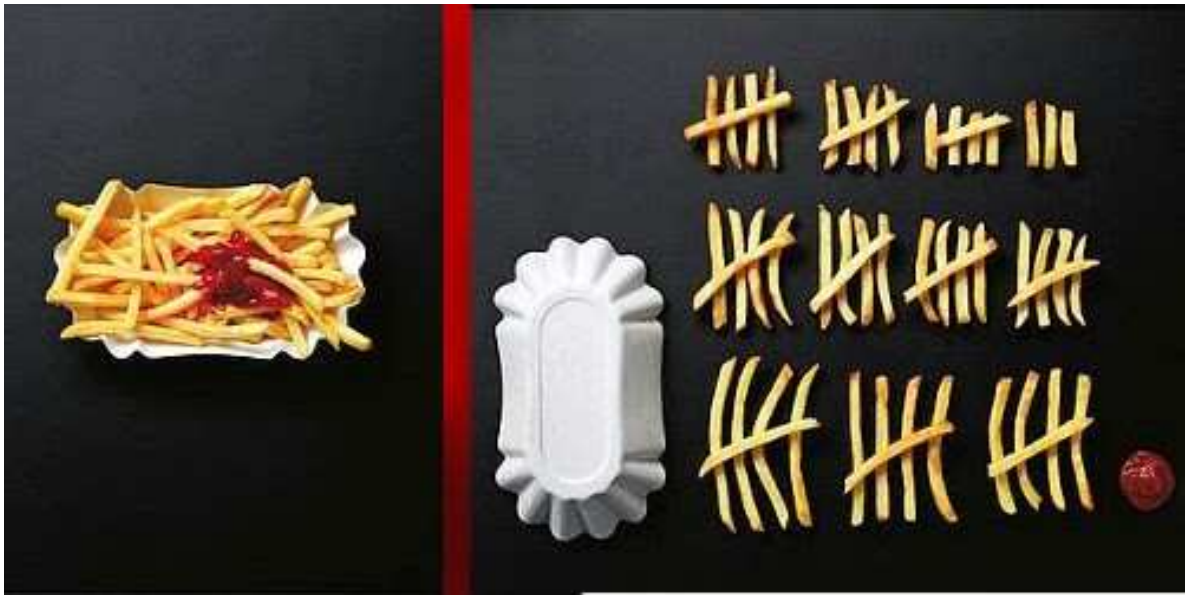


Refactoring with IntelliJ

A short guide

Picture by Ursus Wehrli

Motivation

- Save transformation
- Work is easier
- Use powerfull tooling
- Decide about outcome
- Learn the IDE

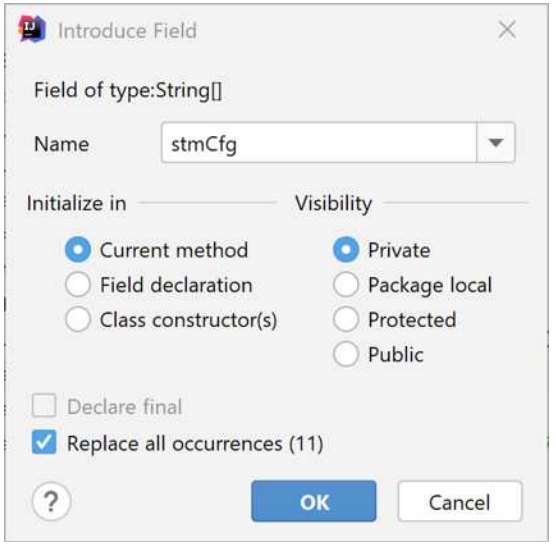
Extract objects

- Done with Ctrl – Alt – type of extraction
  - M extract Method
  - V extract Variable
  - C extract Constant
  - F extract Field
  - P extract Parameter

Field extraction

Extract field with Ctrl-Alt-F

```
StringWriter sw = new StringWriter();
sw.append(Globals.StmCfg[0]);
sw.append("\n");
for (String key : gt.keySet()) {
    var t : Quintet<String, Date, String, Integer, Integer> = gt.get(key);
    String sn = t.getValue0();
    int nOfShs = t.getValue4();
    DecimalFormat dc = new DecimalFormat(Globals.StmCfg[1]);
    SimpleDateFormat df = new SimpleDateFormat(Globals.StmCfg[2]);
    var cl : HttpClient = HttpClient.newHttpClient();
    var r : HttpRequest = HttpRequest
```

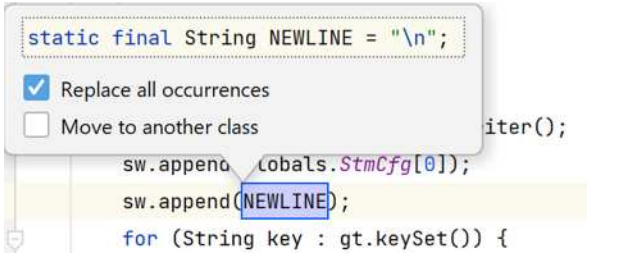



```
String[] stmCfg = Globals.StmCfg;
sw.append(stmCfg[0]);
sw.append("\n");
for (String key : gt.keySet()) {
    var t : Quintet<String, Date, String, Integer, Integer> = gt.get(key);
    String sn = t.getValue0();
    int nOfShs = t.getValue4();
    DecimalFormat dc = new DecimalFormat(stmCfg[1]);
    SimpleDateFormat df = new SimpleDateFormat(stmCfg[2]);
    var cl : HttpClient = HttpClient.newHttpClient();
    var r : HttpRequest = HttpRequest
            .newBuilder(URI.create(stmCfg[3] + URLEncoder.encod
```

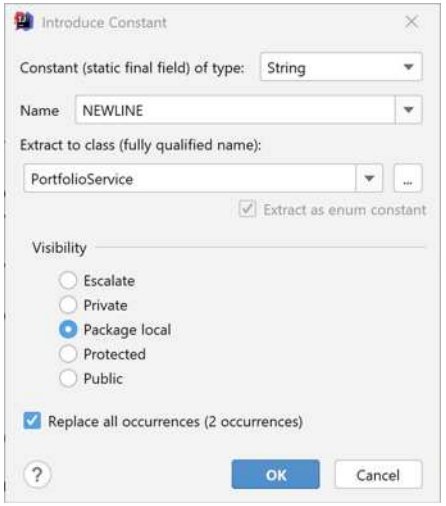
Extract constant

Extract field with Ctrl-Alt-C

```
StringWriter sw = new StringWriter();
sw.append(Globals.StmCfg[0]);
sw.append("\n");
for (String key : gt.keySet()) {
    var t : Quintet<String, Date, String, Integer, Integer> = gt.get(key);
    String sn = t.getValue0();
```

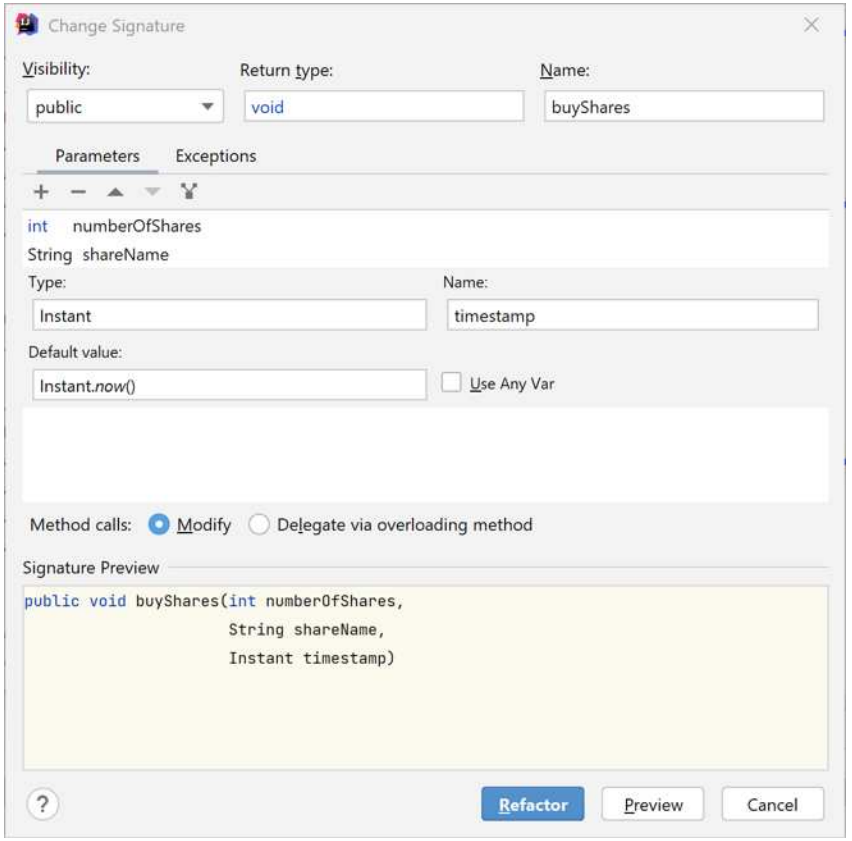


Press Ctrl-Alt-C again



Change signature

Change of method signature Ctrl-F6



- Change name of method
- add/remove parameters
- give new parameters default
- rename parameters
- change order of parameters
- use preview

Extract superclass

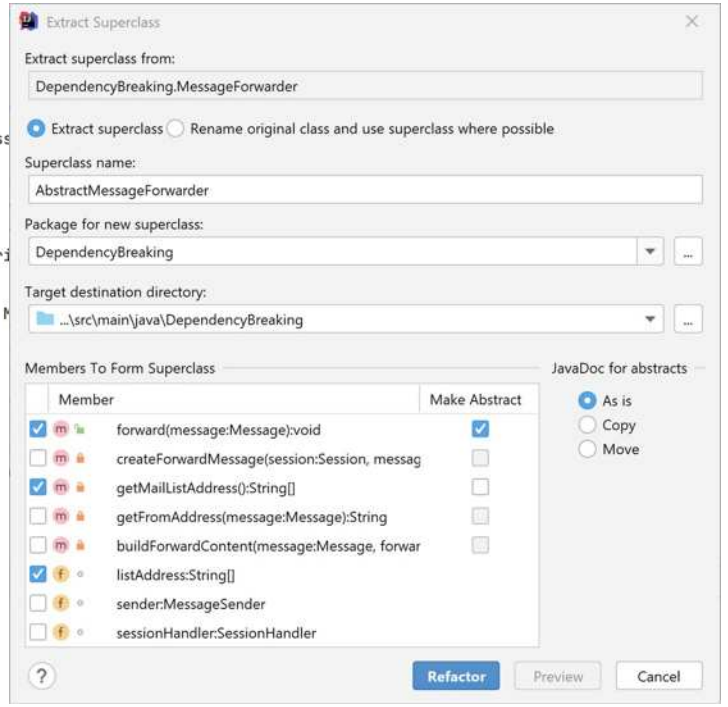
```java
public class MessageForwarder {
    String[] listAddress = new String[]{};
    MessageSender sender;
    SessionHandler sessionHandler;

    public MessageForwarder(SessionHandler sessionHandler, MessageSender sender) {
        this.sessionHandler = sessionHandler;
        this.sender = sender;
    }

    public void forward(Message message) {
        Message messageToForward =
                createForwardMessage(sessionHandler.getSess
        sender.send(messageToForward);
    }

    private String[] getMailListAddress() { return new Stri

    private Message createForwardMessage(Session session, M
        Message forward = new MimeMessage(session);
        forward.setFrom(getFromAddress(message));
```




```java
public class MessageForwarder extends AbstractMessageForwarder {
    MessageSender sender;
    SessionHandler sessionHandler;

    public MessageForwarder(SessionHandler sessionHandler, MessageSender sender) {
        this.sessionHandler = sessionHandler;
        this.sender = sender;
    }

    @Override
    public void forward(Message message) {
        Message messageToForward =
                createForwardMessage(sessionHandler.getSession(), message);
        sender.send(messageToForward);
    }

    private Message createForwardMessage(Session session, Message message) {
        Message forward = new MimeMessage(session);
```

```java
public abstract class AbstractMessageForwarder {
    String[] listAddress = new String[]{};

    public abstract void forward(Message message);

    protected String[] getMailListAddress() { return new String[0]; }
}
```

Conclusions

- use IDE when ever possible
- manual work much more error-prone
- easier then in Eclipse
- it's fun with the right tools
- try *analyze* possibilities 
- look for further refactoring possibilities

References

- Code refactoring on jetbrains:
  https://www.jetbrains.com/help/idea/refactoring-source-code.html
- Keyboard shortcuts:
  https://www.jetbrains.com/help/idea/reference-keymap-win-default.html
- In IntelliJ: Help → Keymap reference

Thanks for your attention and have fun with IntelliJ

code refactoring → art style

CodeReorderer.java

```java
package DependencyBreaking;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.*;
import java.util.stream.Collectors;
import java.util.stream.Stream;

public class CodeReorderer {

    public static void main(String[] args) throws IOException {
        Path fileName = Path.of( first: "src/main/java/DependencyBreaking/CodeReorderer.java");
        String actual = Files.readString(fileName);

        CodeReorderer codeReorderer = new CodeReorderer();
        Map<String, Long> characterCountMap = codeReorderer.getCharacterCountMap(actual);

        Path orderedFile = Path.of( first: "ordered.txt");
        String result = codeReorderer.extractKeyMap(characterCountMap);
        Files.writeString(orderedFile, result);
    }

    public Map<String, Long> getCharacterCountMap(final String input) {
        return Stream.of(input.split( regex: ""))
                .filter(this::isPrintable)
                .collect(Collectors.groupingBy(s -> s, TreeMap::new, Collectors.counting()));
    }

    public String extractKeyMap(Map<String, Long> characterMap) {
        return characterMap.entrySet() Set<Map<K, V>.Entry<String, Long>>
                .stream() Stream<Map<K, V>.Entry<String, Long>>
                .map(es -> String.join( delimiter: "", Collections.nCopies(es.getValue().intValue(),
                .collect(Collectors.joining( delimiter: "\n"));
    }

    private boolean isPrintable(final String character) {
        return character.charAt(0) > ' ';
    }
}
```

ordered.txt

```
"""""""""
''
((((((((((((((((((((((((((
))))))))))))))))))))))))))
*
,,,,,,,,
--
......................................
////
0
::::
;;;;;;;;;;;;;;;;
<<<
======
>>>>>>
A
BBB
CCCCCCCCCCCCCCC
DD
EE
FFFFF
II
KKK
LLL
MMMMMMMMMMMM
NN
OO
PPPPPPP
RRRRRRR
SSSSSSSSSSSSSSS
T
VV
[
\
]
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
bbbbbbb
ccccccccccccccccccccccccccccccccccccccccccccc
dddddddddddddddddddddddd
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
ffffffffff
ggggggggggggggggggggggggggg
hhhhhhhhhhhhhhhh
iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
jjjjjjjjjj
kkk
```