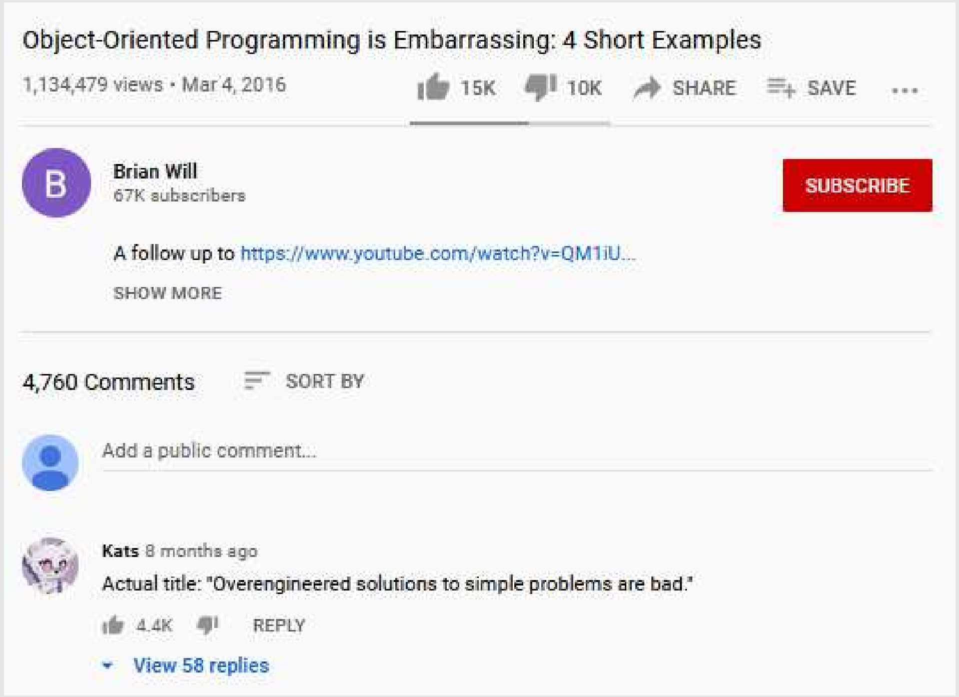
Alcor Training Running Taking a look at „Object–Oriented Programming is Embarassing“

Andy Nyffenegger

andy.nyffenegger@css.ch

- A catchy title
- The problem
- Uncle Bob‘s Solution
- Brian Will‘s Solution
- But ...

A catchy title


Object-Oriented Programming is Embarrassing: 4 Short Examples

1,134,479 views • Mar 4, 2016

Brian Will
67K subscribers

A follow up to https://www.youtube.com/watch?v=QM1iU...

4,760 Comments

Kats 8 months ago
Actual title: "Overengineered solutions to simple problems are bad."

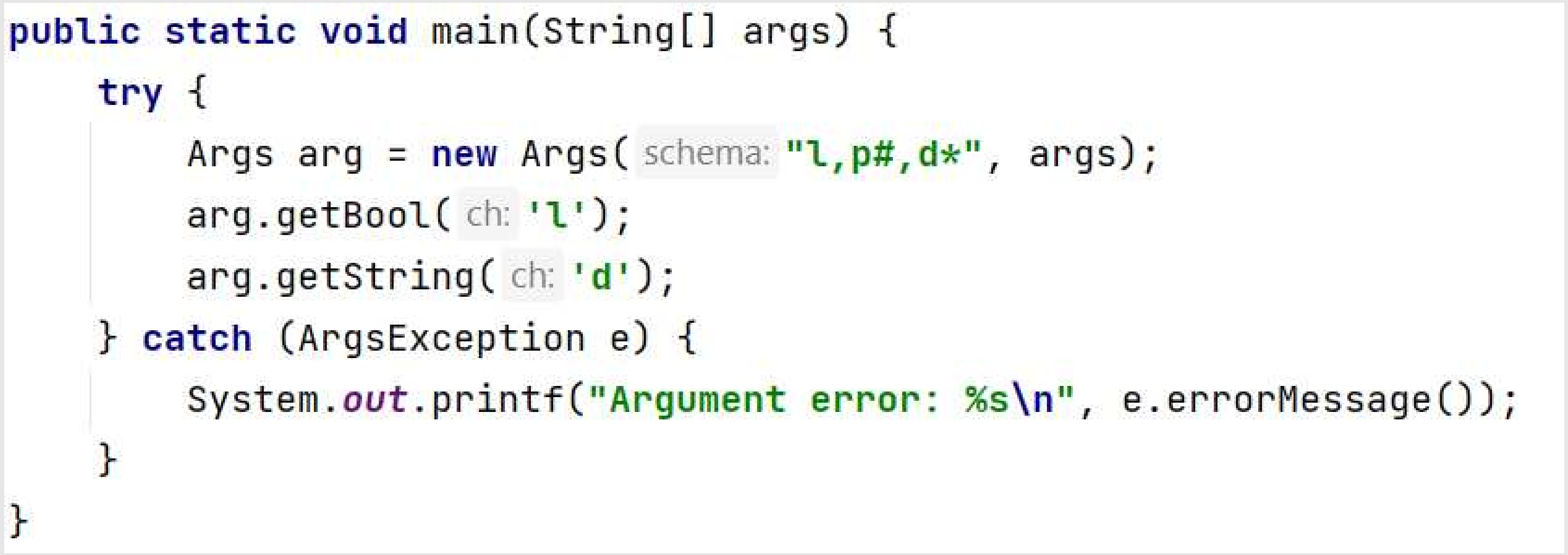
- Youtube Video by Brian Will:
Object-Oriented Programming is Embarrassing: 4 Short Examples
Published: 4.3.2016

„I have nothing nice to say about Object-Oriented programming. I dont like it. I think it‘s really been a disaster for the industry.“ – Brian Will

- 4 Examples: 2 Java, 2 Ruby
- Second Java Example:
Uncle Bob‘s Argument Parser from his book „Clean Code“ chapter 14

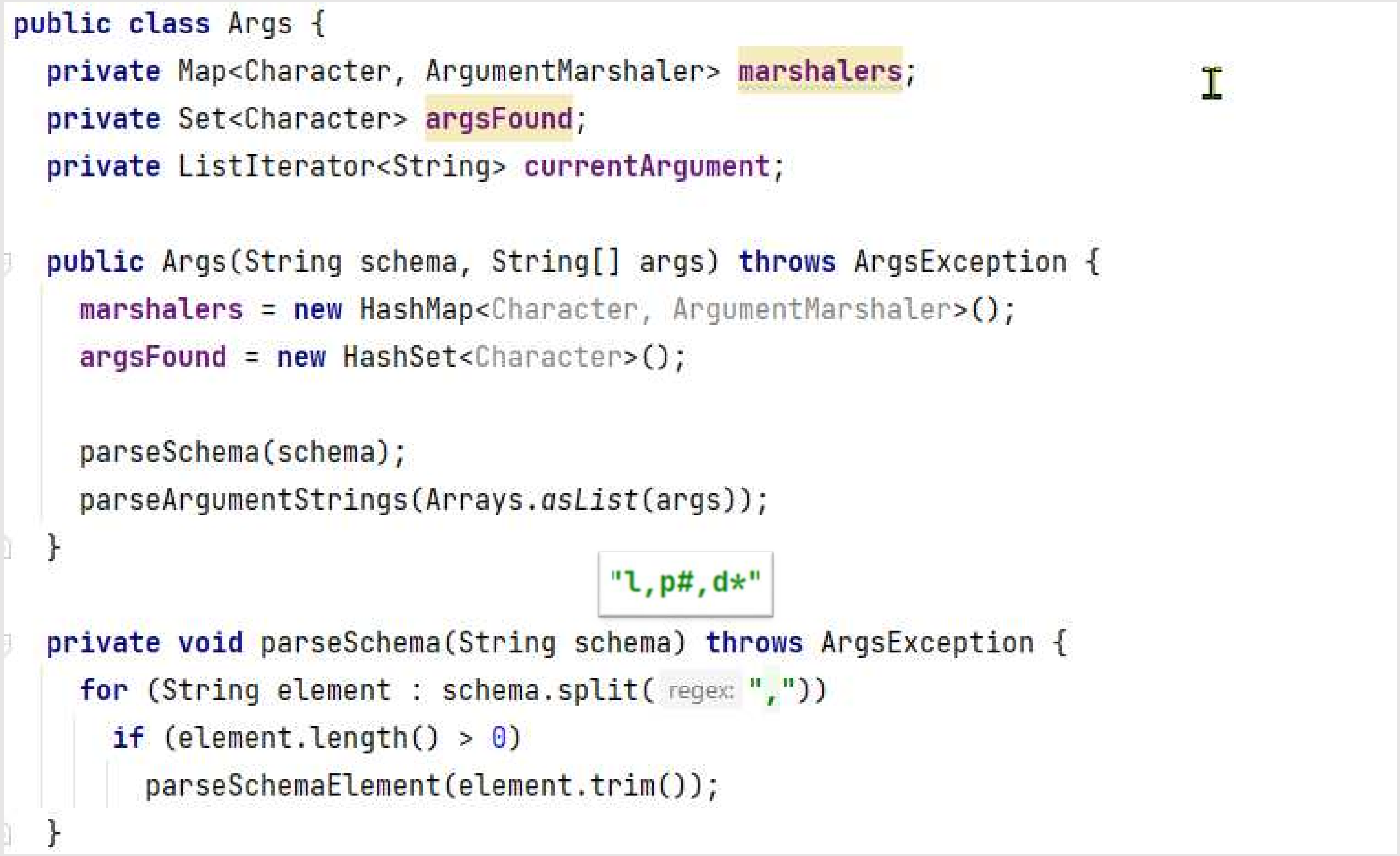
The Problem: Command Line Parser

- Parse command-line parameters to make them easy accessible
- Schema defines the parameters and their types : „l,p#,d*“
  - l -> boolean type (because there is no type Info)
  - p -> integer type (because of #)
  - d -> string type (because of *)
- Command line: myProgram -p 123 -d smells -l
- Args Class:



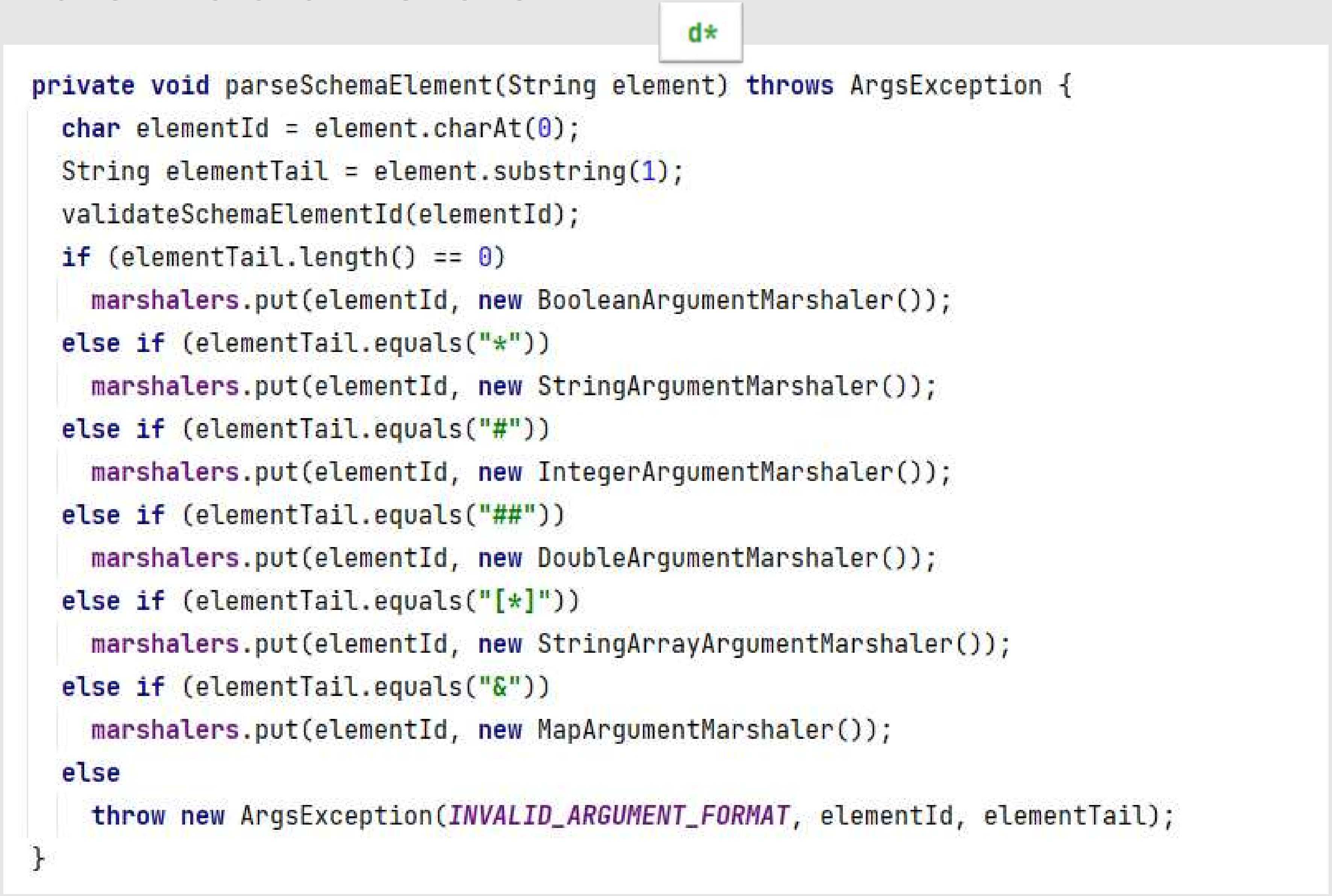
```
public static void main(String[] args) {
    try {
        Args arg = new Args( schema: "l,p#,d*", args);
        arg.getBool( ch: 'l');
        arg.getString( ch: 'd');
    } catch (ArgsException e) {
        System.out.printf("Argument error: %s\n", e.errorMessage());
    }
}
```

Uncle Bob‘s Version I



```
public class Args {
  private Map<Character, ArgumentMarshaler> marshalers;
  private Set<Character> argsFound;
  private ListIterator<String> currentArgument;

  public Args(String schema, String[] args) throws ArgsException {
    marshalers = new HashMap<Character, ArgumentMarshaler>();
    argsFound = new HashSet<Character>();

    parseSchema(schema);
    parseArgumentStrings(Arrays.asList(args));
  }
                                      "l,p#,d*"
  private void parseSchema(String schema) throws ArgsException {
    for (String element : schema.split( regex: ","))
      if (element.length() > 0)
        parseSchemaElement(element.trim());
  }
```

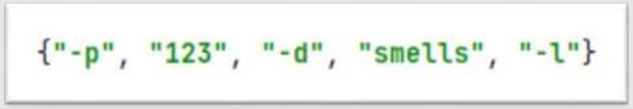
Uncle Bob‘s Version II



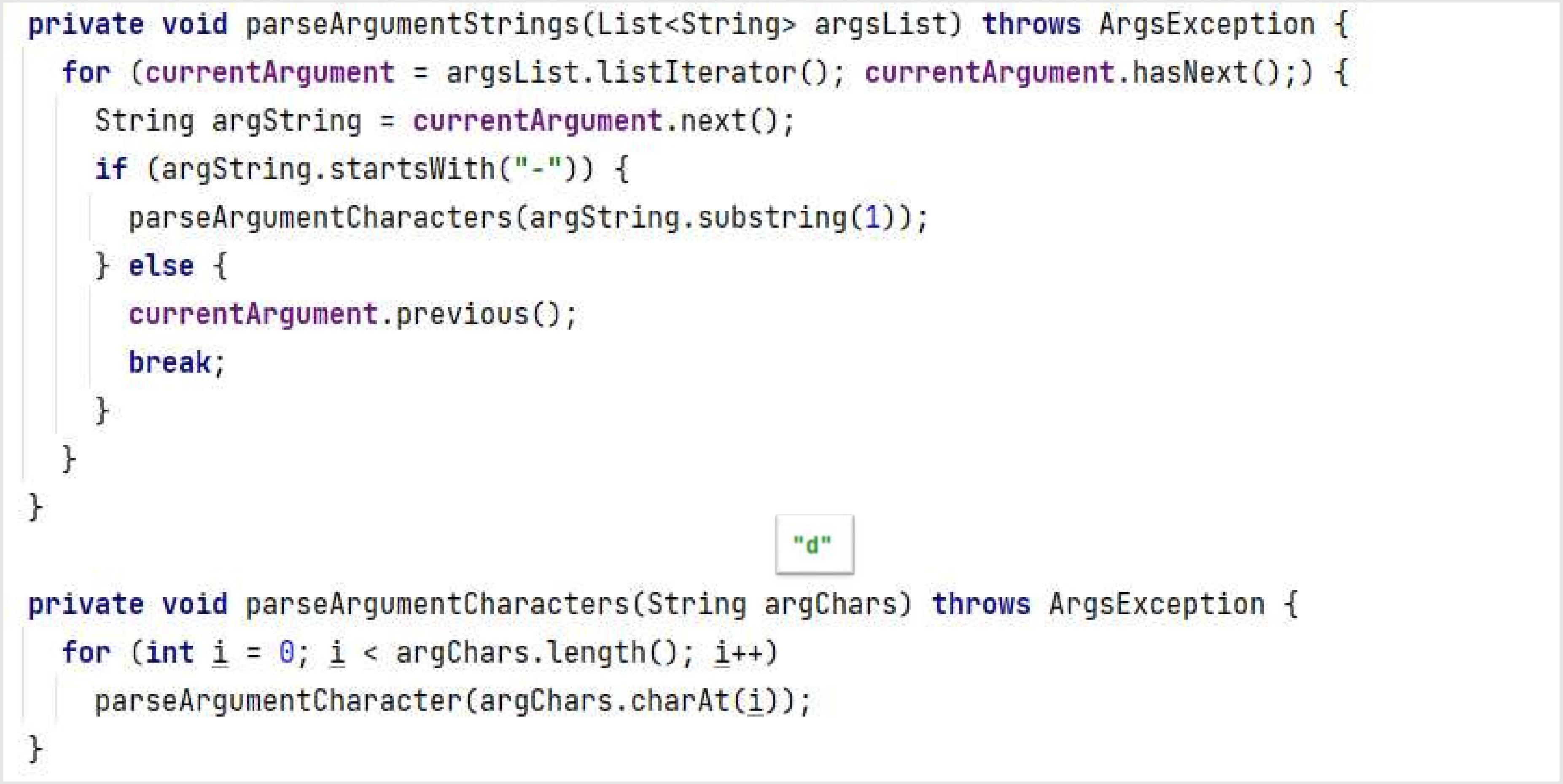
d*

```java
private void parseSchemaElement(String element) throws ArgsException {
  char elementId = element.charAt(0);
  String elementTail = element.substring(1);
  validateSchemaElementId(elementId);
  if (elementTail.length() == 0)
    marshalers.put(elementId, new BooleanArgumentMarshaler());
  else if (elementTail.equals("*"))
    marshalers.put(elementId, new StringArgumentMarshaler());
  else if (elementTail.equals("#"))
    marshalers.put(elementId, new IntegerArgumentMarshaler());
  else if (elementTail.equals("##"))
    marshalers.put(elementId, new DoubleArgumentMarshaler());
  else if (elementTail.equals("[*]"))
    marshalers.put(elementId, new StringArrayArgumentMarshaler());
  else if (elementTail.equals("&"))
    marshalers.put(elementId, new MapArgumentMarshaler());
  else
    throw new ArgsException(INVALID_ARGUMENT_FORMAT, elementId, elementTail);
}
```

Uncle Bob‘s Version IV



```
{"-p", "123", "-d", "smells", "-l"}
```

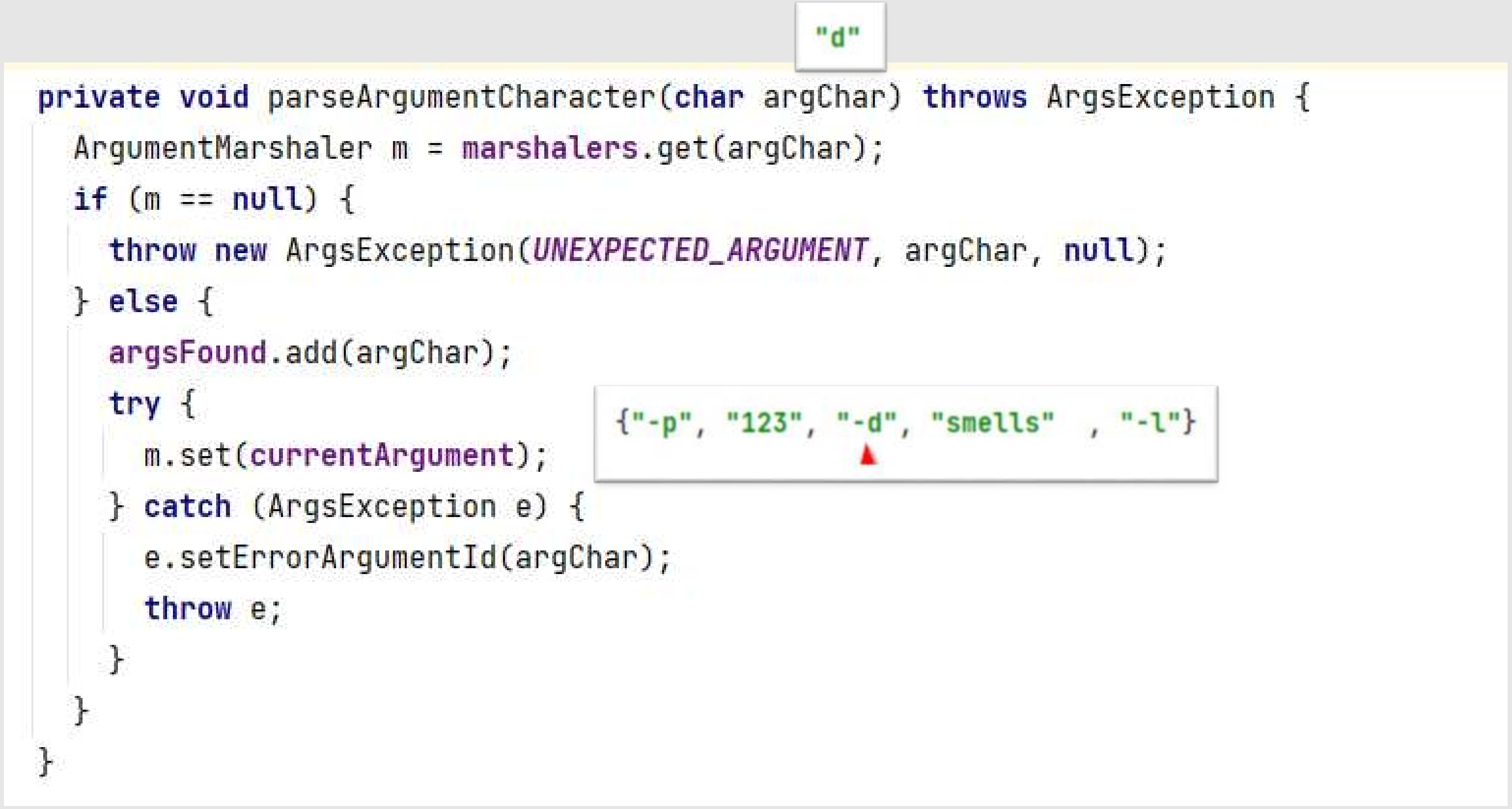


```
private void parseArgumentStrings(List<String> argsList) throws ArgsException {
  for (currentArgument = argsList.listIterator(); currentArgument.hasNext();) {
    String argString = currentArgument.next();
    if (argString.startsWith("-")) {
      parseArgumentCharacters(argString.substring(1));
    } else {
      currentArgument.previous();
      break;
    }
  }
}
```

```
"d"
```

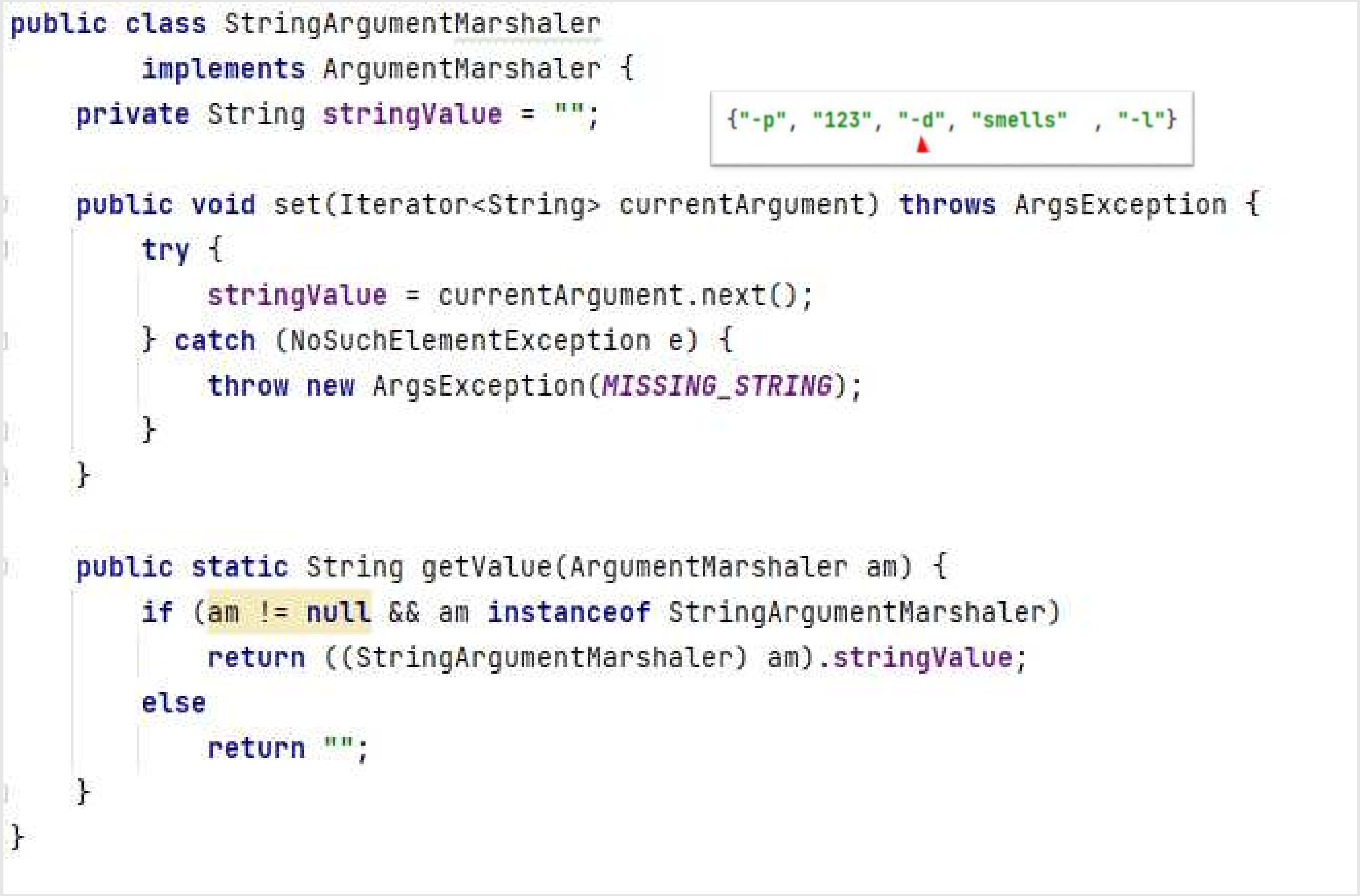
```
private void parseArgumentCharacters(String argChars) throws ArgsException {
  for (int i = 0; i < argChars.length(); i++)
    parseArgumentCharacter(argChars.charAt(i));
}
```

Uncle Bob‘s Version V



```
"d"
private void parseArgumentCharacter(char argChar) throws ArgsException {
  ArgumentMarshaler m = marshalers.get(argChar);
  if (m == null) {
    throw new ArgsException(UNEXPECTED_ARGUMENT, argChar, null);
  } else {
    argsFound.add(argChar);
    try {
      m.set(currentArgument);       {"-p", "123", "-d", "smells"  , "-l"}
    } catch (ArgsException e) {
      e.setErrorArgumentId(argChar);
      throw e;
    }
  }
}
```

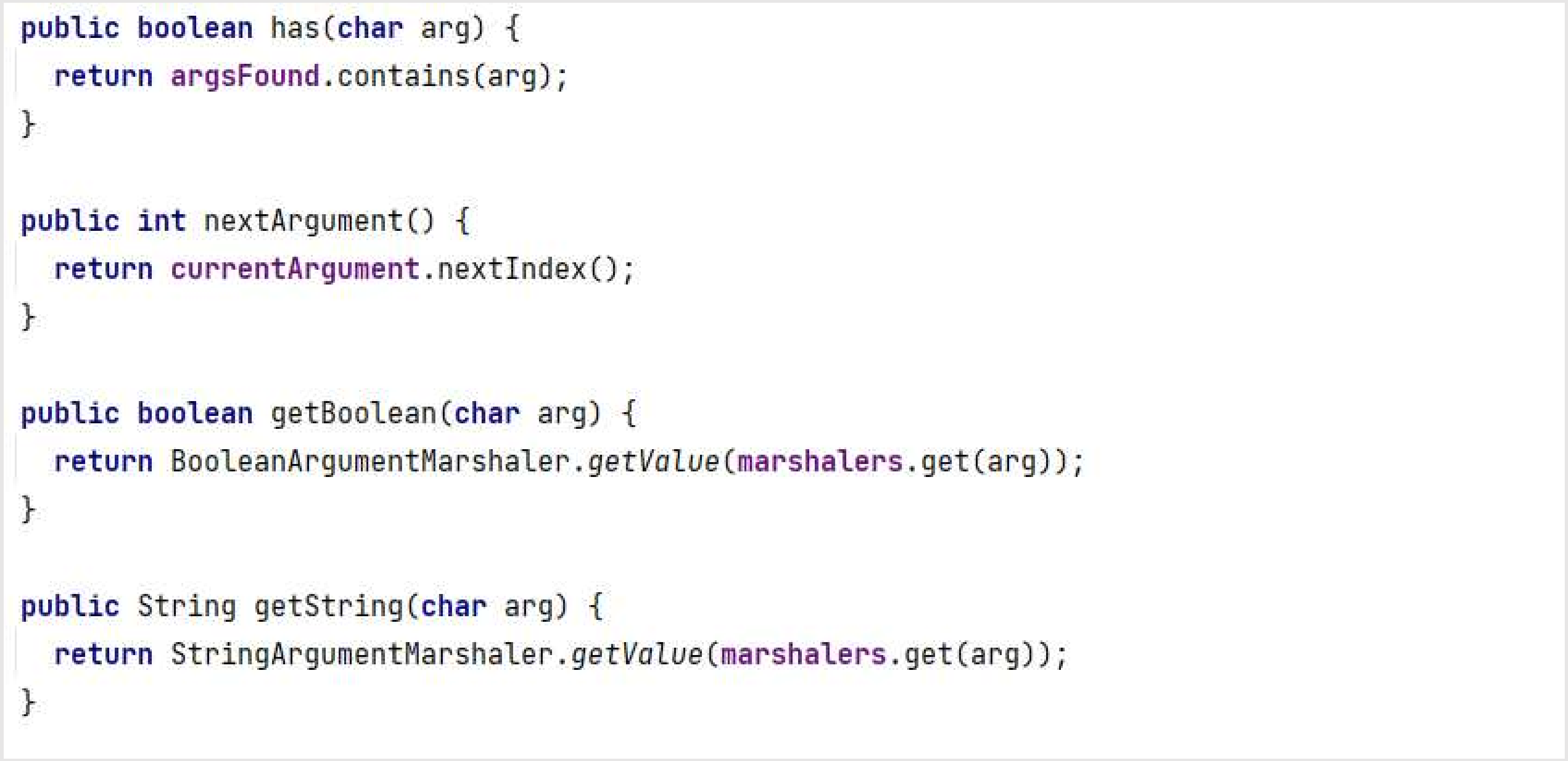
Uncle Bob‘s Version III



```
public class StringArgumentMarshaler
        implements ArgumentMarshaler {
    private String stringValue = "";

    public void set(Iterator<String> currentArgument) throws ArgsException {
        try {
            stringValue = currentArgument.next();
        } catch (NoSuchElementException e) {
            throw new ArgsException(MISSING_STRING);
        }
    }

    public static String getValue(ArgumentMarshaler am) {
        if (am != null && am instanceof StringArgumentMarshaler)
            return ((StringArgumentMarshaler) am).stringValue;
        else
            return "";
    }
}
```

```
{"-p", "123", "-d", "smells"  , "-l"}
```

Uncle Bob‘s Version VI

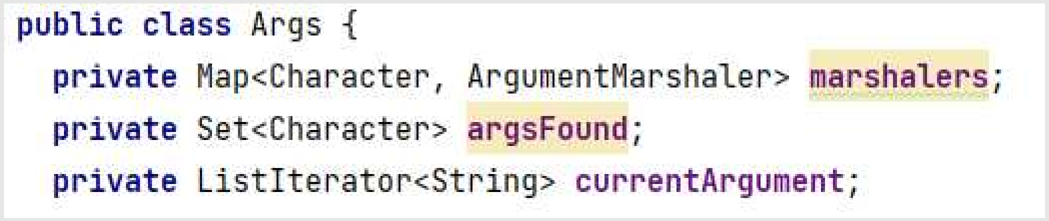


```java
public boolean has(char arg) {
  return argsFound.contains(arg);
}

public int nextArgument() {
  return currentArgument.nextIndex();
}

public boolean getBoolean(char arg) {
  return BooleanArgumentMarshaler.getValue(marshalers.get(arg));
}

public String getString(char arg) {
  return StringArgumentMarshaler.getValue(marshalers.get(arg));
}
```

…

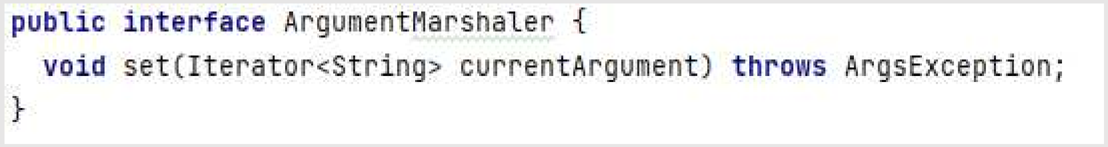
Uncle Bob‘s Version: Code Smells

temporary field:



```
public class Args {
  private Map<Character, ArgumentMarshaler> marshalers;
  private Set<Character> argsFound;
  private ListIterator<String> currentArgument;
```

encapsulation violation:



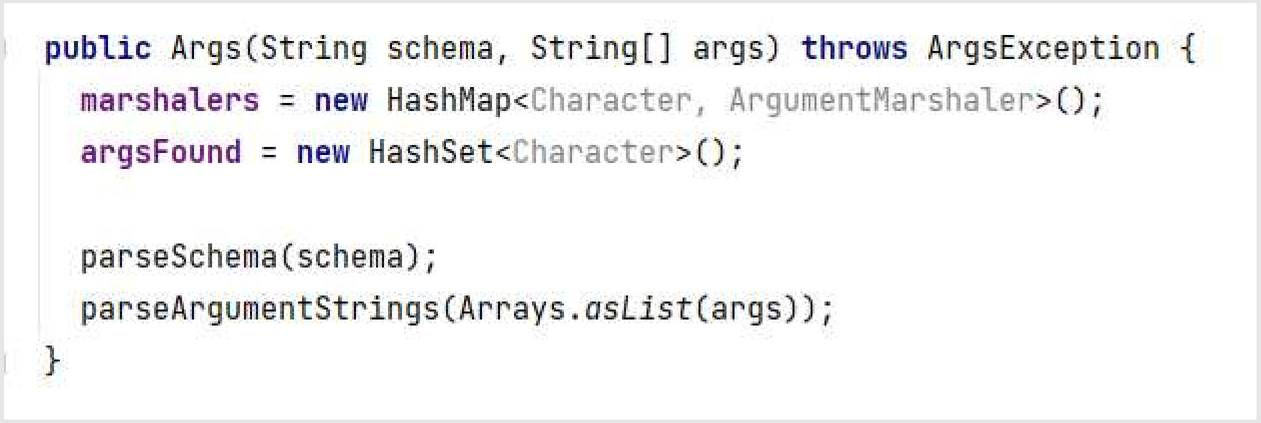
```
public interface ArgumentMarshaler {
  void set(Iterator<String> currentArgument) throws ArgsException;
}
```

large class: Args has about 100 loc

don‘t use else

Uncle Bob‘s Version: Code Smells

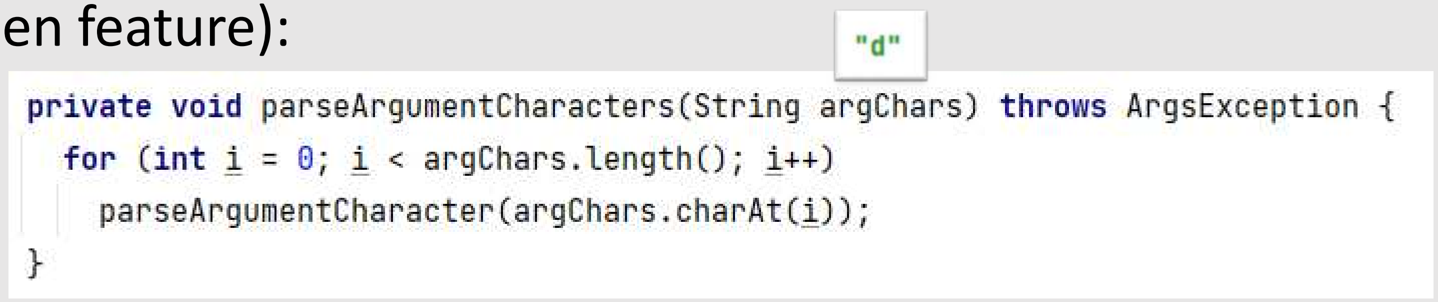
single responsibility:



```
public Args(String schema, String[] args) throws ArgsException {
  marshalers = new HashMap<Character, ArgumentMarshaler>();
  argsFound = new HashSet<Character>();

  parseSchema(schema);
  parseArgumentStrings(Arrays.asList(args));
}
```

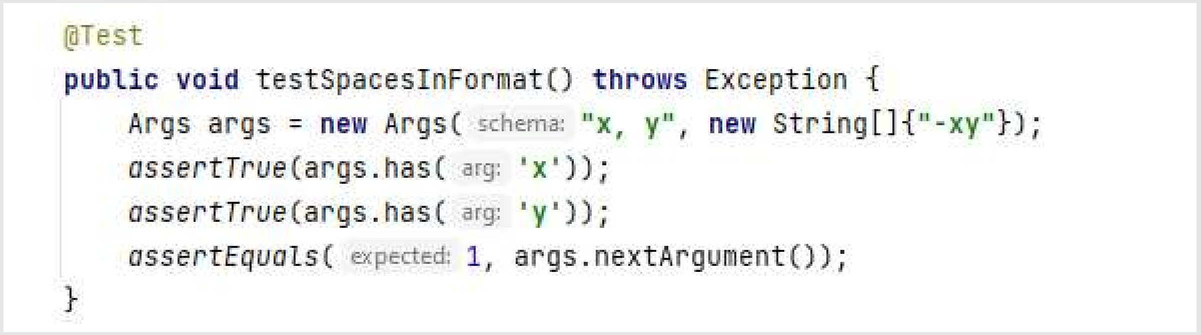
Uncle Bob‘s Version: Code Smells

WTF (hidden feature):



```
private void parseArgumentCharacters(String argChars) throws ArgsException {
  for (int i = 0; i < argChars.length(); i++)
    parseArgumentCharacter(argChars.charAt(i));
}
```

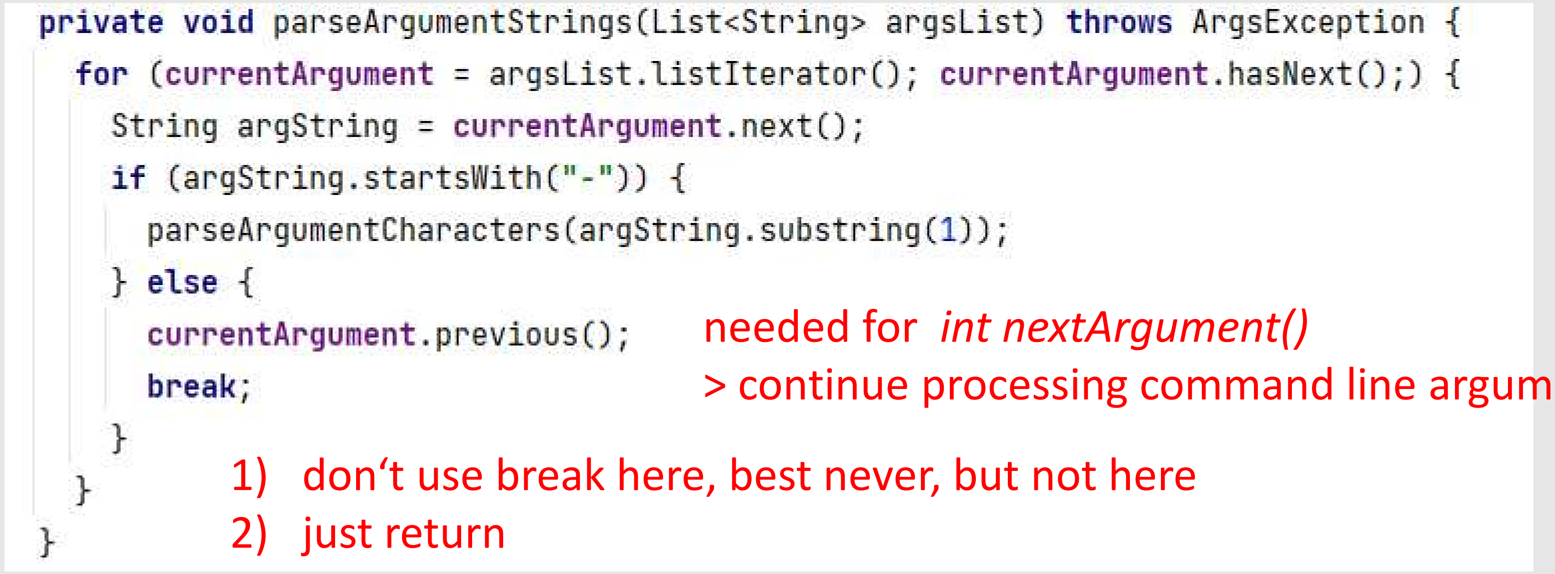
turns out, boolean attributes may be clumped together:



```
@Test
public void testSpacesInFormat() throws Exception {
    Args args = new Args( schema: "x, y", new String[]{"-xy"});
    assertTrue(args.has( arg: 'x'));
    assertTrue(args.has( arg: 'y'));
    assertEquals( expected: 1, args.nextArgument());
}
```

Uncle Bob‘s Version: Code Smells

WTF (hidden feature) & break:



```java
private void parseArgumentStrings(List<String> argsList) throws ArgsException {
  for (currentArgument = argsList.listIterator(); currentArgument.hasNext();) {
    String argString = currentArgument.next();
    if (argString.startsWith("-")) {
      parseArgumentCharacters(argString.substring(1));
    } else {
      currentArgument.previous();
      break;
    }
  }
}
```

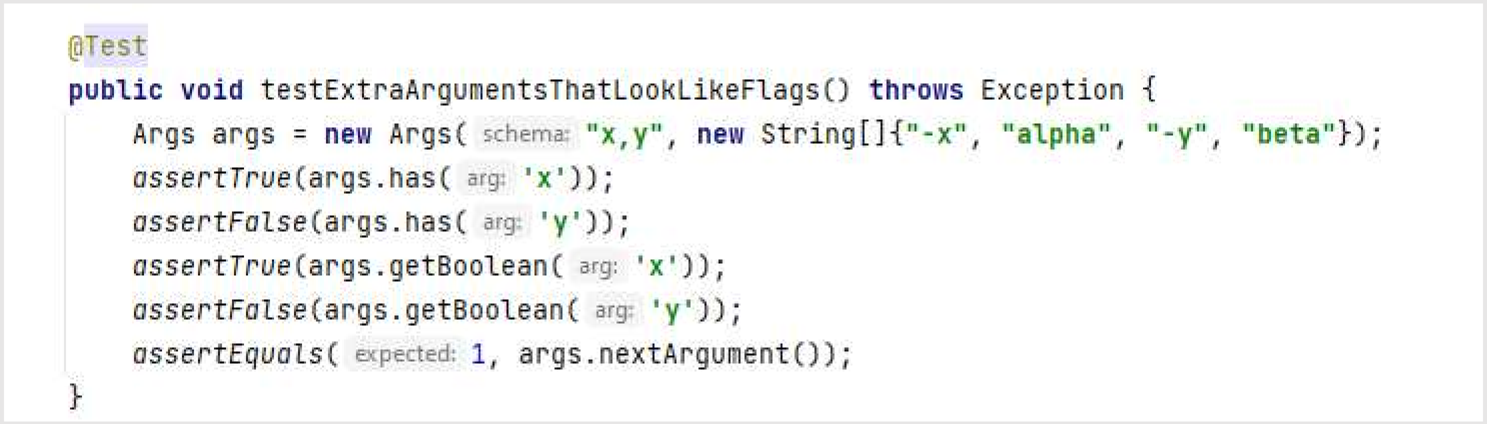
needed for *int nextArgument()*

> continue processing command line argument

1) don‘t use break here, best never, but not here
2) just return

Uncle Bob‘s Version: Code Smells

wtf (hidden feature):

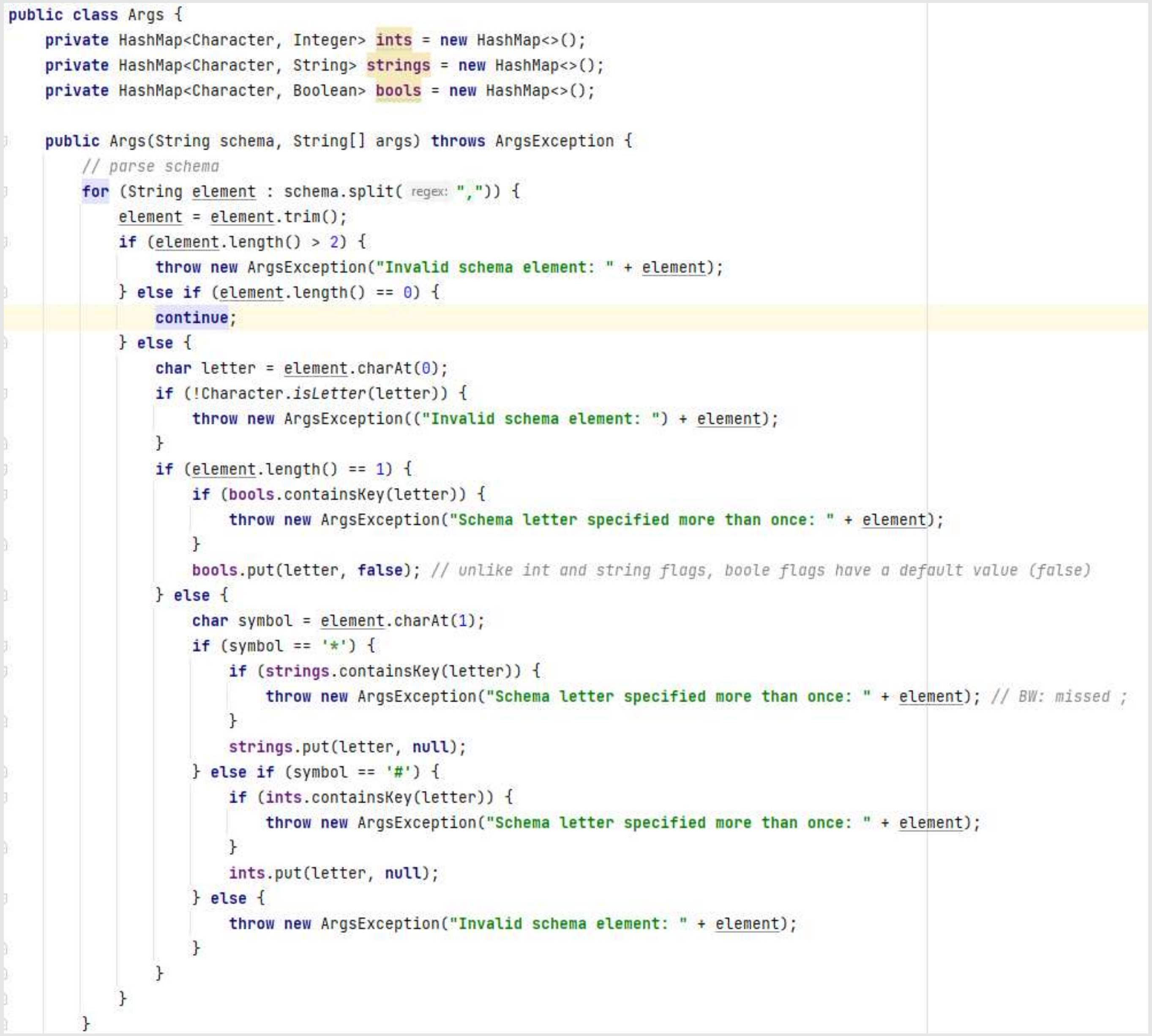


```
@Test
public void testExtraArgumentsThatLookLikeFlags() throws Exception {
    Args args = new Args( schema: "x,y", new String[]{"-x", "alpha", "-y", "beta"});
    assertTrue(args.has( arg: 'x'));
    assertFalse(args.has( arg: 'y'));
    assertTrue(args.getBoolean( arg: 'x'));
    assertFalse(args.getBoolean( arg: 'y'));
    assertEquals( expected: 1, args.nextArgument());
}
```

turns out, parsing stops, as soon as the first „free“ parameter is met.
> ls –al mySubDirectory –notAnArgument

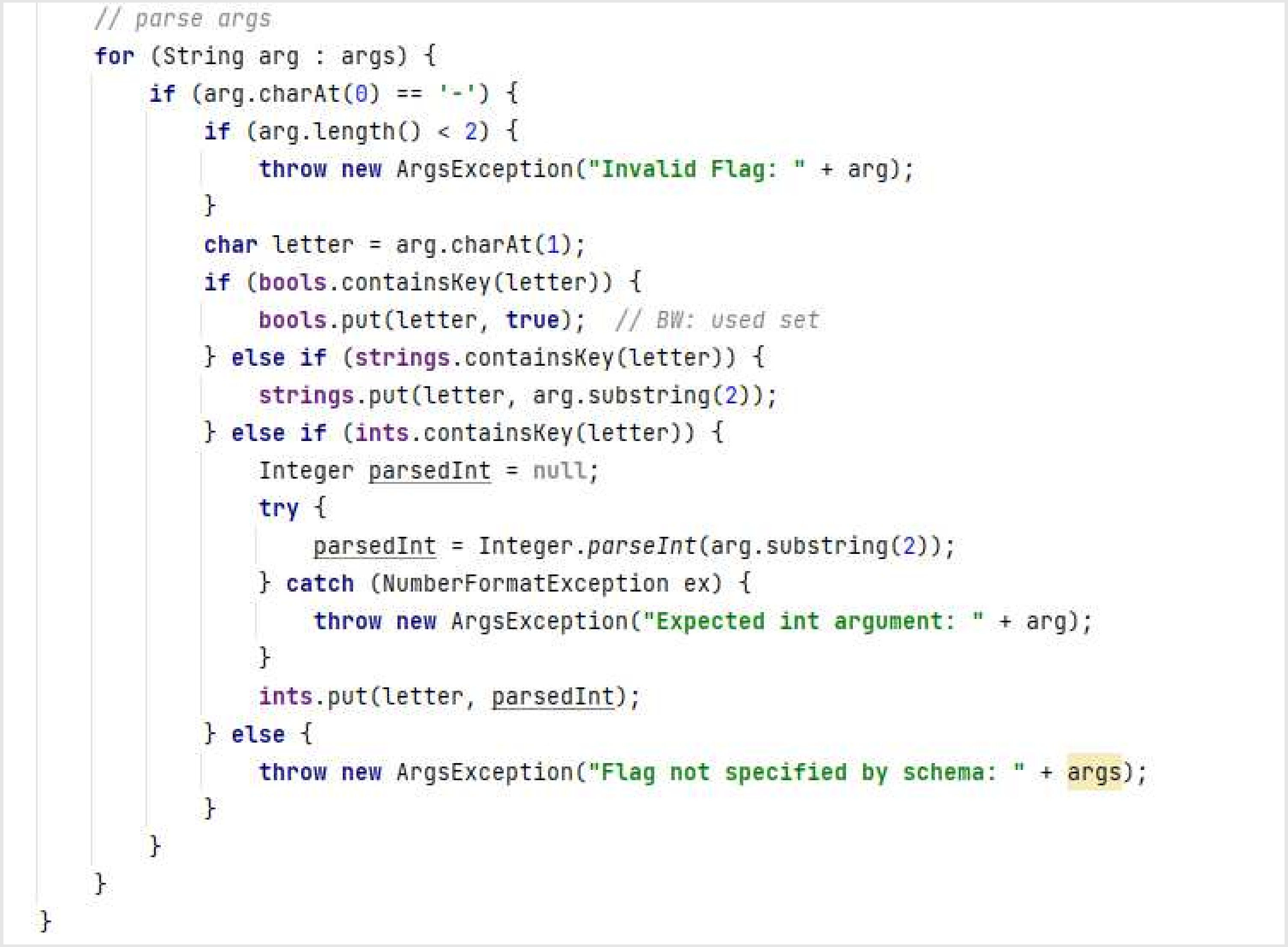
Just like a Unix Command, but it‘s never mentioned.

Brian Will‘s Version I



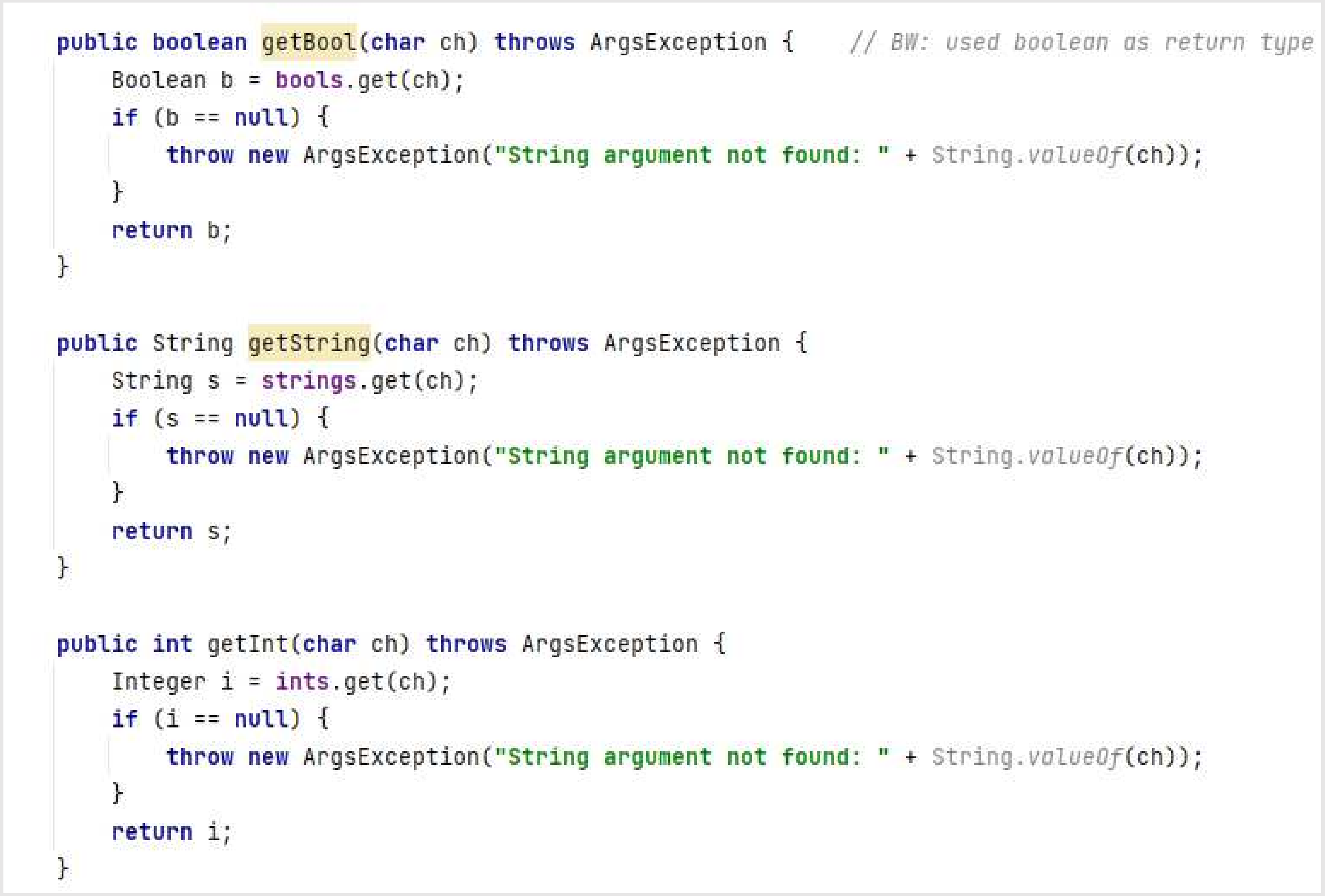
```java
public class Args {
    private HashMap<Character, Integer> ints = new HashMap<>();
    private HashMap<Character, String> strings = new HashMap<>();
    private HashMap<Character, Boolean> bools = new HashMap<>();

    public Args(String schema, String[] args) throws ArgsException {
        // parse schema
        for (String element : schema.split( regex: ",")) {
            element = element.trim();
            if (element.length() > 2) {
                throw new ArgsException("Invalid schema element: " + element);
            } else if (element.length() == 0) {
                continue;
            } else {
                char letter = element.charAt(0);
                if (!Character.isLetter(letter)) {
                    throw new ArgsException(("Invalid schema element: ") + element);
                }
                if (element.length() == 1) {
                    if (bools.containsKey(letter)) {
                        throw new ArgsException("Schema letter specified more than once: " + element);
                    }
                    bools.put(letter, false); // unlike int and string flags, boole flags have a default value (false)
                } else {
                    char symbol = element.charAt(1);
                    if (symbol == '*') {
                        if (strings.containsKey(letter)) {
                            throw new ArgsException("Schema letter specified more than once: " + element); // BW: missed ;
                        }
                        strings.put(letter, null);
                    } else if (symbol == '#') {
                        if (ints.containsKey(letter)) {
                            throw new ArgsException("Schema letter specified more than once: " + element);
                        }
                        ints.put(letter, null);
                    } else {
                        throw new ArgsException("Invalid schema element: " + element);
                    }
                }
            }
        }
```

Brian Will‘s Version II



```
    // parse args
    for (String arg : args) {
        if (arg.charAt(0) == '-') {
            if (arg.length() < 2) {
                throw new ArgsException("Invalid Flag: " + arg);
            }
            char letter = arg.charAt(1);
            if (bools.containsKey(letter)) {
                bools.put(letter, true);  // BW: used set
            } else if (strings.containsKey(letter)) {
                strings.put(letter, arg.substring(2));
            } else if (ints.containsKey(letter)) {
                Integer parsedInt = null;
                try {
                    parsedInt = Integer.parseInt(arg.substring(2));
                } catch (NumberFormatException ex) {
                    throw new ArgsException("Expected int argument: " + arg);
                }
                ints.put(letter, parsedInt);
            } else {
                throw new ArgsException("Flag not specified by schema: " + args);
            }
        }
    }
}
```

Brian Will‘s Version III



```
public boolean getBool(char ch) throws ArgsException {    // BW: used boolean as return type
    Boolean b = bools.get(ch);
    if (b == null) {
        throw new ArgsException("String argument not found: " + String.valueOf(ch));
    }
    return b;
}

public String getString(char ch) throws ArgsException {
    String s = strings.get(ch);
    if (s == null) {
        throw new ArgsException("String argument not found: " + String.valueOf(ch));
    }
    return s;
}

public int getInt(char ch) throws ArgsException {
    Integer i = ints.get(ch);
    if (i == null) {
        throw new ArgsException("String argument not found: " + String.valueOf(ch));
    }
    return i;
}
```

But ...

- Brian Will‘s code:
  - Is not available
  - Missing semicolons
  - map.set() instead map.put()
  - needs a different parameter format:

  > myProgram -p123 -dsmells -l

- Uncle Bob‘s code presented in Brian Will‘s video
  - Marshalers have more methods
  - The interface of the marshalers does not match the call

➡ It‘s all a big scam

Questions?

Ressources

- Brian Will: Youtube Video, Object-Oriented Programming is Embarrassing: 4 Short Examples, https://youtu.be/IRTfhkiAqPw
- Robert C. Martin: Clean Code, Prentice Hall
- https://codingdojo.org/kata/Args/
- Uncle Bob‘s Version: https://github.com/unclebob/javaargs

Thanks for attending