

IDEA: refactoring

Java Collections
9 уровень, 7 лекция

ОТКРЫТА

— И снова здравствуйте.

— Привет!

— Сегодня я расскажу тебе про рефакторинг(re-factoring). Рефакторинг – это изменение кода проекта без изменения его функционала.

— А как такое может быть?

— Ну, в самых простых случаях – мы переименовываем переменные и/или методы. От переименования переменных программа ведь по-другому работать не станет?

— Нет, конечно.

— Вот, затем можно разбить большие методы на несколько отдельных.

Можно вынести повторяющиеся куски кода в отдельный метод.

Некоторые функции обозначить как static, а затем вынести их в утилитные классы.

Но это рефакторинг в узком смысле.

Иногда под рефакторингом понимают переписывание (улучшение) архитектуры проекта, без добавления новой функциональности. Это рефакторинг в широком смысле.

IntelliJ IDEA стала самой популярной IDE (среда разработки) именно потому, что первая внедрила очень мощные средства рефакторинга.

— Что же это за волшебные штуки такие-то?

Волшебство первое – изменение имени метода.

— Представь, что в твоём коде есть метод, который вызывается из 100-500 других мест. И ты решил сменить его имя на более читаемое. Скажем, было run(), а стало runDownloadTaskAsync(). Насколько быстро это сделать?

— Ну, сначала надо поменять имя этого метода, затем искать в программе все места, где этот метод вызывается, и менять его имя там.

— А как ты будешь его искать?

— Я бы просто запустил программу, и IntelliJ IDEA бы выдала все ошибки, где вызывается метод, которого нет.

— Хорошо, а допустим, у тебя у каждого метода есть поясняющий комментарий (JavaDoc), в котором написано что делает этот метод, и там написано его старое имя.

— Я поменяю и комментарии.

— А ведь могут еще быть переменные, название которых связано с именем метода, хорошо бы и их поменять тоже:

Было		Стало	
1	Task task = manager.run();	1	Task asyncTask = manager.runDownloadTaskAsync();

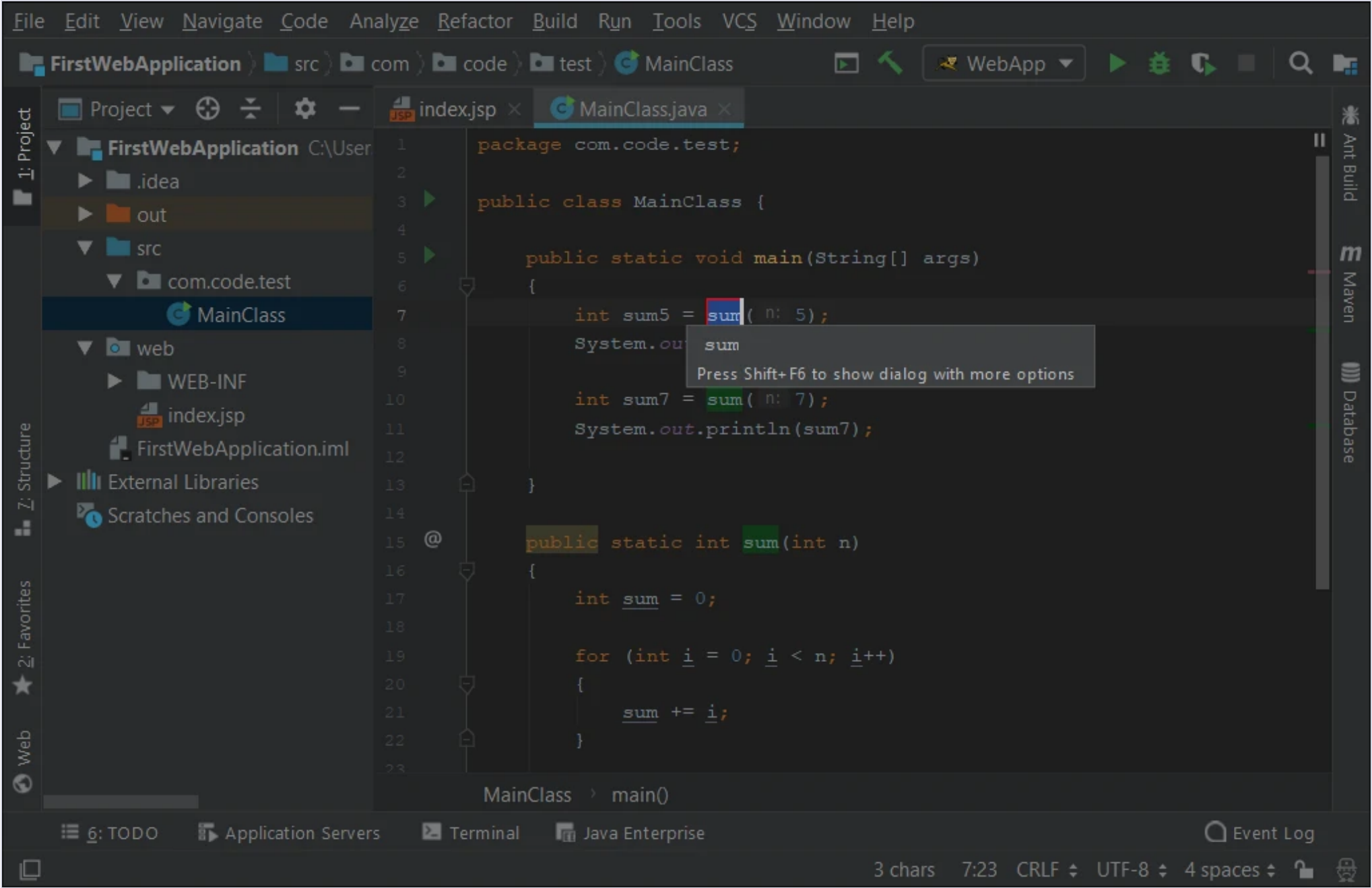
— Да, хорошо бы поменять имена и этим переменным. Не помешало бы.

— Так вот, это все можно сделать с помощью IntelliJ IDEA за пару секунд!

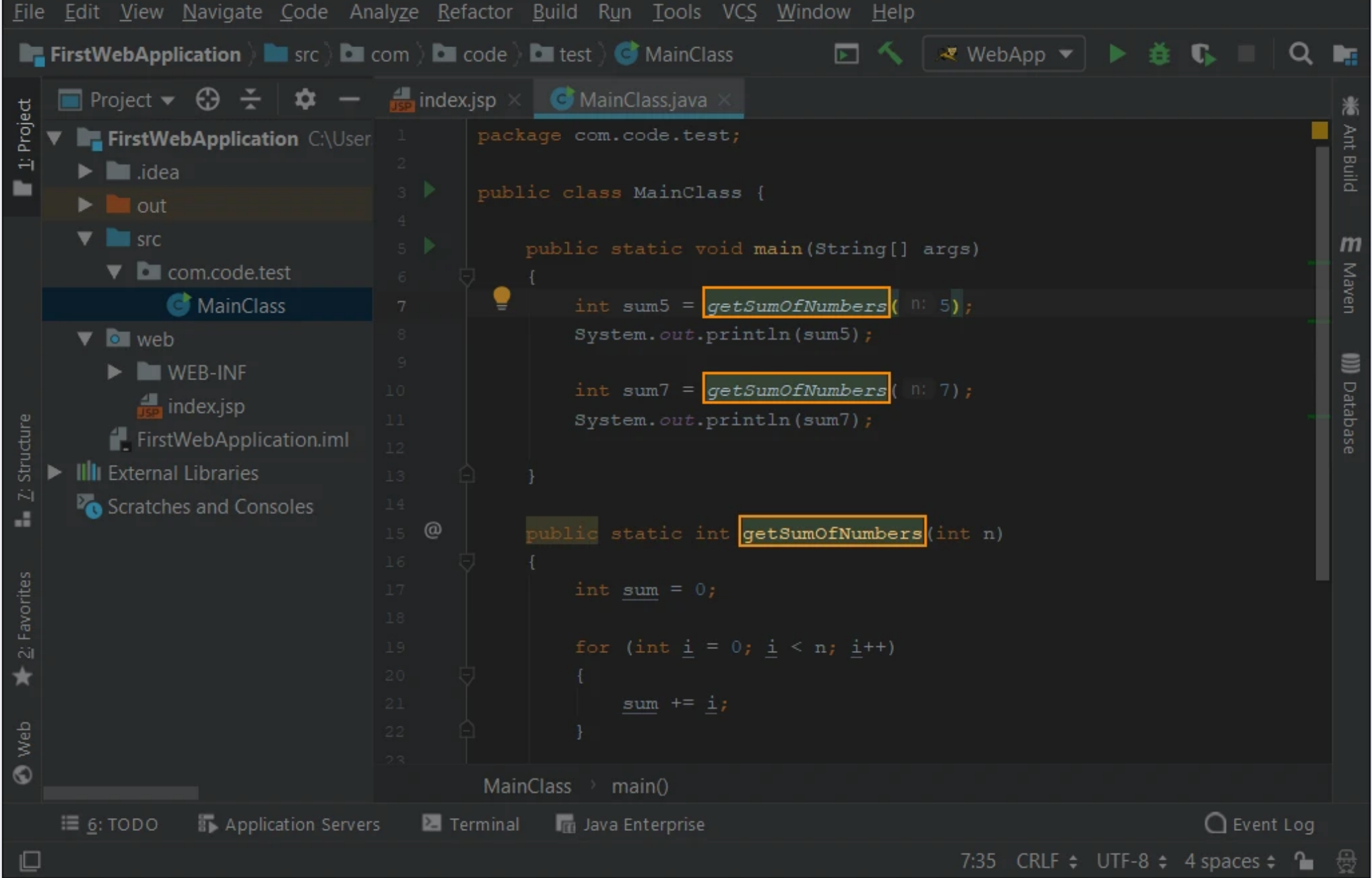
Становишься курсором на имя метода (или кликаешь по нему мышкой), затем жмешь **Shift+F6** и просто начинаешь вводить нужное имя метода.

Пример: редактирование имени метода.

Начало редактирования:



Указали новое имя:



Просто вводишь новое имя, жмешь enter и все. Все вызовы данного метода в проекте переименованы.

IntelliJ IDEA также обычно дополнительно спрашивает – надо ли переименовывать переменные и имена методов в комментариях. Достаточно просто нажать «да», и она все сделает.

При этом если до рефакторинга проект компилировался, будет и после.

— Звучит очень многообещающе.

— Кстати, имена переменных можно поменять так же. Стал на имя, просто жмешь **Shift+F6** и все – можешь вводить новое имя, и везде, где используется эта переменная, оно поменяется.

Если переменная была полем класса и у нее были getter и setter, они тоже сменят имя, чтобы соответствовать новому имени переменной.

— Попробовал с переменными. Все работает, как ты и говоришь, Элли. Рефакторинг — это круто!

— Ты думаешь – это весь рефакторинг? Да там столько всего, что мы и десятой части не коснулись.

— Ого. И что же там еще?

Волшебство второе – вынос переменной.

— Иногда в коде часто повторяются некоторые выражения, что аж хочется вынести их в отдельную переменную. Пример:

Код	
1	<code>public void printInfo(User user)</code>
2	<code>{</code>
3	<code>System.out.println(user.getProfile().getName());</code>
4	<code>System.out.println(user.getProfile().getAddress().getState());</code>
5	<code>System.out.println(user.getProfile().getAddress().getCity());</code>
6	<code>System.out.println(user.getProfile().getAddress().getStreet());</code>
7	<code>System.out.println(user.getProfile().getAddress().getHomeNumber());</code>
8	<code>}</code>

Как хочется сделать, глядя на него:

```
1 public void printInfo(User user)
2 {
3     Address address = user.getProfile().getAddress();
4
5     System.out.println(user.getProfile().getName());
6     System.out.println(address.getState());
7     System.out.println(address.getCity());
8     System.out.println(address.getStreet());
9     System.out.println(address.getHomeNumber());
10 }
```

— Ага.

— А ведь бывает и куда более сложный код, в котором очень много повторяющихся частей.

Программисты так не специально пишут. Но часто случается, что надо в метод что-то дописать, потом еще что-то, и со временем количество избыточных частей зашкаливает.

К тому же, создавая отдельную переменную, мы может дать ей правильное имя и улучшить читабельность кода.

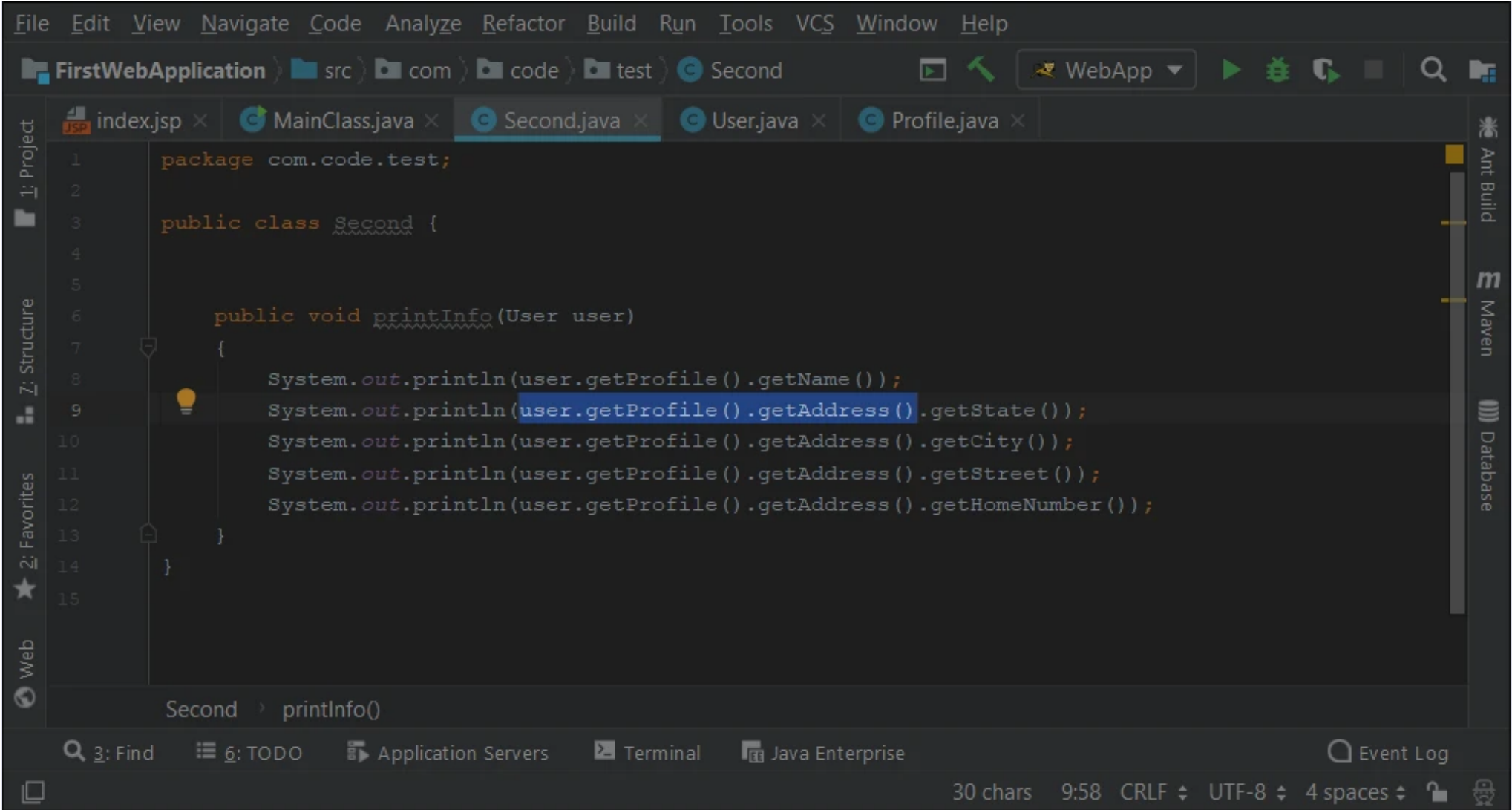
Например, в примере выше, речь может идти не об адресе проживания, а об адресе, куда можно обратиться в случае, если с человеком что-то случится. Тогда можно назвать переменную не address, а, скажем, supportAddress. Тогда, программисту, который видит этот код в первый раз, будет понятнее, что тут происходит.

— Да, согласен, добавление таких переменных имеет смысл.

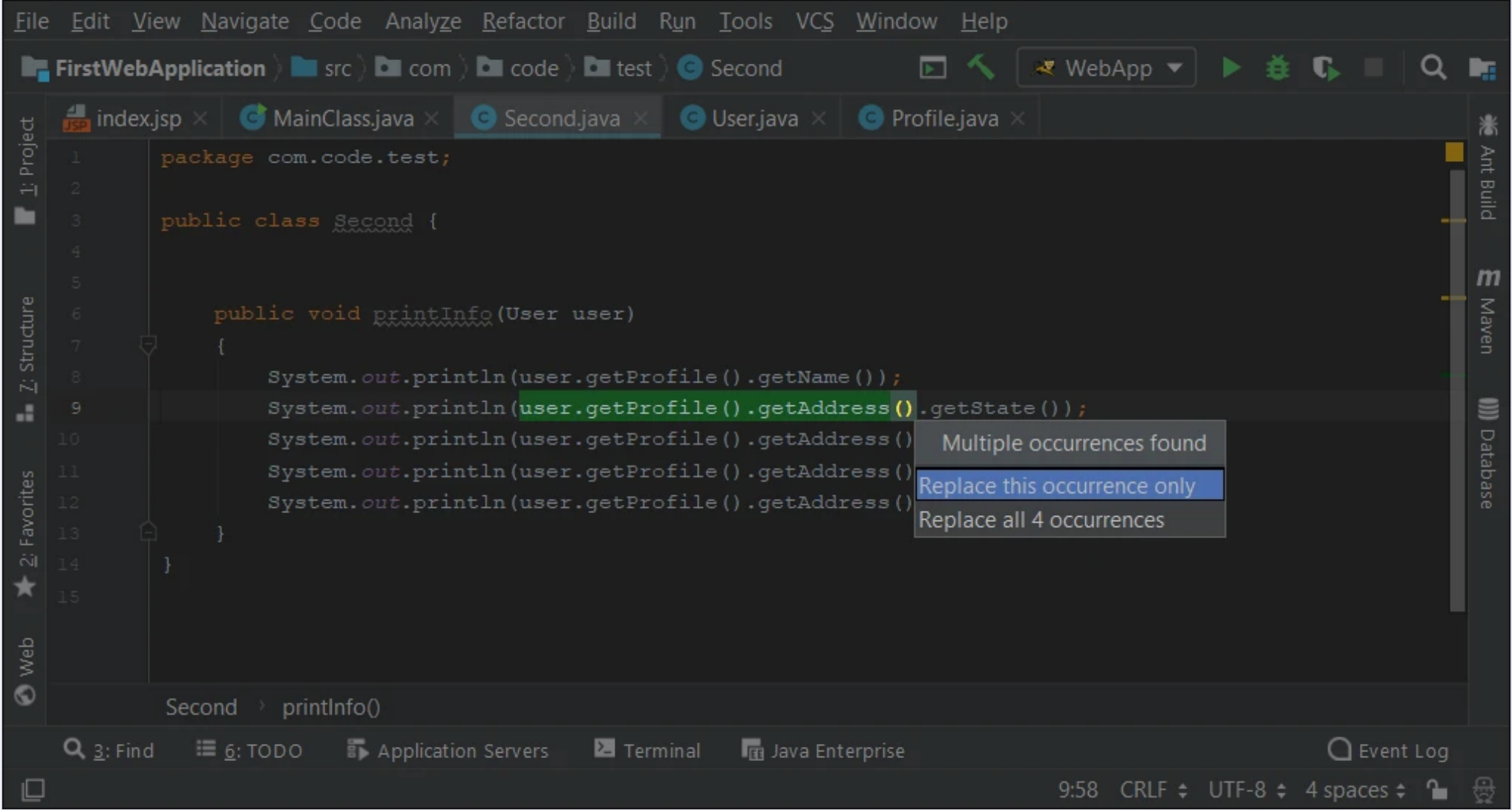
Так как это сделать?

— Вынести какое-то выражение в отдельную переменную очень просто.

Шаг 1 – выделяем мышкой выражение.



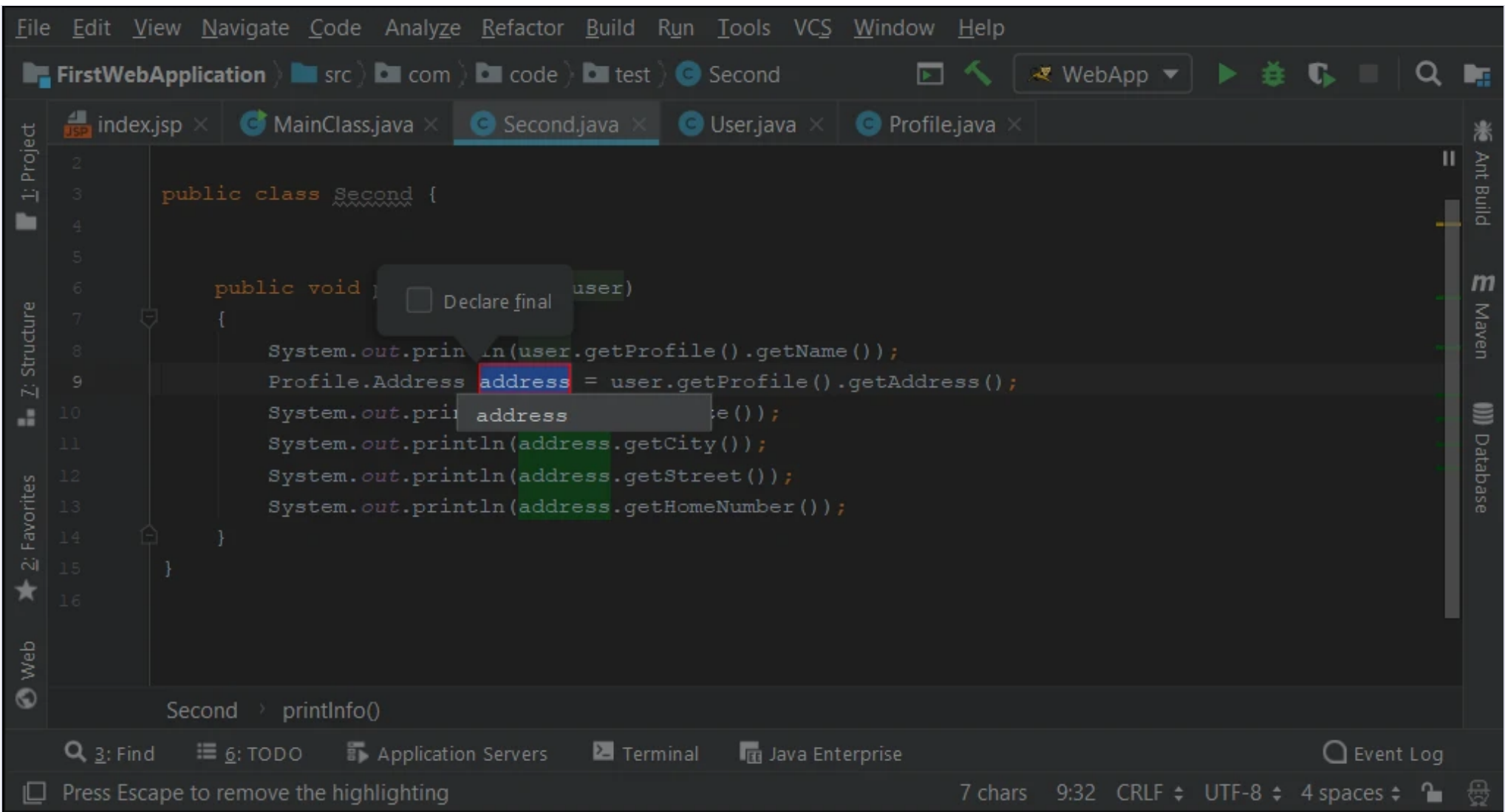
Шаг 2 – нажимаем Ctrl+Alt+V



Появляется окно, где IntelliJ IDEA нас спрашивает – заменить только одно выделенное выражение или заменить все выражения (4 шт.)?

Выбираем второй пункт – заменить все вхождения (Replace all 4 occurrences)

Шаг 3 – нажимаем Enter.



IntelliJ IDEA предлагает ввести имя переменной. И сразу предлагает свой вариант имени. Не так уже он и плох, да?

— Ага. В самую точку. Мы тоже собирались назвать переменную address. А как она догадалась?

— По имени последнего метода в выражении – он возвращает адрес, значит, скорее всего, в переменной хранится именно адрес.

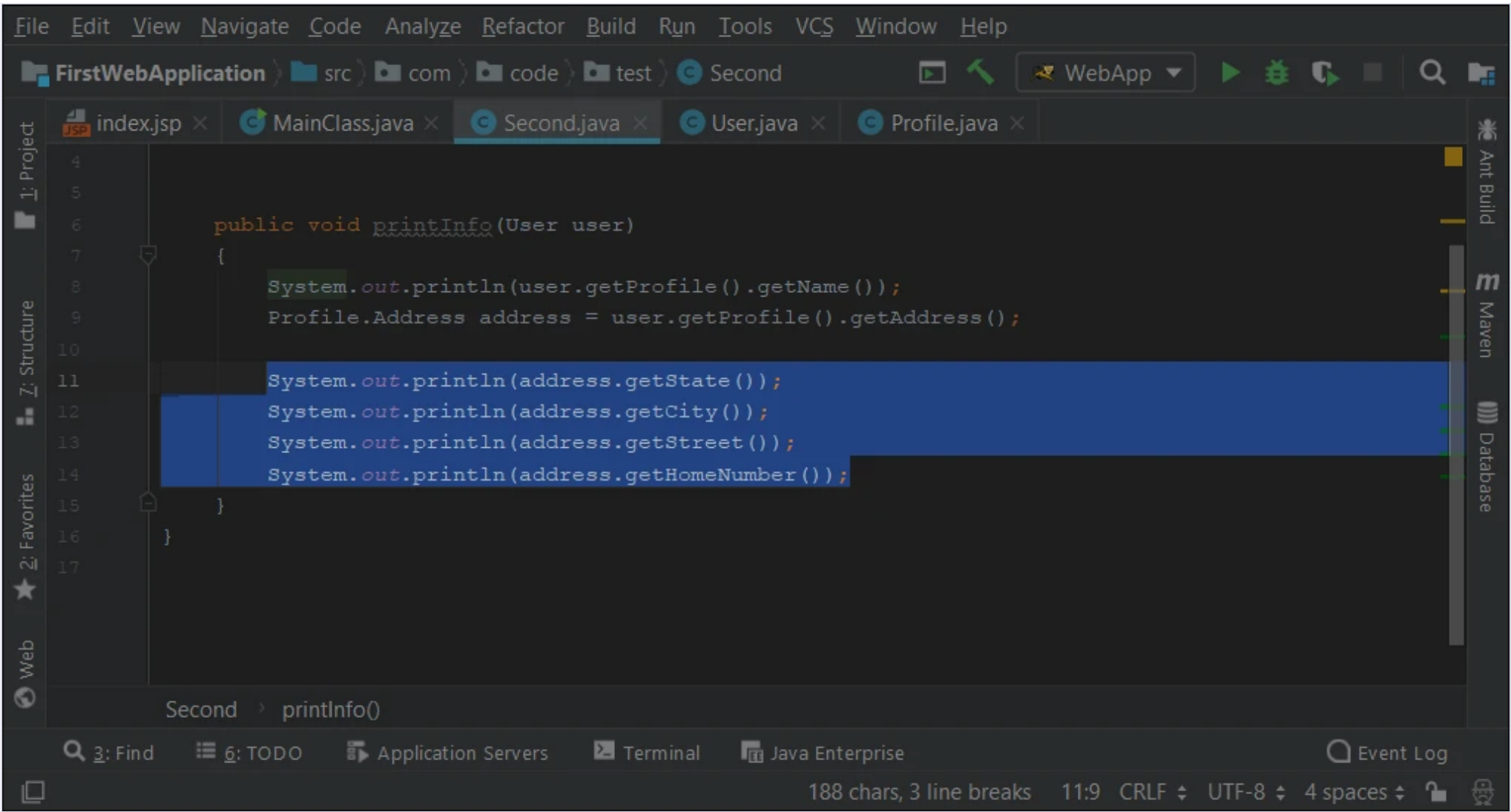
— Отлично получилось. Классная штука, Элли.

Волшебство третье – вынос кода в отдельный метод.

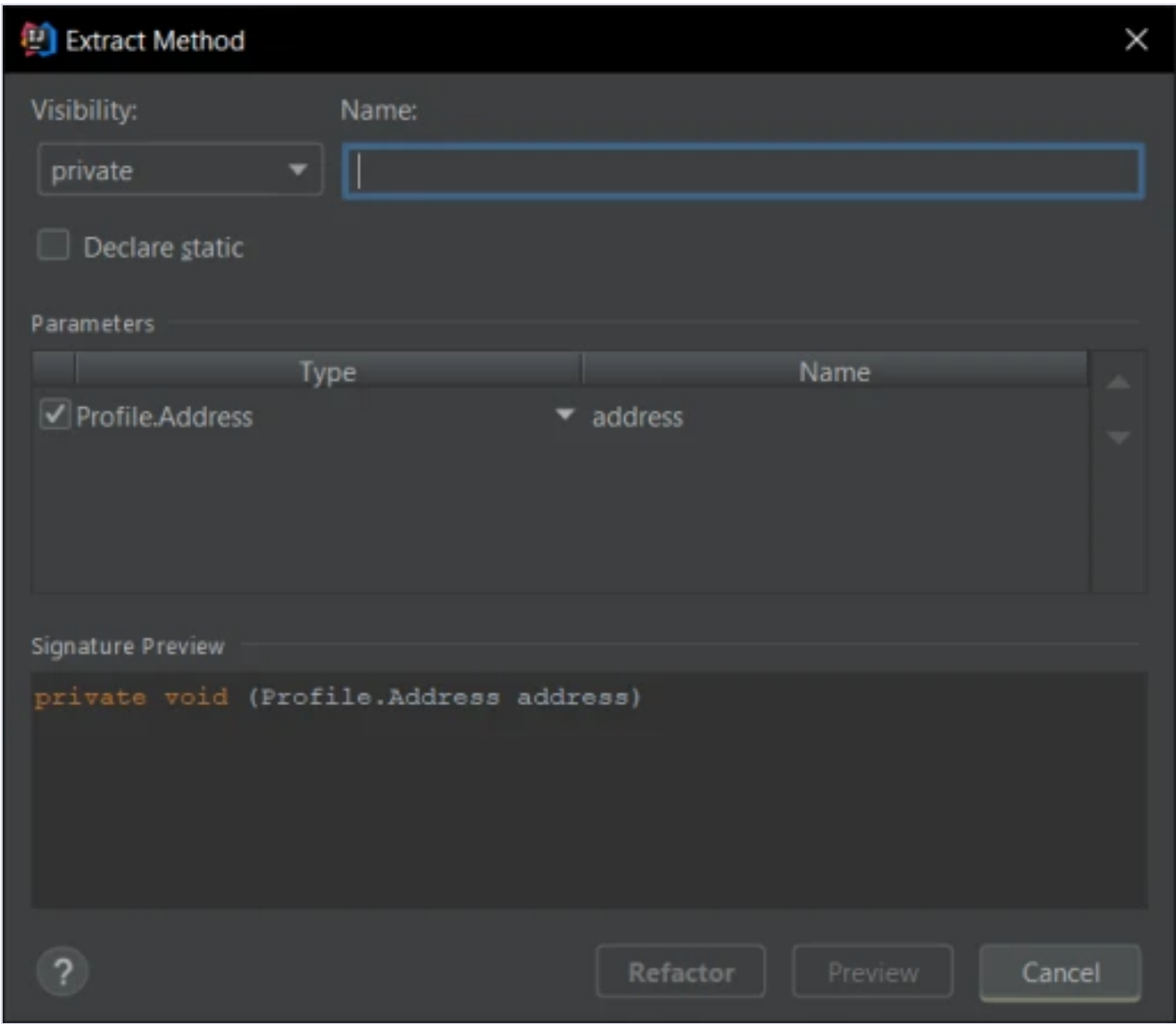
— А ведь можно было сделать и по-другому. Можно было объявить новый метод, например, **printAddress** и вынести весь этот код в него.

Давай, так и попробуем сделать.

Шаг 1 – выделяем 4 строчки кода, которые используют переменную address:

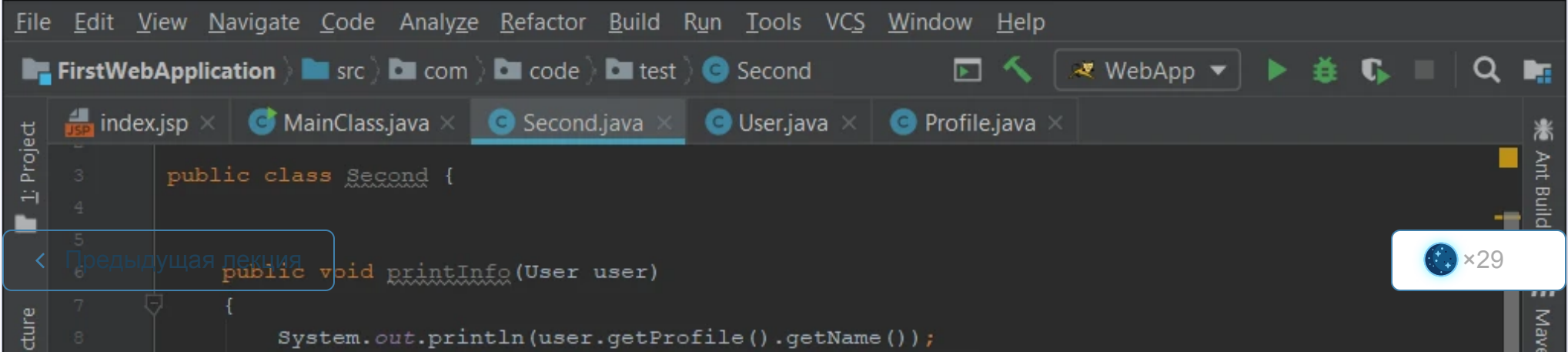


Шаг 2 – нажимаем Ctrl+Alt+M



IntelliJ IDEA сама определила, какие переменные будут нужны этому методу, и предлагает нам, как он будет выглядеть. Осталось только ввести имя метода.

Шаг 3 – вводим имя метода — printAddress и жмем Enter.



− +76 +

Комментарии (34)

популярные новые старые

JavaCoder

Введите текст комментария

Макс Дудин Уровень 40, Калининград, Россия

5 августа, 17:23 ...

Да блин.... и только сейчас..... уже столько раз бы мог...

Ответить

− 0 +

Iliia lenskii Уровень 32, Санкт-Петербург

23 сентября, 01:02 ...

Этим фсё равно никто не пользуется, потому что это не встречается в коде

Ответить

− 0 +

Алексей Мирный Backend Developer

20 января 2021, 10:19 ...

Очень понравилась статья

Ответить

− +3 +

Konstantin Уровень 41, Одесса

18 сентября 2020, 11:05 ...

1. Волшебства в IT нет.
2. Инструменты рефакторинга стандартны для большинства IDE.

P.S. Сделали из Идеи неподражаемый фетиш. Радужный единорог б@@@@.

Ответить

− +4 +

Тот кто в танке Java программист

2 ноября 2020, 11:27 ...

справедливости ради, IDEA мощная штука.

Ответить

− +5 +

Konstantin Уровень 41, Одесса

2 ноября 2020, 12:48 ...

Вопрос не в мощности, а в том что из Идеи сделали гламурный фетиш, считая что она единственная и неподражаемая в своём роде IDE. Это не так. Учитывая что это проприетарный продукт, то его просто разрекламировали маркетологи компании JetBrains.

Более того, всем новичкам пихают на всех учебных курсах, подсаживая как наркоманов на "единственно-правильную" мощную IDE.

Ответить

− 0 +

Тот кто в танке Java программист

2 ноября 2020, 20:50 ...

какую IDE порекомендуете?
Курс тут я заканчиваю уже, вполне можно свой проект на другой попробовать. Сравнить с высоты своего небольшого опыта так сказать)

Ответить

− +3 +

Konstantin Уровень 41, Одесса

2 ноября 2020, 22:35 ...

Я использую Eclipse и вполне доволен. Некоторые вещи, как по мне, реализованы сильно удобнее чем в Идее, например подключение Tomcat и т.п. серверов.

Установить или скачать "Eclipse for JavaEE Developers" - это предсобранный пакет для работы как с Java SE, так и Jave EE.
Но для успешной работы с затмёнкой нужно поучиться и подключить плагины.

Плагины, думаю разберётесь какой за что отвечает:
checkstyle
darkest dark
pmd
eclipse-pmd
jboss-tools (можно также посмотреть плагин по freemarker от jboss, но это когда нужно будет разбираться с freemarker-ом)
spotbugs
spring-tools 4 - ставить после 3-ей версии, иначе будет конфликт
spring tools 3 addon for spring tools 4 - ставить перед 4-ой версией, иначе будет конфликт
markdown text editor
jacoco(already installed)
infinittest
dbeaver
ANSI Escape on Console
ResourceBundle Editor
<https://www.eclipse.org/mat/> - анализатор дампов кучи полученных с помощью visualVM -
<https://visualvm.github.io/>
Enhanced Class Decompiler - декомпилятор
grep console - парсит вывод в консоль. Удобно для раскрашивания логов выводимых в консоль.
Lombok - полезная штука упрощающая написание шаблонного кода

Курс, который мне понравился - "Learning Path: Beginning Java, 2nd Edition" . Найдёте на каком-нибудь треккере. Там в составе идёт курс по Eclipse.

Книг не читал, поэтому сами что-то ищите.

Сразу хочу оговорить, чтобы потом не было вопросов - проблемы с установкой и обновой плагинов бывают довольно часто(гарантировано голова болеть будет всегда). Например я некоторые плагины не мог установить потому что версия JDK была 11, а когда поставил 14-ую, то проблема ушла.
Также обновлять клиент придётся вручную и переустанавливать плагины(по крайней мере у меня это регулярная проблема). Этот процесс можно упростить добавив нужные плагины в Favorites и сохранив их список в аккаунте на сайте eclipse.org. В дальнейшем в Eclipse Marketplace можно будет выбрать для установки сразу все, ну или почти все плагины пачкой.

Просто не будет, но оно того стоит.

Ответить

+7

Тот кто в танке

Java программист

2 ноября 2020, 23:44

спасибо!

Ответить

0

Konstantin

Уровень 41, Одесса

3 ноября 2020, 09:43

Пожалуйста.

Вот ещё ссыль на доку по разным плагинам, включая базовый интерфейс - <https://help.eclipse.org/2020-09/index.jsp> .

И ещё один плагин для работы с UML-диаграммами - **Papyrus**. Устанавливать лучше на отдельный клиент для JavaEE Developer-ов. Одной из удобных фич Papyrus-а является возможность реверс-инжиниринга исходного кода проекта в UML-диаграммы, и что более интересно из UML-диаграмм в код :) (в теории, при определённой практике, можно упростить генерацию исходников начального проекта). Очень помогает в понимании структуры проекта.

А вообще - Eclipse - швейцарский нож. В нём столько свободного функционала, который проприетарным IDE и не снился, в том числе и Идее, что можно изучать годами. Кроме того, если мне нужен будет инструмент под другую задачу, то я его просто скачаю, а не буду покупать очередной пакет от проприетарной компании.

Успешного освоения.

Ответить

0

Тот кто в танке

Java программист

3 ноября 2020, 09:48

Ответить

0

Konstantin

Уровень 41, Одесса

3 ноября 2020, 11:08

Пожалуйста.

Ответить

0

Justinian

Judge в **Mega City One** MASTER

6 января 2021, 12:59

// считая что она единственная и неподражаемая в своём роде IDE классика манипуляции.

Самому придумать утверждение, самому опровергать и используя его обвинять кого-то. Отлично.

Лично я не встречал ни одного человека, который считал Идею "единственной ИДЕ, волшебным единорогом". Фетишем? Возможно. Как нож является фетишем для повара, а авто является фетишем для водителя, меч для самурая и тд.
Это нормально профессионалу относится с уважением к своему инструменту

Кей Хорстман, Роберт Мартин, Мартин Фаулер, Евгений Борисов , сотни авторов книг, тысячи докладчиков на крупнейших джава конференциях, десятки тысяч синиоров, техлидов и архитекторов, подавляющее большинство отрасли использует Идею, это просто факт. Доля эклипса неуклонно снижается с годами.

При этом, эклипс имеет свои преимущества, большую гибкость в расширениях, там где нетривиальные задачи на джаве, там с помощью джавы сделать расчеты симуляции ядерного взрыва или еще что, в основном в научной сфере, или в коммерческих специфических кейсах, то да, Эклипс может быть хорошим выбором.

Но большинство джавистов будут на работе использовать Идею, она для веб энтерпрайза безальтернативно удобная.

Идея, Эклипс, Нетбинс и еще с десятков ИДЕ, это лишь инструменты. Для каждой задачи есьт свой инструмент.
Web проекты и BE enterprise это Интеллидж Идея, и маркетологи здесь не причем, бизнес считает деньги, если бы Эклипс был бы хоть даже на 10-15% хуже Идеи, никто бы Идею не пользовался, бизнес (клиенты) бы сказал разработчикам "используйте Эклипс!". Но разрыв гораздо больше, поэтому те ктосчитает деньги, и знает что за одно и то же время, разработчики на эклипсе сделают меньше чем те, кто на Идее, и начал внедрять Идею, которая продолжает увеличивать долю на рынке.

Не благодаря маркетологам, а благодаря своим потребительским качествам.

Я использовал и то и то, для стандартных задач и для веба Идея на порядок удобней, я использую то что удобней. Станет завтра Эклипс удобней буду использовать его.

Ответить

+8

Konstantin Уровень 41, Одесса6 января 2021, 15:27

Не будете вы использовать Eclipse. Потому что считаете IDEA лучше, о чём так горячо писали.

В попытке обвинить меня в манипуляции, вы сами отманипулировали предлагая в качестве подтверждения свои слова о миллионах пользователей, которые безусловно пользуют IDEA, потому что она выгоднее для бизнеса. Хотя вот этот отрывок меня удивил - "если бы Эклипс был бы хоть даже на 10-15% хуже Идеи, никто бы Идею не пользовался". Т.е. если Eclipse хуже чем нынешняя IDEA, то IDEA никто бы не пользовал? Или что-то не так или одно из двух. Да и проценты видимо среднепотолочные.

Все IDE одинаковые и имеют в большинстве своём одинаковый функционал, где-то больше, где-то меньше. Они даже плагины имеют одни и те же. IDEA проприетарный продукт и он рекламируется в том числе через обучающие курсы вроде этого, предоставляя новичкам, как наркоманам, первую дозу-ключик на полгода бесплатно. Это действия маркетологов, а не самураев. Маркетологи у проприетарной компании есть и продукт они свой делают не за ради красивых отзывов, а за бабло. А в капиталистическом обществе, в котором погрязла вся планета, бабки это единственный достойный внимания фетиш и ценность. А потому IDEA будут продвигать всему доступными способами, вкладывая в умы, что это "люччая" IDE, а потому достойная оплаты. Ведь если вы не будете считать что она "люччая" вы ведь не заплатите за неё деньги? Или заплатите? Ну тогда вы платите за плохой продукт, просто потому что... что? Зачем вы платите за плохой продукт? Платить можно только за "люччее".

(Продолжение следует)

Ответить

0

Konstantin Уровень 41, Одесса6 января 2021, 15:28

Я обучался на трёх проектах и все они втюхивают только IDEA, даже не предлагая вариантов. Просто размазывают сопли о том что она лучше. Самое интересное мне всегда было интересно узнать, а какие аргументы будут приведены в пользу IDEA. Ответы прям парад лени и гламура:

1. На первом месте конечно же для любого - "Alt+Enter" - режим отключения мозга. Особенно падки юные падаваны.
2. На втором месте - "потому что мерседес". Тут сразу же надо предположить что высказывающийся является счастливым обладателем гламурного фетиша - макбук от компании Огрызок.
3. На третьем - "я считаю IDEA лучше потому что привык и мне в других IDE, в частности Eclipse неудобно". Ну так это - если тяжело выучить другое расположение кнопочек, так это не инструмент гавно, а элементарная лень понять как он работает.

А потом, когда новички привыкнут к Alt+Enter, потому что так проще, а не "лучше", как вы выразились, когда IDE сама пишет за них код, то начинают бегать по интернету за следующей дозой-ключиком, потому что платить не хочется, а условный рефлекс нажимания Alt+Enter выработался. А если, вдруг, возникнет вопрос о переходе на другую IDE, то ничего не произойдёт. Знаете почему? А потому что лень. Я таких примеров ленивого разума видел немало, когда у человека начинается истерика и паника когда он видит другие кнопочки.

Потому и работодатели предпочитают IDEA - потому что она стандартизирует код за программиста. А программисты превращаются в кноподавов. Хотя профессия программиста уже давно стандартизирована и настоящие программисты пишут языки, а не фреймворки, а тем более не пользуютс фреймворками.

Моё мнение - наркоманы пусть бегают за дозой дальше.

Те же кто понимает язык и продолжает в нём развиваться будут писать хороший код даже в блокноте.

(Продолжение следует)

Ответить

0

Konstantin Уровень 41, Одесса6 января 2021, 15:28

Вот например, если JavaRush наконец-то запилит плагин для задач не только для IDEA, но и хотя бы для Eclipse и таким образом можно будет делать выбор, вот тогда можно будет поговорить о непредвзятости. Но этого не будет. Знаете почему? Потому что курсу 8 лет, а они его до сих пор обновить не могут. Как пример - до сих пор не могут доделать висящие в ветке квестов "SQL+Hibernate" и "JSP+Servlets". С чем это связано - только им известно, хотя конечно

предположения есть.

Если вы добрались до конца текста, не перепрыгивая в конец, и смогли его осилить с начала и до конца, то поздравляю. Надеюсь я смог донести свои мысли по данному поводу и вы не посчитаете что я пробовал вами манипулировать и поднять революцию из одного поста под никому неизвестной статьёй о проприетарной IDE. Хотя как знать - "Из искры возгорится пламя" :). Что врядли, не те люди, не то время.

Спасибо за внимание.

Ответить

+2

Justinian Judge в Mega City One MASTER

6 января 2021, 16:01

1 Не будете вы использовать Eclipse. Потому что считаете IDEA лучше, о чём так

я использую, то что лучше, будут задачи где Эклипс будет лучше, я буду использовать Эклипс. У меня не подгорает от какой-то определенной ИДЕшки, я использую те инструменты, которые больше подходят и более эффективны для той или иной задачи. Так что, твои экстрасенсорные способности - знать кто и что будет использовать здесь дают сбой

1 Все IDE одинаковые

все автомобили одинаковы. Заз 968. Ока. Мерседес. Бугатти Вейрон. Автобус. Белаз. Они же автомобили, с чего бы им быть разными. Ну ну.

1 Потому и работодатели предпочитают IDEA - потому что она стандартизирует код
2 программиста. А

ты работодатель? Сколько проектов закрыл, каких масштабов, когда у тебя было IPO, на какой бирже и какие его результаты, сколько и кто у тебя работает? Откуда тебе знать почему и что предпочитают работодатели?

1 А программисты превращаются в кнопкодавов.
2 ...
3 Моё мнение - наркоманы пусть бегают за дозой дальше.

Фаулер, Борисов, Хорстман, Мартин кнопкодавы "наркоманы бегающие за дозой"? М-да уж.

Во всем нужен баланс, слепо во что-то верить плохо, но повторюсь, то что ты описываешь существует либо в твоей голове, либо разве что у некоторых только приступивших к изучению джавы, а это не самое страшное, что есть в их головах, и оно со временем уйдет.

Я знаю десятки, сотни джавистов, от синиоров и архитекторов до очень известных личностей, из всего многообразия, на эклипсе видел несколько преподав 50+ в вузе, +на продакшене 1 кейс знаю, когда клиент захотел эклпис, но что за проект детали не помню. 3 человека из сотен.

Идея это всего лишь инструмент.

Tools doesn't matters. People do.

Ответить

+4

Konstantin Уровень 41, Одесса

6 января 2021, 18:07

Ну вот опять. Расписал всё подробно, а толку ноль.

В ответ только пропаганда "люччего" :). Не будете вы использовать Eclipse, потому что уже пользуетесь "лючим".

У меня не "подгорает" от IDE, у меня подгорает от того что из неё сделали гламурный фетиш, а из программистов сделали блондинок рассказывающих какую классную новую IDE они купили.

Программисты должны быть оплотом разума и холодного расчёта, а не хипстерами, которых из них делает мода.

Не путайте предназначение разных видов машин с понятием самой машины. Писать код можно и в блокноте и в Word-е, а я вам об IDE говорил, а не обо всех видах редакторов текста. Как вы там говорили - подмена понятий, манипулирование? Ну-ну.

А вы работодатель? Раз можете так свободно рассуждать о том что выгодно бизнесу? Думаю если бы были работодателем, то врядли зашли на этот ресурс под эту тему. У работодателя есть более интересные занятия чем разъяснять о преимуществах "самой люччей".

Даже не знаю откуда вам известно какими IDE пользуются перечисленные люди. но даже если и так, то почему я должен их считать авторитетами в пользовании IDE? Я посмотрел не один десяток курсов в которых IDEA встречалась 1 из 10. И что теперь? Эти четверо круче всех остальных? Подмена понятий. Вы всё ещё пытаетесь манипулировать начав с этого эту дискуссию.

А за дозой бегают как раз новички. Подсевшие её просто покупают без бега.

Баланс будет только при одинаковых условиях распространения IDE. А без этого только дисбаланс. Так что не надо делать из себя помимо программиста ещё и психиатора или экстрасенса и пытаться заглянуть мне в голову. Заблудитесь.

Десятки, сотни "от синиоров и архитекторов"... МИЛЛИОНЫ! и "Имя им Легион" :).

Пафосно, но может для детей сойдёт :).

В остальном скучно с вами. Вы ни в чём меня не убедите. Аргументов не хватит. Вы и те что имели по второму кругу пустили и использовали мои слова извратили не поняв.

Не смею вас отвлекать от ваших более продвинутых сотен сеньоров и архитекторов :).

Ответить

0

Андрей

Уровень 41, Москва, Россия

15 декабря 2021, 00:38

Думал как раз об альтернативе идее, но мне не нравится, когда голова болит.

Ответить

0

SuperFox

Уровень 41, Москва, Россия

EXPERT

18 марта, 19:58

1. Волшебства в IT нет.

Тогда зачем у меня лежит бубен перед компом?...

Ответить

+1

skrskr

Уровень 39, Санкт-Петербург, Россия

6 февраля 2020, 02:16

Для понимания насколько мощные инструменты рефакторинга в IntelliJ IDEA.
Ходил я в 2019 году на конференцию для джавистов JPoint, там была куча докладов, которые я наполовину понимал (Про Jenkins, Kafka, Netflix Eureka, Spring и тд). Один из докладчиков был действующим разработчиком IntelliJ IDEA, и его доклад был на тему рефакторинга в нашей ide. Суть примерно такая - IntelliJ IDEA позволяет вам делать рефакторинг только горячими клавишами или инструментами ide на 95%+, то есть вы практически не пишете новый код ручками, максимум имена переменным или методам даёте. Так что хорошее знание внутренних инструментов вам сильно поможет.

Ответить

+11

Jaroslav Melnychenko

Уровень 41, Харьков

15 мая 2020, 15:46

Оно? <https://www.youtube.com/watch?v=C5eD-K8AO3o>

Ответить

+11

skrskr

Уровень 39, Санкт-Петербург, Россия

27 мая 2020, 23:17

Да, всё верно

Ответить

+3

Илья

Уровень 24, Запорожье, Украина

3 ноября 2021, 14:52

Отличное видео!)Частью фич уже пользовался, но узнал для себя много нового и полезного)

Ответить

+2

Andrey Morozov

Уровень 40, Москва, Россия

20 ноября 2018, 20:11

Выделяем кусок кода, ПКМ > Refactor > Extract > и там указаны все варианты с горячими клавишами (которые почему-то не всегда отрабатывают, так что через контекстное меню надежнее)

Ответить

+11

Павел Х.

Team Lead в Иннотех

EXPERT

20 декабря 2018, 20:41

Советую поставить плагин Key Promoter X. Он будет подсказывать горячие клавиши.

Ответить

+11

Вадим

Уровень 40

22 февраля 2018, 19:38

Это всё уже было когда проходили рефакторинг

Ответить

+2

NodeOne

Уровень 41

EXPERT

27 марта 2019, 15:28

эх... мне бы вашу пясть, да хоть половинку(

Ответить

+5

Ro Do

Уровень 40, Киев, Украина

13 декабря 2017, 17:25

Полезно.

Ответить

0

Паша Дворяк

Уровень 40, Киев, Украина

12 декабря 2017, 01:14

Странно, что здесь нету комментариев о полезности этой статьи на 39 уровне)

Ответить

+8

Валера

Уровень 40

16 декабря 2017, 23:12

Потому что в отличии от дебагера и поиска и копирования (F5) класса, рефакторинг использовался почти никогда.

Ответить

Вадим

Уровень 40

22 февраля 2018, 19:38

Устали комментировать :D

Ответить

yury

Уровень 41, Москва, Россия

26 сентября 2019, 16:50

Проходили же уже рефакторинг раньше. И даже блок интересных задач решали на эту тему.

Ответить

Хорс

Уровень 41, Харьков

6 октября 2020, 16:29

потому что писать одинаковые строки- плохая практика)))

Ответить

+3

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка

JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский

▼

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

Доступно в Google Play

Загрузите в App Store

"Программистами не рождаются" © 2022 JavaRush