

## ПЗ 16. Изучение встроенной поддержки рефакторинга кода в Visual Studio

**Задание.** На основе предыдущих практик (с тестовым редактором, заметками) произвести рефакторинг кода с помощью встроенного функционала Visual Studio, руководствуясь теоретической частью задания и главой «Код с душком» книги М.Фаулера «Рефакторинг. Улучшение существующего кода»

В качестве ответа приложить отчет, отражающий этапы применения рефакторинга:

- Скриншот кода ДО (с проблемой)
- Операция рефакторинга
- Скриншот кода ПОСЛЕ
- Пояснение (кратко – в чем была необходимость изменений)

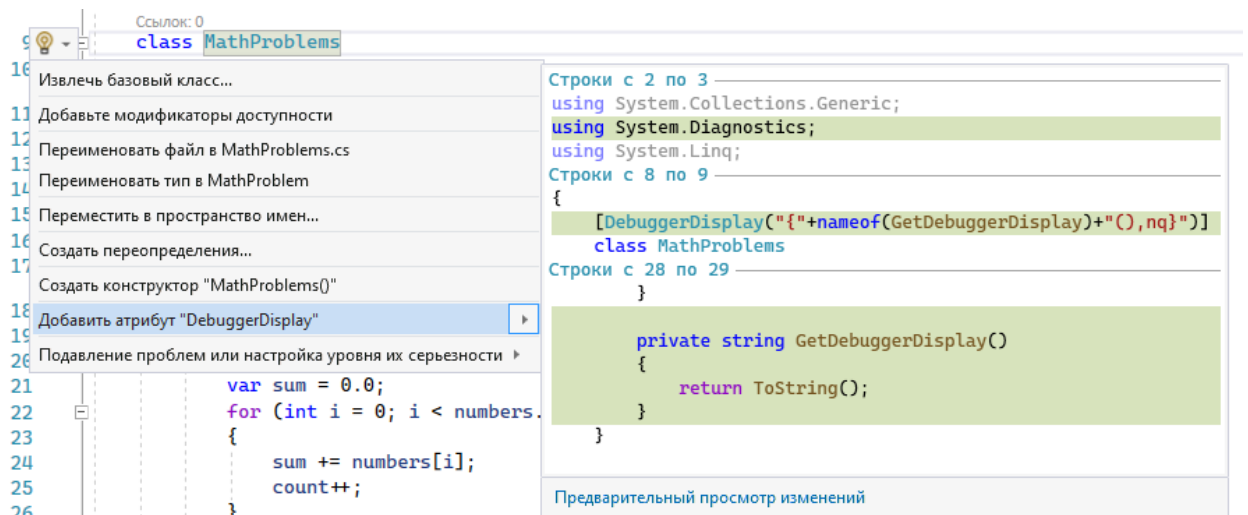
Объем отчета – минимум пять выявленных проблем и их исправление

**Примечание:** размер скриншотов строго регулировать областью проблемы. Не нужно скринить все доступное пространство.

### Теоретическая часть:

Существует много способов вызвать инструменты для рефакторинга в системе Visual Studio, включая контекстное меню, которое открывается после щелчка ПКМ, интеллектуальные дескрипторы и меню Refactor в главном меню (pro)

На рисунке ниже показано контекстное меню Refactor. Список действий, меняется в зависимости от контекста. Вызвать его можно выделив проблемный код и в контекстном меню выбрать пункт «Быстрые действия и рефакторинг» ctrl+ю

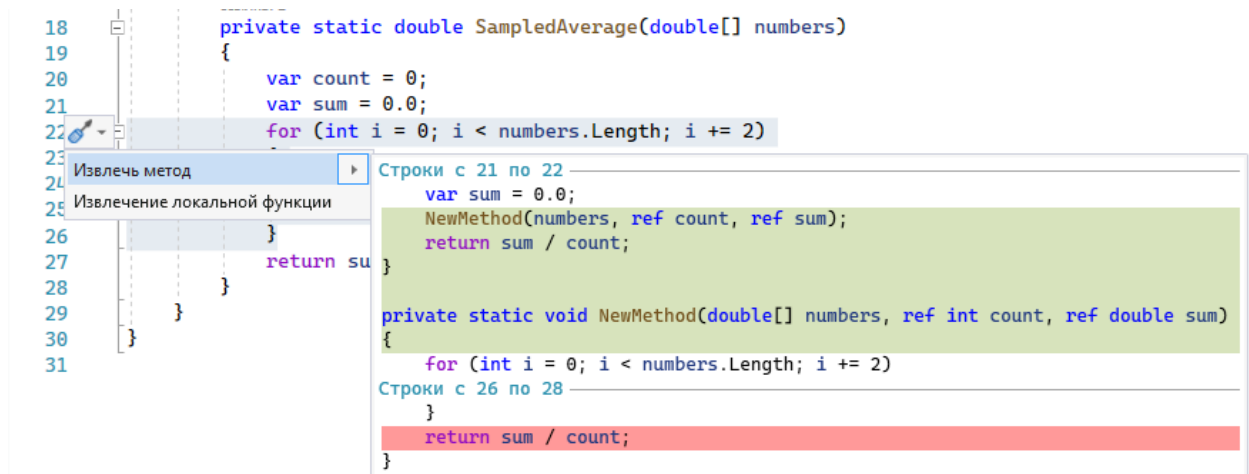


### Операции рефакторинга

#### Операция Extract Method (извлечь метод)

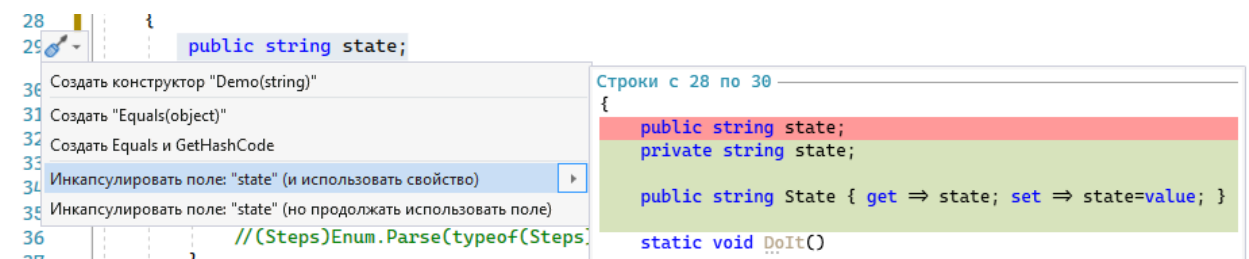
Чтобы начать рефакторинг длинного метода, лучше всего разбить его на несколько небольших частей. Для этого необходимо выделить область кода, которую разработчик хочет удалить из исходного метода, и выбрать команду Extract Method из контекстного меню. В языке C# это приведет к появлению приглашения ввести имя нового метода, как показано на рисунке ниже. Если в удаляемом блоке кода есть переменные, которые

использовались ранее, они автоматически становятся переменными в сигнатуре нового метода. После подтверждения имени нового метода его код вставляется сразу после исходного метода. Удаленный блок метода заменяется вызовом нового метода.



### Операция Encapsulate Field (инкапсулировать поле)

При рефакторинге часто требуется инкапсулировать в существующий класс некое свойство. Для этого предназначена операция Encapsulate Field. Чтобы выполнить это действие, следует выбрать переменную, которую требуется инкапсулировать, и выбрать соответствующую команду в контекстном меню. Это дает разработчику возможность указать свойство и выбрать место, в котором следует искать ссылку на него:

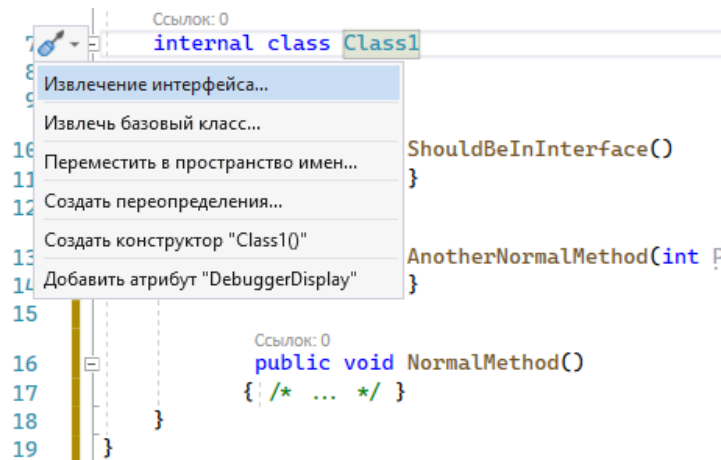


На следующем шаге после указания нового свойства следует определить, какие ссылки на переменную класса следует заменить ссылкой на новое свойство. Откроется окно предварительного просмотра по завершении поиска ссылок. В верхней панели окна размещено дерево, в котором указываются поля и методы, имеющие ссылки на данную переменную. Флажок, расположенный после каждой строки, указывает, следует ли выполнять замену. Выбрав строку в верхней части окна, необходимо установить на нее фокус в нижней панели. Как только будет обеспечена корректность каждой ссылки, инкапсуляция считается завершенной. Переменная класса устанавливается закрытой, а соответствующие ссылки обновляются.

### Операция Extract Interface (извлечь интерфейс)

По мере того как проект проходит этапы развития от прототипа на ранней стадии разработки до полной реализации или стадии роста, часто возникает необходимость выделить основные методы в виде интерфейса, доступного другим приложениям, или определить границы между непересекающимися системами. Прежде для этого приходилось копировать весь метод в новый файл и удалять его содержимое, оставляя лишь заглушку интерфейса. Операция рефакторинга Extract Interface позволяет извлечь интерфейс, используя любое количество методов в классе. При выполнении этой операции открывается

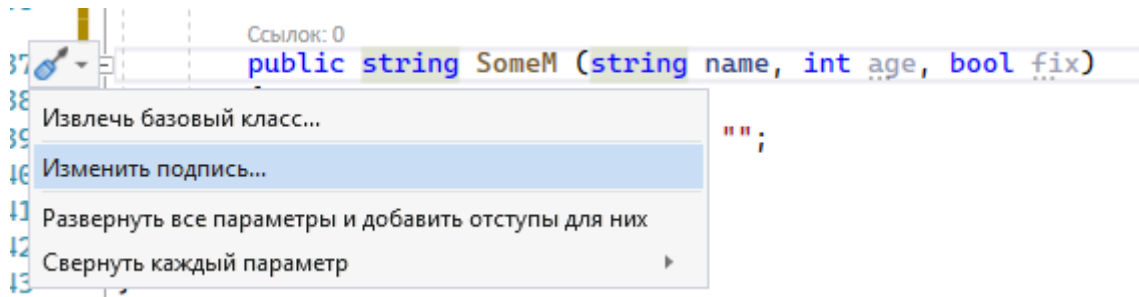
диалоговое окно, показанное на рисунке ниже, которое позволяет пользователю выбрать метод, который он хочет включить в интерфейс. После того как разработчик сделает выбор, эти методы добавляются в новый интерфейс. Кроме того, новый интерфейс добавляется в исходный класс.



После выбора команды Extract Interface из контекстного меню, появляющегося после щелчка правой кнопкой мыши, и выбора метода ShouldBeInInterface, подлежащего извлечению, в диалоговом окне Extract Interface (как показано на рисунке выше), в новом файле возникает новый интерфейс, а исходный файл обновляется.

### Операция Reorder Parameters (изменить подпись.)

Иногда требуется переупорядочить параметры. Часто это делают по "косметическим" причинам, но эта операция может помочь повысить читабельность и иногда является необходимой при реализации интерфейсов.



Диалоговое окно Reorder Parameters dialog, показанное на рисунке ниже, позволяет переставлять параметры в списке в соответствии с требуемым порядком:

Изменить сигнатуру

Окно ? X

Параметры:

Индекс	Модификатор	Тип	Параметр	По умолчанию	Место вызова
1		string	name		
2		int	age		
3		bool	fix		

↑

↓

Добавить

Удалить

Восстановить

Предпросмотр сигнатуры метода:

public string SomeM(string name, int age, bool fix)

☐ Предпросмотр изменений в ссылках

OK

Отмена

Как только требуемый порядок будет достигнут, пользователь может предварительно просмотреть результат. По умолчанию параметры в каждом вызове данного метода переставляются автоматически в соответствии с новым порядком.

## Операция Remove Parameters

Удаление параметра из метода с помощью соответствующей операции рефакторинга значительно сокращает объем поиска ошибок компиляции, которые могут при этом возникнуть. Кроме того, эта операция очень полезна при многочисленных перегрузках метода, при которых удаление параметра не может порождать ошибки компиляции. В этом случае ошибка во время выполнения программы может возникнуть только по семантической, а не по синтаксической причине.

Если параметр был удален непреднамеренно, его легко восстановить. Как указывает предупреждение, размещенное в диалоговом окне, удаление параметров часто приводит к неожиданным функциональным ошибкам, поэтому важно контролировать внесенные изменения. Для того чтобы оценить внесенные изменения, можно снова использовать окно предварительного просмотра.