему в коде всех клиентов. Это изменение может затронуть внешний интерфейс программы, кроме того, не всегда разработчику, изменяющему интерфейс, доступны все клиенты этого интерфейса, поэтому может потребоваться та или иная форма регистрации изменений интерфейса для последующей передачи их вместе с новой версией программы.

Инкапсуляция поля (Encapsulate field)

В случае, если у класса имеется открытое поле, необходимо сделать его закрытым и обеспечить методы доступа.

Извлечение метода (Extract Method)

Из большой процедуры извлекается фрагмент и оформляется как небольшая процедура/функция.



Извлечение метода (Extract Method)

```
void printOwing(double amount) {  
    printBanner();  
  
    //print details  
    System.out.println ("name:" + _name);  
    System.out.println ("amount" + amount);  
}
```



```
void printOwing(double amount) {  
    printBanner();  
    printDetails(amount);  
}  
  
void printDetails (double amount) {  
    System.out.println ("name:" + _name);  
    System.out.println ("amount" + amount);  
}
```

Перемещение метода (Move Method)

Перемещение метода применяется по отношению к методу, который чаще обращается к другому классу, чем к тому, в котором сам располагается.

Замена условного оператора полиморфизмом (Replace Conditional with Polymorphism)

Условный оператор с несколькими ветвями заменяется вызовом полиморфного метода некоторого базового класса, имеющего подклассы для каждой ветви исходного оператора. Выбор ветви осуществляется неявно, в зависимости от того, экземпляру какого из подклассов оказался адресован вызов.

Inline Method

```
int getRating() {  
    return (moreThanFiveLateDeliveries()) ? 2 : 1;  
}  
boolean moreThanFiveLateDeliveries() {  
    return _numberOfLateDeliveries > 5;  
}
```



```
int getRating() {  
    return (_numberOfLateDeliveries > 5) ? 2 : 1;  
}
```

Inline Temp

```
double basePrice = anOrder.basePrice();  
return (basePrice > 1000)
```



```
return (anOrder.basePrice() > 1000)
```

Replace Temp with Query

```
double basePrice = _quantity * _itemPrice;  
if (basePrice > 1000)  
    return basePrice * 0.95;  
else  
    return basePrice * 0.98;
```



```
if (basePrice() > 1000)  
    return basePrice() * 0.95;  
else  
    return basePrice() * 0.98;  
  
double basePrice() {  
    return _quantity * _itemPrice;  
}
```

Split Temporary Variable

```
double temp = 2 * (_height + _width);  
System.out.println (temp);  
temp = _height * _width;  
System.out.println (temp);
```



```
final double perimeter = 2 * (_height + _width);  
System.out.println (perimeter);  
final double area = _height * _width;  
System.out.println (area);
```

...

- Выделение класса (Extract Class)
- Выделение интерфейса (Extract Interface)
- Выделение локальной переменной (Extract Local Variable)
- Выделение метода (Extract Method)
- Генерализация типа (Generalize Type)
- Встраивание (Inline)
- Введение фабрики (Introduce Factory)
- Введение параметра (Introduce Parameter)
- Подъём поля/метода (Pull Up)
- Спуск поля/метода (Push Down)

Полный список

Составление методов

- Выделение метода (Extract Method)
- Встраивание метода (Inline Method)
- Встраивание временной переменной (Inline Temp)
- Замена временной переменной вызовом метода (Replace Temp with Query)
- Введение поясняющей переменной (Introduce Explaining Variable)
- Расщепление временной переменной (Split Temporary Variable)
- Удаление присваиваний параметрам (Remove Assignments to Parameters)
- Замена метода объектом методов (Replace Method with Method Object)
- Замещение алгоритма (Substitute Algorithm)

Перемещение функций между объектами

- Перемещение метода (Move Method)
- Перемещение поля (Move Field)
- Выделение класса (Extract Class)
- Встраивание класса (Inline Class)
- Соккрытие делегирования (Hide Delegate)
- Удаление посредника (Remove Middle Man)
- Введение внешнего метода (Introduce Foreign Method)
- Введение локального расширения (Introduce Local Extension)

Полный список

Организация данных

- Самоинкапсуляция поля (Self Encapsulate Field)
- Замена значения данных объектом (Replace Data Value with Object)
- Замена значения ссылкой (Change Value to Reference)
- Замена ссылки значением (Change Reference to Value)
- Замена массива объектом (Replace Array with Object)
- Дублирование видимых данных (Duplicate Observed Data)
- Замена однонаправленной связи двунаправленной (Change Unidirectional Association to Bidirectional)
- Замена двунаправленной связи однонаправленной (Change Bidirectional Association to Unidirectional)
- Замена магического числа символической константой (Replace Magic Number with Symbolic Constant)
- Инкапсуляция поля (**Encapsulate Field**)
- Инкапсуляция коллекции (**Encapsulate Collection**)
- Замена записи классом данных (**Replace Record with Data Class**)
- Замена кода типа классом (**Replace Type Code with Class**)
- Замена кода типа подклассами (**Replace Type Code with Subclasses**)
- Замена кода типа состоянием/стратегией (**Replace Type Code with State/Strategy**)
- Замена подкласса полями (**Replace Subclass with Fields**)

Полный список

Упрощение условных выражений

- Декомпозиция условного оператора (Decompose Conditional)
- Консолидация условного выражения (Consolidate Conditional Expression)
- Консолидация дублирующихся условных фрагментов (Consolidate Duplicate Conditional Fragments)
- Удаление управляющего флага (Remove Control Flag)
- Замена вложенных условных операторов граничным оператором (Replace Nested Conditional with Guard Clauses)
- Замена условного оператора полиморфизмом (Replace Conditional with Polymorphism)
- Введение объекта Null (Introduce Null Object)
- Введение утверждения (Introduce Assertion)

Упрощение вызовов методов

- Переименование метода (Rename Method)
- Добавление параметра (Add Parameter)
- Удаление параметра (Remove Parameter)
- Разделение запроса и модификатора (Separate Query from Modifier)
- Параметризация метода (Parameterize Method)
- Замена параметра явными методами (Replace Parameter with Explicit Methods)
- Сохранение всего объекта (Preserve Whole Object)
- Замена параметра вызовом метода (Replace Parameter with Method)
- Введение граничного объекта (Introduce Parameter Object)
- Удаление метода установки значения (Remove Setting Method)

ПОЛНЫЙ СПИСОК

- Соккрытие метода (Hide Method)
- Замена конструктора фабричным методом (Replace Constructor with Factory Method)
- Инкапсуляция нисходящего преобразования типа (Encapsulate Downcast)
- Замена кода ошибки исключительной ситуацией (Replace Error Code with Exception)
- Замена исключительной ситуации проверкой (Replace Exception with Test)

Решение задач обобщения

- Подъем поля (Pull Up Field)
- Подъем метода (Pull Up Method)
- Подъем тела конструктора (Pull Up Constructor Body)
- Спуск метода (Push Down Method)
- Спуск поля (Push Down Field)
- Выделение подкласса (Extract Subclass)
- Выделение родительского класса (Extract Superclass)
- Выделение интерфейса (Extract Interface)
- Свертывание иерархии (Collapse Hierarchy)
- Формирование шаблона метода (Form Template Method)
- Замена наследования делегированием (Replace Inheritance with Delegation)
- Замена делегирования наследованием (Replace Delegation with Inheritance)

Крупные рефакторинги

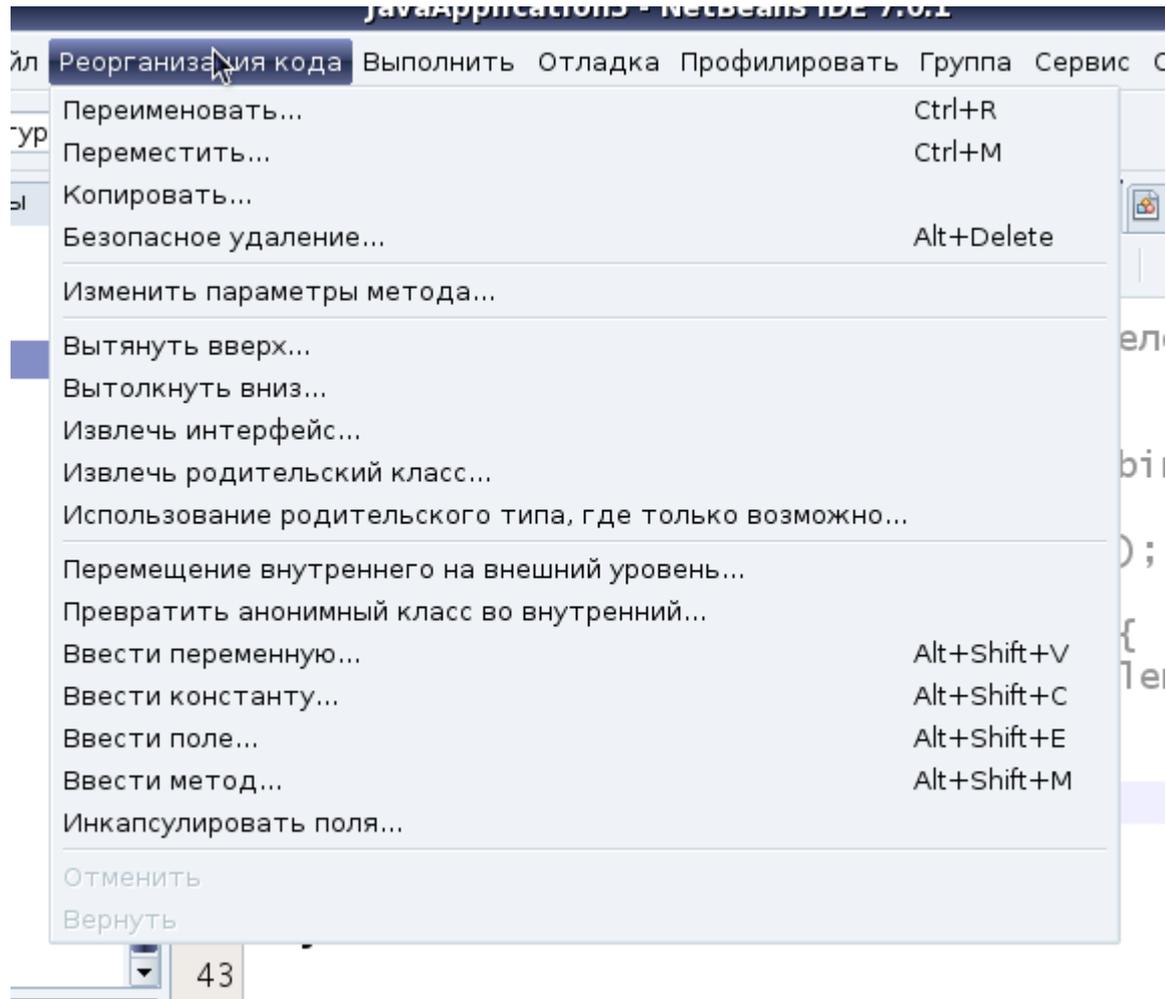
- Разделение наследования (Tease Apart Inheritance)
- Преобразование процедурного проекта в объекты (Convert Procedural Design to Objects)
- Отделение предметной области от представления (Separate Domain from Presentation)
- Выделение иерархии (Extract Hierarchy)

Проблемы рефакторинга

- Базы данных
- Интерфейсы
- Тесная интеграция

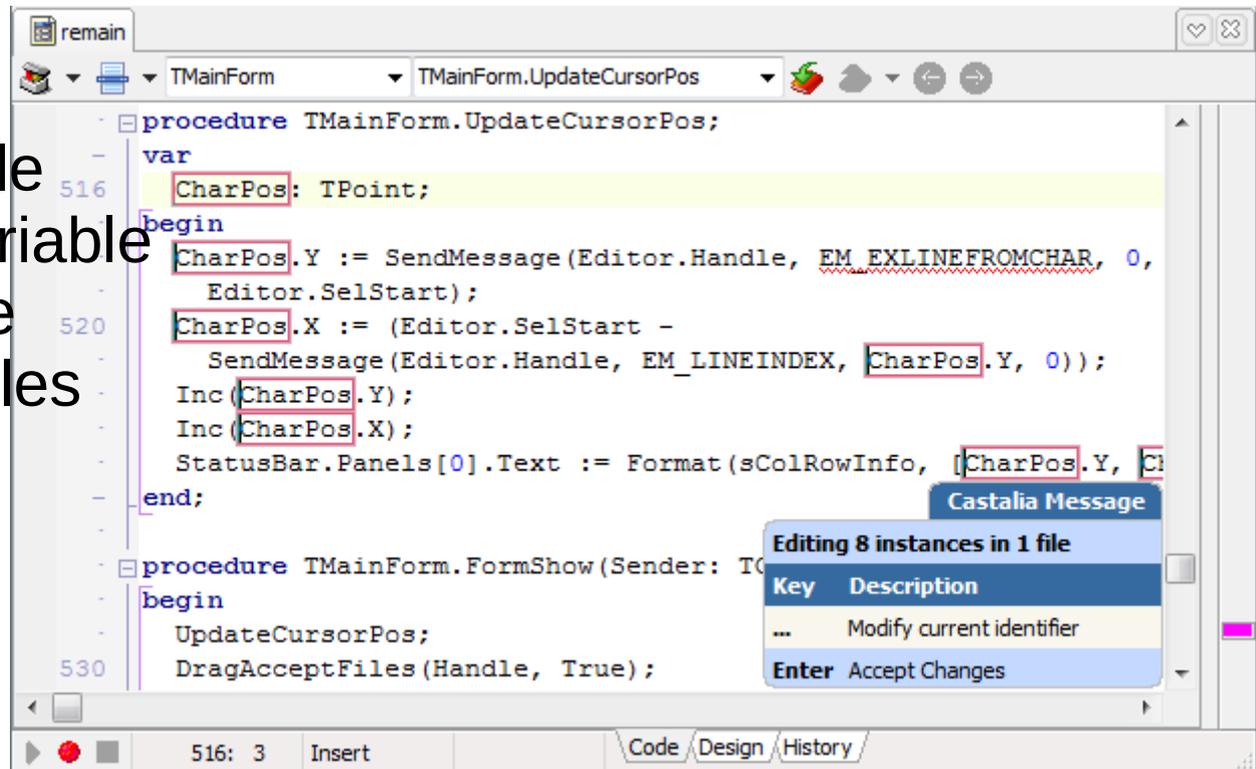
Средства автоматизации

IDE (Delphi,
NetBeans)



Castalia for Delphi

- Rename Local Variable
- Rename Parameter
- Inline Temporary Variable
- Introduce Explaining Variable
- Split Temporary Variable
- Remove Unused Variables
- Eliminate 'with'
- Rename Method
- Extract Method
- Add Parameter
- Rename Class
- Move Class



```
procedure TMainForm.UpdateCursorPos;
var
  CharPos: TPoint;
begin
  CharPos.Y := SendMessage(Editor.Handle, EM_EXLINEFROMCHAR, 0,
    Editor.SelStart);
  CharPos.X := (Editor.SelStart -
    SendMessage(Editor.Handle, EM_LINEINDEX, CharPos.Y, 0));
  Inc(CharPos.Y);
  Inc(CharPos.X);
  StatusBar.Panels[0].Text := Format(sColRowInfo, [CharPos.Y, C]);
end;

procedure TMainForm.FormShow(Sender: TComponent);
begin
  UpdateCursorPos;
  DragAcceptFiles(Handle, True);
end;
```

Key	Description
...	Modify current identifier
Enter	Accept Changes

Средства автоматизации

Метод	IntelliJ IDEA	Eclipse	Sun NetBeans
Rename	Shift+F6	Alt+Shift+R	Alt+Shift+R
Extract method	Ctrl+Alt+M	Alt+Shift+M	Ctrl+Shift+M
Change method parameters	Ctrl+F6	Alt+Shift+F6	+
Encapsulate fields	+	+	+
Pull up	+	+	+
Pull down	+	+	+
Move Class	F6	Alt+Shift+V	Alt+Shift+V
Move inner to output level	+	+	+
Convert Anonymous Class to Inner	+	+	+
Extract interface	+	+	+
Extract superclass	+		+
Use supertype where possible	+	+	+
Safely delete	Alt+Delete	+	+
Make static	+		
Convert to instance method	+		
Copy	+		
Replace Method Code Duplicates	+		
Introduce variable	Ctrl+Alt+V	Alt+Shift+L	
Introduce field	Ctrl+Alt+F		
Introduce constant	Ctrl+Alt+C	+	
Introduce parameter	Ctrl+Alt+P	+	
Replace inheritance with delegation	+		
Inline	Ctrl+Alt+N	Alt+Shift+I	
Replace temp with query	+		
Replace constructor with factory method	+	+	
Generify template classes	+	+	
Migrate packages	+		

2006 год