JavaSpecialists

IntelliJ Super Productivity in 45 Minutes

Dr Heinz M. Kabutz

Last Updated 2023-10-04

Why IntelliJ IDEA?

◉ **My story**

- Started with Borland JBuilder 1997

- Then used Eclipse for a year in 2002

- Needed something to work with horrible messy Java code
  - And create some Swing GUIs at the same time

- Downloaded IntelliJ IDEA
  - No free version at the time
  - Used it for 30 days
  - Bought it

- Paid twice
  - Once for the license and then reduced hours worked
    - But work was far more pleasant, less frustration, better life

Download JetBrains Toolbox

◉ **Keeps the IDEs up to date**

   – https://www.jetbrains.com/toolbox-app/
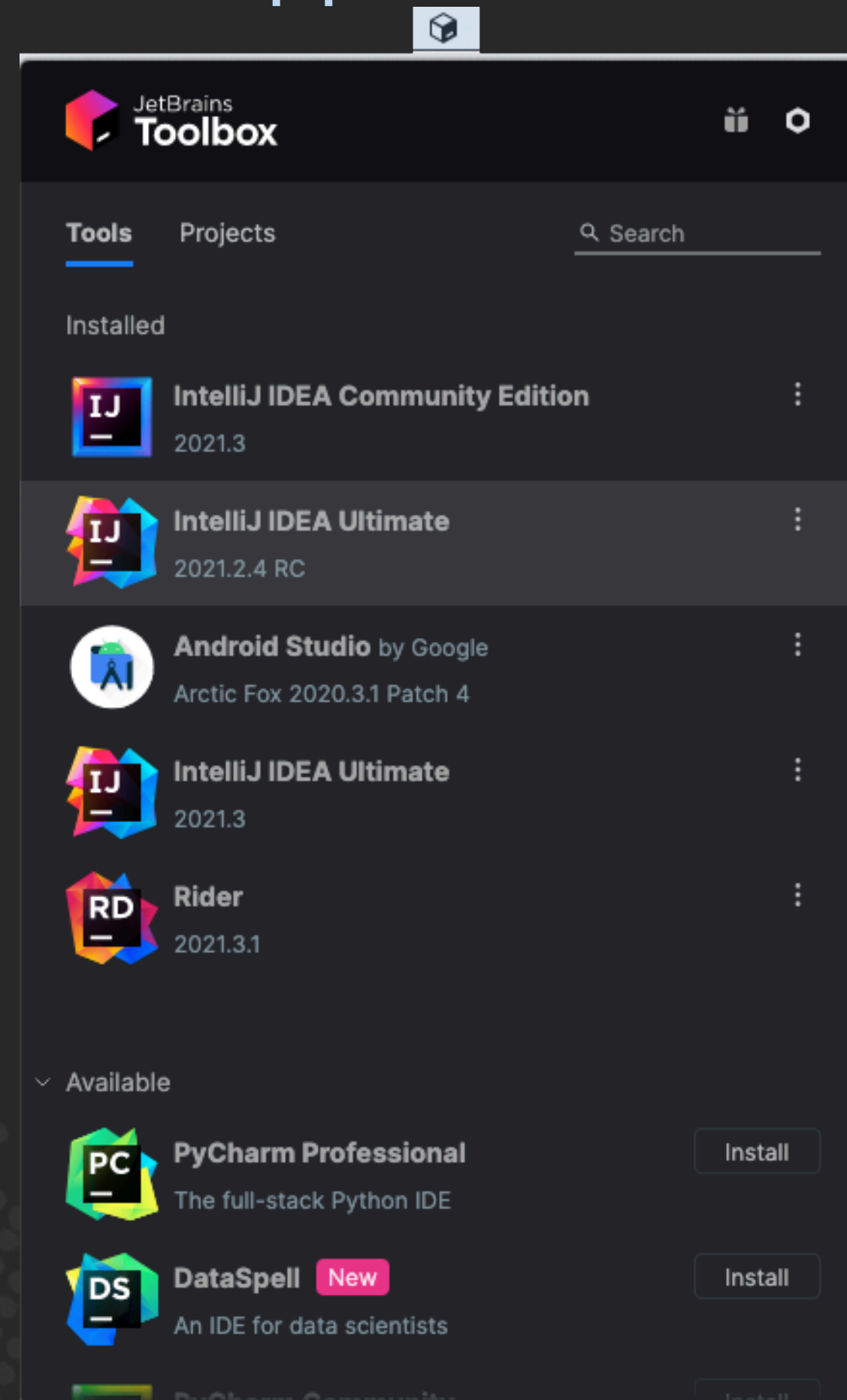
Versions of IntelliJ

◉ **Ultimate Edition**

– Paid version, has many great features

– Early Access version is integrated with AI

• See the keynote from yesterday

◉ **Community Edition**

– Free version, excellent tool for editing Java code

– This is what we will be using today (mostly)

Setting up IntelliJ IDEA

◉ **Appearance & Behavior**

– Darcula Scheme most popular, but errors harder to see

– Easy to switch between schemes with Ctrl + `

◉ **Keymap, Mac OS X**

– IntelliJ IDEA classic has more finger friendly shortcuts on the Mac
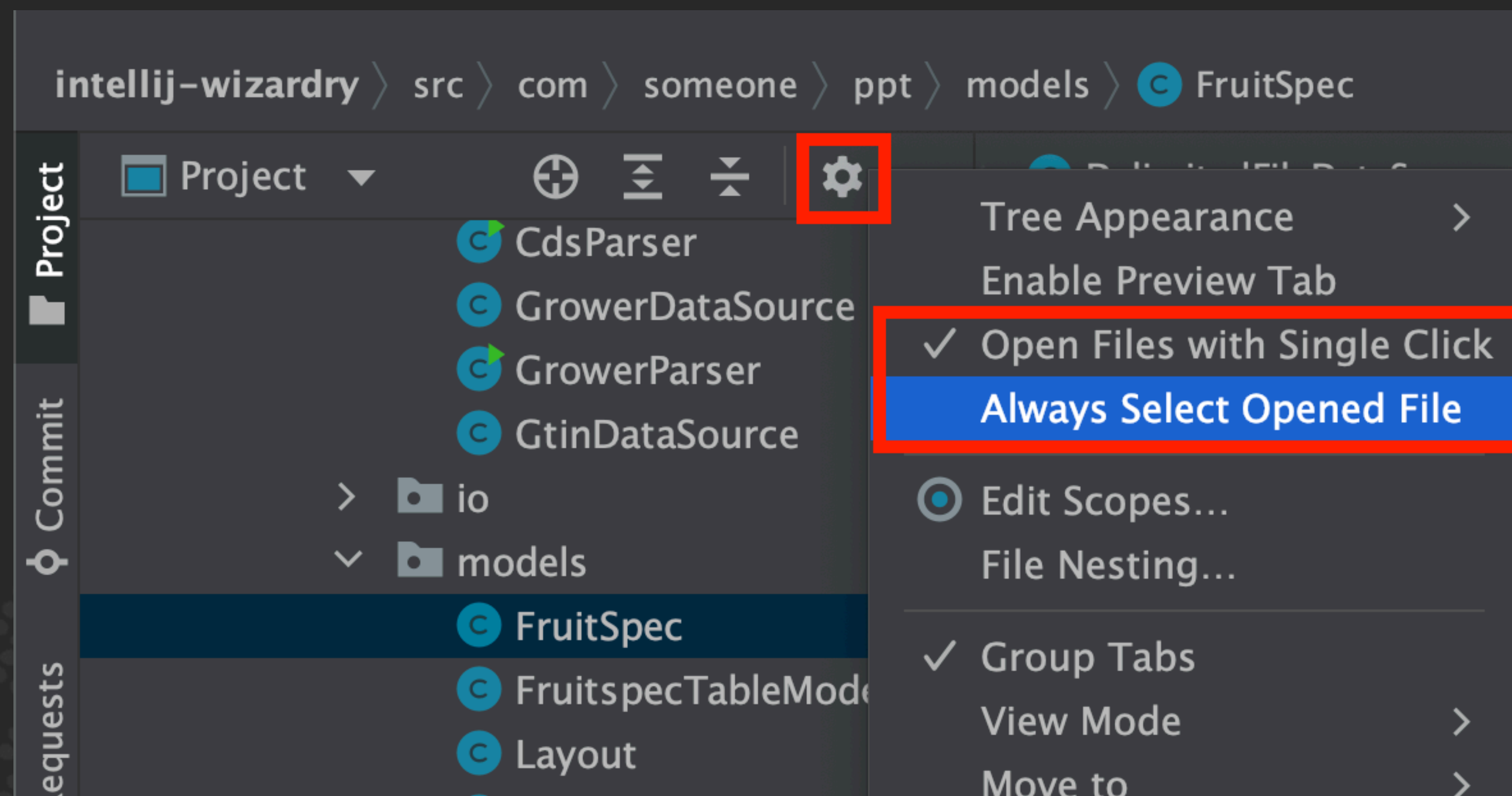
Editor Settings

◉ **Editor →**

– General → Smart Keys: Select Use "CamelHumps" words

Autoscroll to/from source

◉ Seems to be on by default now

- • Lots of times saw programmers editing the wrong file

- – ✓ Open Files with Single Click

- – ✓ Always Select Opened File

JavaSpecialists

IntelliJ IDEA Philosophy

IntelliJ IDEA Philosophy

◉ **IntelliJ designed to be used mostly without mouse**

◉ **Hotkeys for almost everything**

– Print out Help → Keyboard Shortcuts PDF

– Memorize one new shortcut per day ≈ 6 months

Help → My Productivity

◉ **Track progress in how productive you have become**

Productivity Guide

IntelliJ IDEA uptime: 1 day, 4 hr, idle time: 2 sec, 804 ms
Code completion has saved you from typing at least 802,0K characters since 2014/01/09 (~611 per working day)
Quick fixes have saved you from 18,906 possible bugs since 2014/01/12 (~15 per working day)

Feature	Group	Used ▼	Last Used
Syntax aware selection	Code Editing	167,508 times	16 hours ago
Variable name completion	Code Completion	144,269 times	16 hours ago
Basic code completion	Code Completion	88,357 times	16 hours ago

To expand selection, press ⌘W. Each time you press ⌘W, the selection expands to other areas of code.

For example, the selection expands from a method name to the expression calling this method, then to the whole statement, then to the containing block, and so on.

?   Close

Searching

- **"Search Everywhere"**

  - Windows/Linux: Double Shift

  - Mac OS X: ⇧⇧

- **"Search Everywhere and Include non-project items"**

  - Windows/Linux: Quadruple Shift

  - Mac OS X: ⇧⇧⇧⇧

- **Kept on hitting this by mistake when pressing ↑**

JavaSpecialists

When IntelliJ IDEA Hangs Up

Livelock from Corrupt Local Files

Recovering from Broken Indexes

◉ **IntelliJ gets into a bad state if indexes are corrupted**

– File ➞ Repair IDE...

• Try this first, stepping through until everything works

– File ➞ Invalidate Caches...

• Usually takes a while to reindex

– But solves most of the problems

– Try to not lose local history
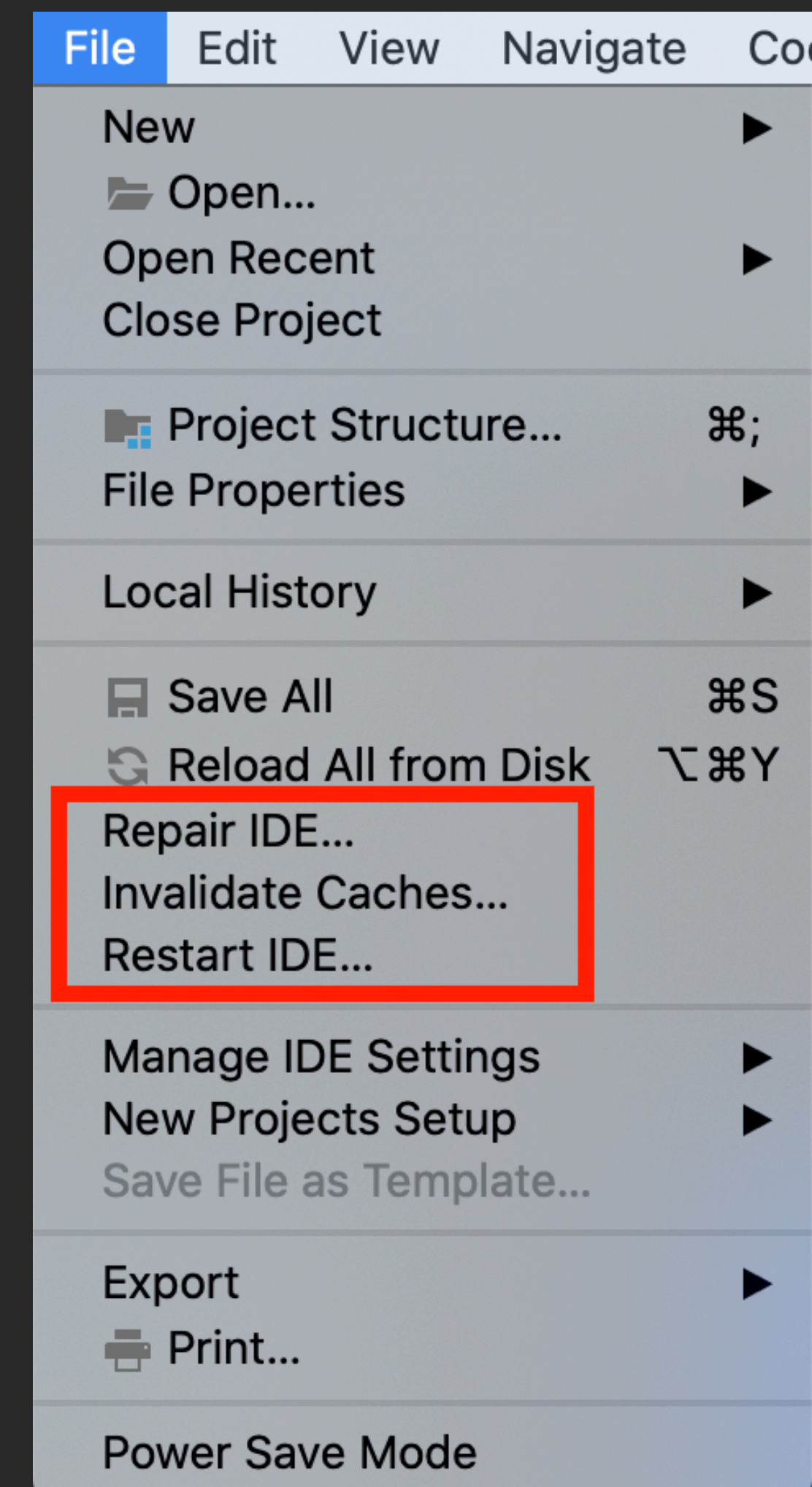
Opening old projects

◉ **Modern versions of IntelliJ IDEA might go into infinite loops**

  – Need to kill with -9

◉ **Create new project with existing sources and overwrite old project files**

JavaSpecialists

Essential Shortcuts

Superkey for fixing almost anything

◉ **"Show intention actions and quick-fixes"**

– Windows/Linux: Alt + Enter

– Mac OS X: ⌥↩

Generate new code

◉ **"Generate Code"**

  – Windows/Linux: Alt + Ins

  – Mac OS X (Official): ⌘N

  – Mac OS X (Heinz): ^↩ or ^N

◉ **Quick demo creating playground.Person**

  – Add final field, show difference between ^↩ and ⌥↩

Before we continue ...

◉ **Get our Data Structures in Java Course here**

– tinyurl.com/devoxx-23

– Coupon expires today at 11:00 Brussels Time

  • But you have life-time access once you have redeemed it

◉ **For those watching the recording**

– Sign up to The Java Specialists' Newsletter

  • www.javaspecialists.eu

  • Reply to the welcome mail that you would like this course

Live Templates

◉ **We can generate code quickly with live templates**

– psvm or main: Main method

– sout, soutv, soutm, soutp: Output

– iter, fori, itco, itar: Iteration

tinyurl.com/devoxx-23

Navigation

◉ **"Go to declaration"**

- – Windows/Linux: Ctrl + B or Ctrl + Click

- – Mac OS X: ⌘B or ⌘Click

◉ **"Navigate back / forward"**

- – Windows/Linux: Ctrl + Alt + Left / Right

- – Mac OS X: ⌘⌥← / ⌘⌥→

tinyurl.com/devoxx-23

Should you throw away your mouse?

◉ **Everything can be done with keyboard in IDEA**

– It is useful to learn to touch type

– I usually have left hand on keyboard and right on mouse

• Easy enough to find the correct keys - index fingers on F & J

◉ **For navigating, I find the mouse faster**

– Hold down Ctrl or ⌘ and everything becomes a hyperlink

– Scrolling with mouse or touchpad smoother

tinyurl.com/devoxx-23

Bookmarks

◉ **Quick navigation between locations in project**

◉ **Set with**

– Windows/Linux: Ctrl + Shift + #[0-9]

– Mac OS X: ⌃⇧0 ... ⌃⇧9

◉ **Navigate with**

– Windows/Linux: Ctrl + #[0-9]

– Mac OS X: ⌃0 ... ⌃9

Syntax Aware Selection

◉ **"Extend Selection"**

– Windows/Linux: Ctrl + W

– Mac OS X (Official): ⌥↑

Feature	Group	Used ▼
Syntax aware selection	Code Editing	167,508 times
Variable name completion	Code Completion	144,269 times
Basic code completion	Code Completion	88,357 times

– Mac OS X (Heinz): ⌘W

  • Closes windows in other Mac OS X programs

  • But my left thumb and middle finger and pinkie do this nicely

  • By FAR my most used shortcut, 167k times since 2014

◉ **"Shrink Selection"**

– Windows/Linux: Ctrl + Shift + W

– Mac OS X (Official): ⌥↓

– Mac OS X (Heinz): ⌘⇧W
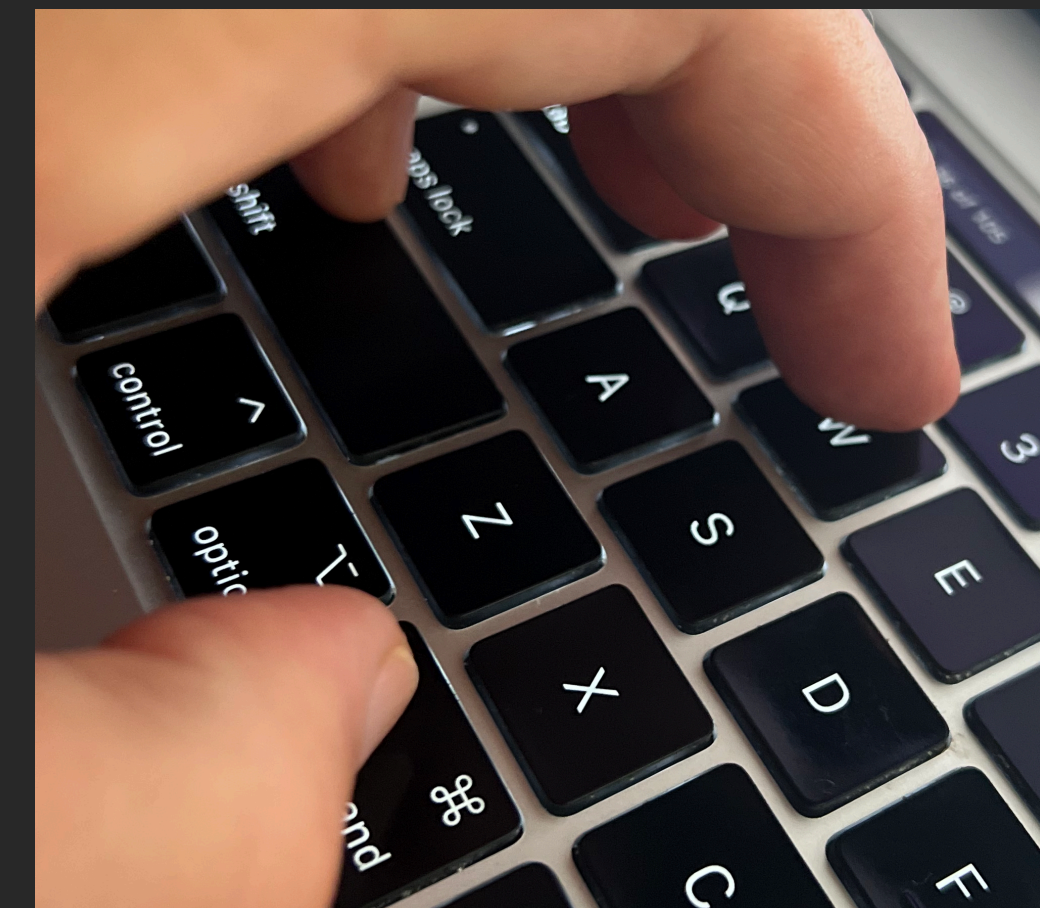
Move code up / down

⊙ **"Move Code Up"**

– Windows/Linux: Ctrl + Shift + Up

– Mac OS X: ⌘⇧↑

⊙ **"Move Code Down"**

– Windows/Linux: Ctrl + Shift + Down

– Mac OS X: ⌘⇧↓

⊙ **Note: If nothing is selected, we consider current line to be selected**

JavaSpecialists

Turbo-boosted Productivity

Surround with Live Template

◉ **"Surround with Live Template"**

– Windows/Linux: Ctrl + Alt + J

– Mac OS X: ⌘ ⌥ J

◉ **Again context aware, for example with Java**

C. Surround with Callable
RL. Surround with ReadWriteLock.readLock
WL. Surround with ReadWriteLock.writeLock
I. Iterate Iterable or array

◉ **Demo: changing main(String[]) to main(String...)**

Define your own Live Templates

◉ **Preferences -> Editor -> Live Templates**

◉ **e.g. Wrap code in System.nanoTime()**

– Fantastic for demos, use JMH for serious benchmarks
Abbreviation: nanoTime
Description: System.nanoTime()
Template text:

```
long $TIME$ = System.nanoTime();
try {
  $SELECTION$
} finally {
  $TIME$ = System.nanoTime() - $TIME$;
  System.out.printf("$TIME$ = %dms%n",
    ($TIME$/1_000_000));
}
```

Column Select Editing

◉ **With the mouse**

– Windows/Linux: Alt + Drag Mouse

– Mac OS X: ⌥+drag mouse

◉ **With keyboard toggle column selection mode**

– Windows/Linux: Alt + Shift + Insert

– Mac OS X (Official): ⌘⇧8

– Mac OS X (Heinz): ⌘^⇧C

– Make sure to turn column selection mode off afterwards

Demo

◉ **Inside the FruitSpec, create an enum that holds all the fields**

```java
public enum Field {
    BARCODE("BarCode"),
    RUNNUMBER("RunNumber"),
    PUC("PUC"),
    /* etc. ... */
    GTIN("GTIN");
    private final String name;

    Field(String name) {
        this.name = name;
    }
}
```

JavaSpecialists

Tips for Speakers

Assign shortcut for Presentation Mode

◉ **I used Command + Option P, same as for Keynote**

◉ **Change fonts or zoom level for presentation mode in**

– Appearance & Behavior -> Appearance -> Presentation Mode

Define different Color Schemes for Resolutions

◉ **E.g. Darcula Talks 1080p, Darcula Talks 720p, etc.**

– Fonts should be easily readable when watching on a smart phone

◉ **Might need to use white background with bad projector**

◉ **Switch quickly with Control + `**

JavaSpecialists

Code Completion

Basic Code Completion

◉ **Code completion traditionally uses Ctrl+Space**

  – Windows/Linux: Ctrl + Space

  – Mac OS X: ^Space
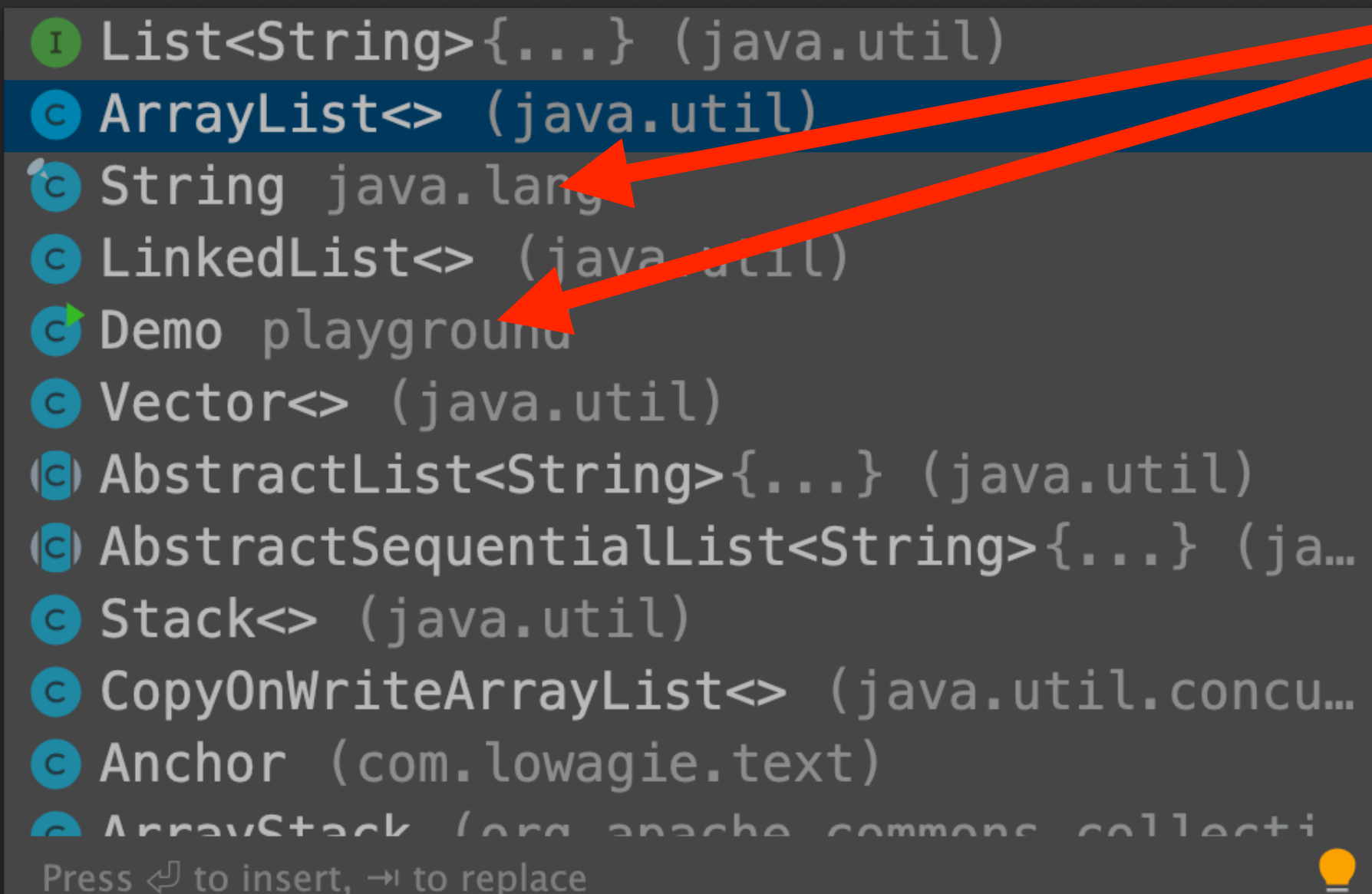
```java
import java.util.List;

public class Demo {
    public static void main(String[] args) {
        List<String> names = new
    }
}
```

| I List<String>{...} (java.util) |
| C ArrayList<> (java.util) |
| C String java.lang |
| C LinkedList<> (java.util) |
| C Demo playground |
| C Vector<> (java.util) |
| C AbstractList<String>{...} (java.util) |
| C AbstractSequentialList<String>{...} (ja… |
| C Stack<> (java.util) |
| C CopyOnWriteArrayList<> (java.util.concu… |
| C Anchor (com.lowagie.text) |
| C ArrayStack (org.apache.commons.collecti… |

Press ↵ to insert, →I to replace

?

Smart Code Completion

◎ **This gives much better result - I always use this**

– Windows/Linux: Ctrl + Shift + Space

– Mac OS X: ⌃⇧Space

```java
import java.util.List;

public class Demo {
    public static void main(String[] args) {
        List<String> names = new |
    }
}
```

> Ⓘ List<String>{...} (java.util)
> Ⓒ Anchor (com.lowagie.text)
> Ⓒ ArrayList<> (java.util)
> Ⓒ LinkedList<> (java.util)
> Ⓒ AbstractList<String>{...} (java.util)
> Ⓒ Vector<> (java.util)
> Ⓒ AbstractSequentialList<String>{...} (ja…
> Ⓒ Stack<> (java.util)
> Ⓒ Phrase (com.lowagie.text)
> Ⓒ CopyOnWriteArrayList<> (java.util.concu…
> Ⓒ ArrayStack (org.apache.commons.collecti…
> Ⓒ Chapter (com.lowagie.text)
> Press ⏎ to insert, ⇥ to replace

JavaSpecialists

Refactoring

Alter Class / Field / Method

◉ **"Move"**

– Windows/Linux: F6

– Mac OS X: F6

◉ **"Rename ..."**

– Windows/Linux: Shift + F6

– Mac OS X: ⇧F6

◉ **"Change Signature" - applies to methods**

– Windows/Linux: Ctrl + F6

– Mac OS X: ⌘F6

Extract Method

⊙ **Select a block of code and "Extract Method"**

– Windows/Linux: Ctrl + Alt + M

– Mac OS X: ⌘⌥M

⊙ **Some restrictions**

– Cannot have more than one return value

– Block must represent a set of statements or expressions

⊙ **Additional benefits**

– Extracting a method can discover other, similar, code

Demo

◉ **Extract snippets from generatePcdRemarks() method in PcdGenerator**

```java
String remark = resultSet.getString("Remark1");
if (!"".equals(remark)) {
    remarks.put(remark, remark);
}


remark = resultSet.getString("Remark2");
if (!"".equals(remark)) {
    remarks.put(remark, remark);
}


remark = resultSet.getString("Remark3");
if (!"".equals(remark)) {
    remarks.put(remark, remark);
}
```

Inline Code

- **Applies to methods, fields, local variables**

- **"Inline"**
  – Windows/Linux: Ctrl + Alt + N

  – Mac OS X: ⌘⌥N

- **Conveniently close to "Extract Method" shortcut**

JavaSpecialists

Analyzer

Analyzer

◉ **IntelliJ shines with its code analyzer and refactoring**

◉ **Code → Inspect Code ...**

– Whole project

• Or: Right-click in project → Analyze → Inspect Code ...

– Inspection profile: Default IDE

◉ **It checks for the most glaring code inconsistencies**

– Let's try it out together

Java Language Migration Aids

◉ **Demos**

- Use enhanced switch in
  - CdsGenerator#generateCfg()
  - PcdGenerator#generatePcdTemplate()
- Use pattern variable in ElegantTable#initGui()
- Use try-with-resource in LineSocket#receiveFile()

JavaSpecialists

Conclusion

The Java Specialists' Newsletter

◉ **Make sure to subscribe**

  – www.javaspecialists.eu/archive/subscribe/

◉ **Readers in 150+ countries**

◉ **Over 22 years of newsletters on advanced Java**

  – All previous newsletters available on www.javaspecialists.eu

  – Courses, additional training, etc.

Remember please

◉ **Get our Data Structures in Java Course here**

– tinyurl.com/devoxx-23

– Coupon expires today at 11:00 Brussels Time

• But you have life-time access once you have redeemed it

tinyurl.com/devoxx-23

Conclusion IntelliJ Super Productivity

◉ **Many more keystrokes and features to learn**

◉ **One new one per day**

◉ **Happy coding!**