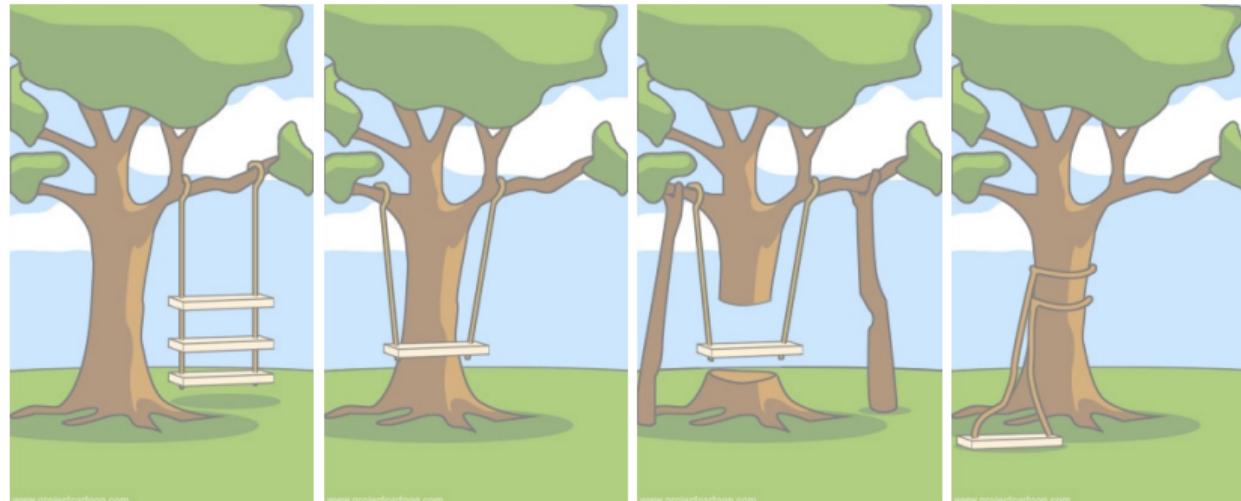




Software Engineering

6. Implementation | Thomas Thüm | November 24, 2021

Implementation in Software Projects



how the customer
explained it

how the project
leader understood it

how the analyst
designed it

how the programmer
implemented it

Lecture Overview

1. Programming Languages
2. Coding Conventions
3. Tools and Environments

Lecture Contents

1. Programming Languages

Questionnaire Results

Analysis and Design

History of Programming Languages

Programming Languages Today

Choice of Programming Languages

Popularity of Programming Languages

Excursion: Windows Calc

Lessons Learned

2. Coding Conventions

3. Tools and Environments

Questionnaire Results

1

Warst du jemals betroffen von einer Softwarepanne?

Antworten	relative Häufigkeit	absolute Häufigkeit
Ja	82%	59
Nein	18%	13

2

Hast du dich im letzten Semester über Softwarefehler geärgert?

Antworten	relative Häufigkeit	absolute Häufigkeit
Ja	88%	61
Nein	12%	8

3

Hast du selbst Programmiererfahrung?

Antworten	relative Häufigkeit	absolute Häufigkeit
Ja	93%	65
Nein	7%	5

4

Hast du bereits Software getestet?

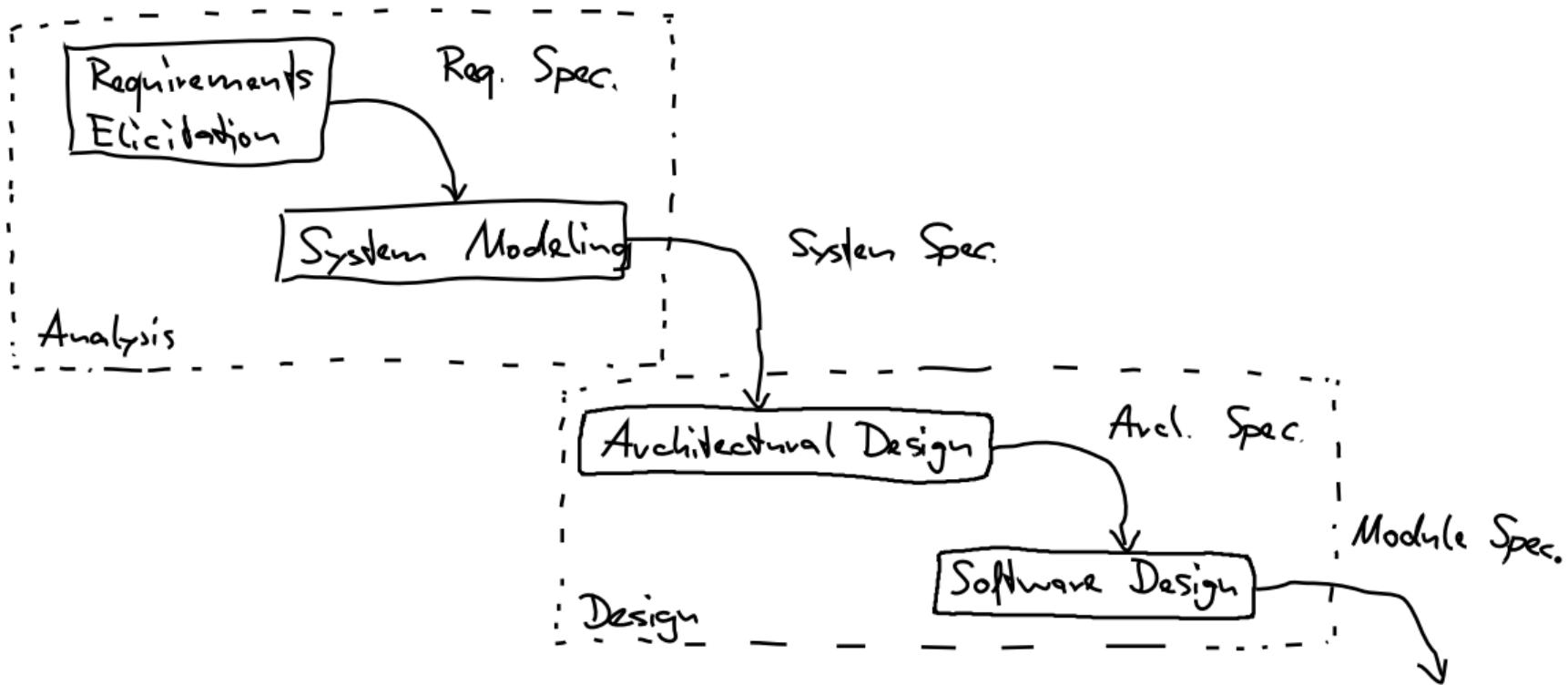
Antworten	relative Häufigkeit	absolute Häufigkeit
Ja	53%	38
Nein	47%	34

5

Hast du schon mal Programmierfehler verursacht?

Antworten	relative Häufigkeit	absolute Häufigkeit
Ja	87%	60
Nein	13%	9

Analysis and Design



History of Programming Languages

Languages

[Jones + Krypczyk/Bochkor]

- 1945: first high-level language Plankalkül by Konrad Zuse (compiler written in 1998)
- 1954: first professional high-level language FORTRAN (Formula Translator) by IBM
- 1963: Basic as general-purpose language
- 1959: functional language Lisp
- 1970: first object-oriented lang. Smalltalk-80
- 1970: declarative language SQL
- 1971: Pascal by Niklaus Wirth for teaching
- 1974: very common procedural language C
- 1977: logical language Prolog
- 1980: C++ as object-oriented extension of C
- 1990: object-oriented language Java
- 1990: functional language Haskell
- 1991: multi-paradigm language Python
- 1995: scripting language JavaScript

Milestones

[Jones]

- controlling behavior of mechanical devices by wiring or with punchcards ([Lochkarten](#))
- machine languages used during World War II
- assembly languages: distinction between human-readable instructions (source code) and executable instructions (object code)
- birth of compilers and interpreters having a one-to-many mapping between source and object code (opposed to one-to-one mapping in assemblers)
- structured programming pioneered by David Parnas and Edsger Dijkstra
- high-level programming languages: high number of executable for each human-readable instruction
- domain-specific languages, later general-purpose programming languages

Programming Languages Today

Today

[Jones + Krypczyk/Bochkor]

- 2002: C# by Microsoft
- 2009: Go by Google
- 2010: Rust by Mozilla Research
- 2014: Swift by Apple
- thousands of programming languages
- very few programming languages used for more than 10 years
- languages used for more than 25 years: Ada, C, C++, COBOL, Java, Objective C, PL/I, SQL, Visual Basic, ...

Many Languages

[Jones]

- good: fit for every use case
- bad: developer training for new and dead languages, expensive tool support

Choice of Programming Languages

Desired Properties

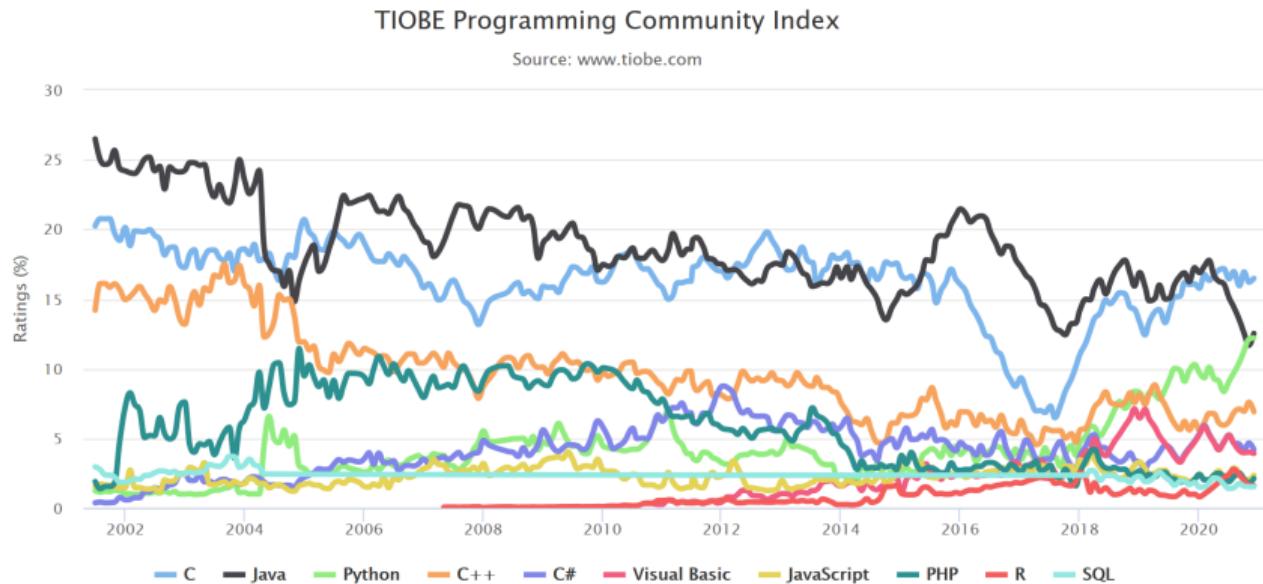
[Ludewig and Licher]

- modular implementation
- separation of interfaces and implementations
- type system: strongly/weakly typed languages
- readable syntax (FORTRAN vs ALGOL60)
- automatic pointer management (C vs Java)
- exception handling

Criteria in Practice

- language required by the company or customer?
- existing infrastructure?
- domain-specific languages available?
- language known/liked by developers?
- available libraries?
- available tool support?
- language popularity?
- what may change in the future?

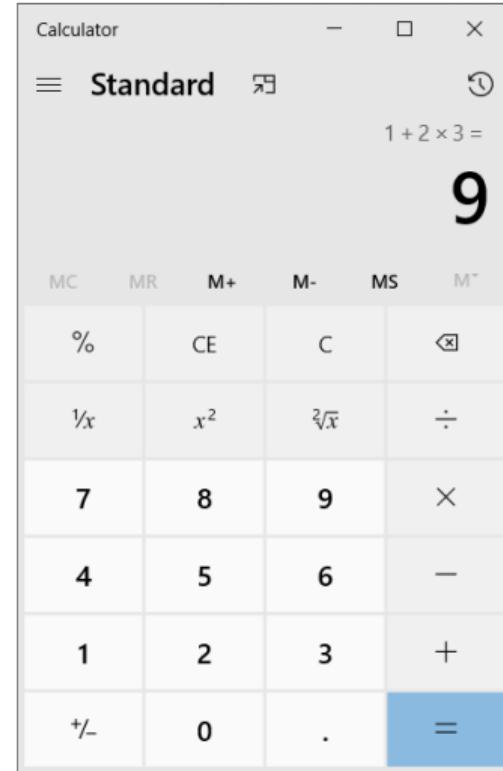
Popularity of Programming Languages



Popularity of Programming Languages

Programming Language	2020	2015	2010	2005	2000	1995	1990	1985
C	1	2	2	1	1	1	1	1
Java	2	1	1	2	3	29	-	-
Python	3	5	6	7	22	13	-	-
C++	4	3	3	3	2	2	2	8
C#	5	4	5	6	10	-	-	-
JavaScript	6	8	10	10	7	-	-	-
PHP	7	6	4	4	19	-	-	-
SQL	8	-	-	-	-	-	-	-
R	9	14	46	-	-	-	-	-
Swift	10	15	-	-	-	-	-	-
Lisp	29	26	14	13	9	6	4	2
Fortran	31	21	24	14	13	14	3	5
Ada	34	23	21	16	17	3	9	3

Excursion: Windows Calc





Patrick McKenzie

[[twitter.com](#)]

“Every great developer you know got there by solving problems they were unqualified to solve until they actually did it.”



Bill Gates

[[code.org](#)]

“Learning to write programs stretches your mind, and helps you think better, creates a way of thinking about things that I think is helpful in all domains.”

Programming Languages

Lessons Learned

- Historical perspective on programming
- Criteria for choosing languages
- Popularity of programming languages
- Further Reading on Programming Languages: [Jones](#), Chapter 8 Programming and Code Development +
[Krypczyk/Bochkor](#), Chapter 2.4 Programming Languages

Practice

- See [Moodle](#)
- Look at the [code of my calculator](#) and think about possible improvements to the code quality
- Share your thoughts with your colleagues in Moodle (before watching Part 2)

Lecture Contents

1. Programming Languages

2. Coding Conventions

Understanding the Corona-Warn-App

Code Formatting

Rules on Naming

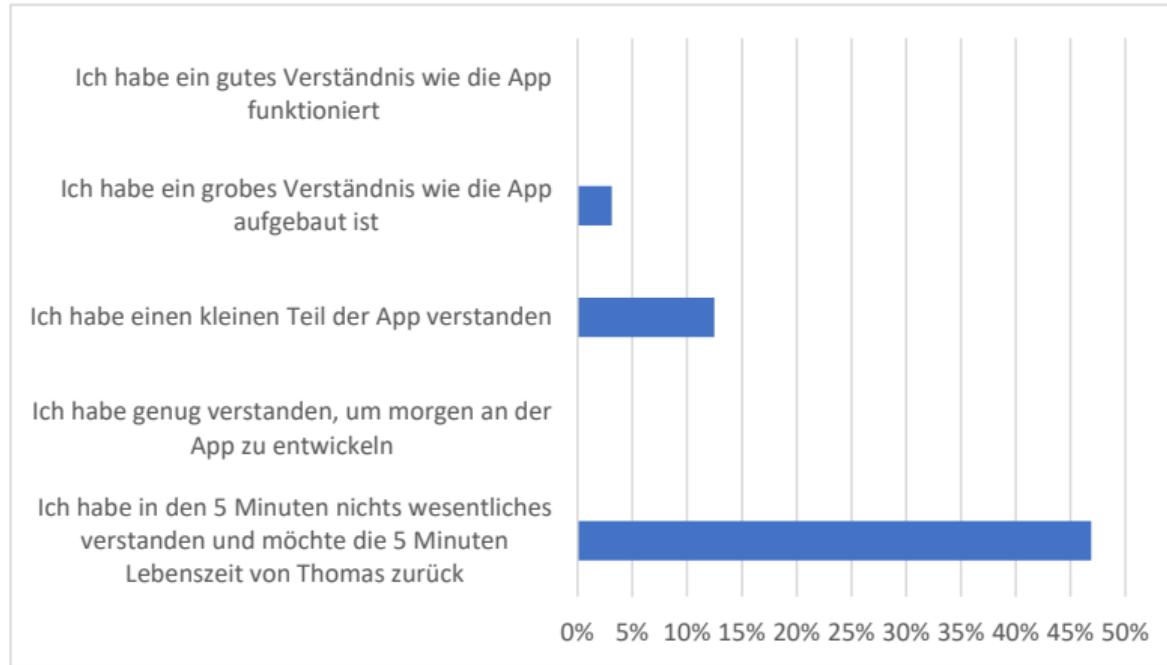
Code Documentation

Excursion: Revising Thomas' Calculator

Lessons Learned

3. Tools and Environments

Understanding the Corona-Warn-App





Douglas Crockford

[Crockford 2008]

"It turns out that style matters in programming for the same reason that it matters in writing. It makes for better reading."



François Chollet

[twitter.com]

"In software, naming matters, because names reflect how you think about a problem. Code is also communication, and naming is a big part of making it work."

Code Formatting

Code Formatting

- motivation: code is read much more often and by more developers than written
- avoid differences by each programmer
- indentation: typically 4 characters per level
- length of a line: often 80 or 100 characters
- extra indentation: typically 8 characters when breaking extra long lines
- empty lines between methods and attributes
- automated code formatters available (on demand or when saving the editor)
- typical formatting rules for each language
- automated code formatters are configurable (handle with care)

Rules on Naming

Unwanted Names

- single character as a name
- very long names
- names consisting only of special chars
- synonyms: delete, remove, clear
- abbreviations (unless very common)

Wanted Names

- nouns for class names: Calculator
- nouns for attribute names: calculateButton
- verbs for method names: getCalculator(), evaluate(), isZero(), hasChildren(), setValue()
- CamelCaseNotation for classes, attributes, methods, local variables, parameters
- UPPER_CASE_NOTATION for constants
- lowercasenotation for package names



Martin Fowler (1999)

[Fowler's Refactoring]

“Any fool can write code that a computer can understand. Good programmers write code that humans can understand.”



Cory House

[twitter.com]

“Code is like humor. When you have to explain it, it’s bad.”

Code Documentation

Comments ...

- in source code are easier to maintain (than in external documents)
- should be written while editing the code
- can be used to generate documentation (e.g., JavaDoc, Doxygen)
- are used to specify classes and public methods (e.g., parameters, exceptions, dependencies)
- document hacks, side effects and unfinished parts (e.g., TODO)
- should not paraphrase the code



Ryan Campbell

[problemsolving.io]

“Commenting your code is like cleaning your bathroom - you never want to do it, but it really does create a more pleasant experience for you and your guests.”

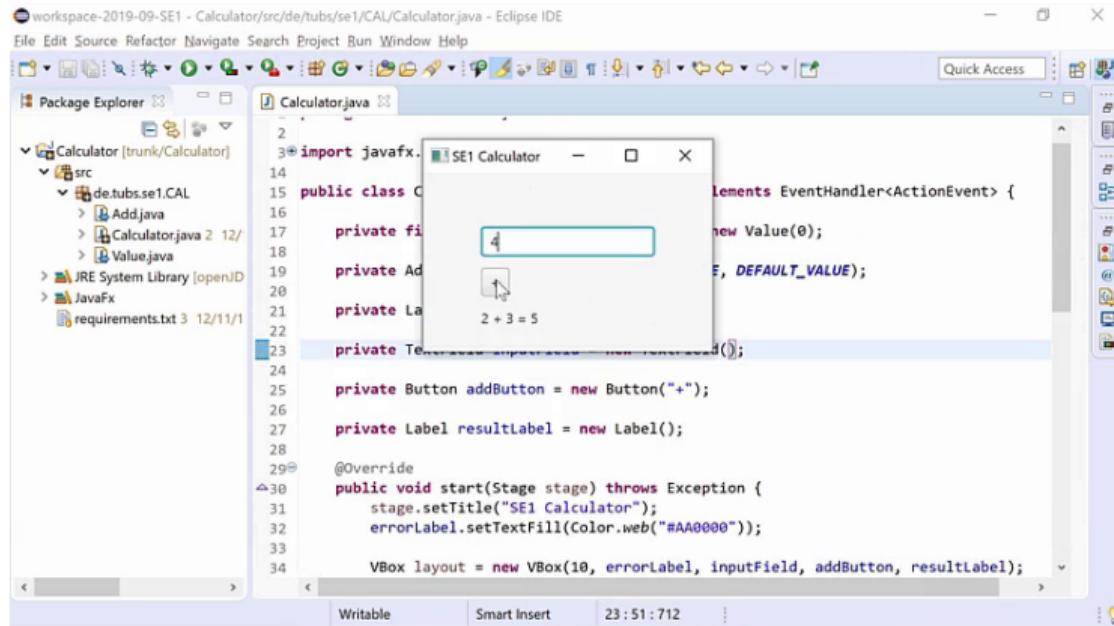


Steve McConnell (2004)

[Code Complete]

“Good code is its own best documentation. As you’re about to add a comment, ask yourself, “How can I improve the code so that this comment isn’t needed?” Improve the code and then document it to make it even clearer.”

Excursion: Revising Thomas' Calculator



The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** workspace-2019-09-SE1 - Calculator/src/de/tubs/se1/CAL/Calculator.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- Package Explorer:** Shows the project structure:
 - Calculator [trunk/Calculator]
 - src
 - de.tubs.se1.CAL
 - Add.java
 - Calculator.java 2 12/
 - Value.java
 - JRE System Library [openJD]
 - JavaFx
 - requirements.txt 3 12/11/
- Code Editor:** Displays the `Calculator.java` file content:

```
2
3 import javafx...
4
5 public class C...
6
7     private fi...
8
9     private Ad...
10
11    private La...
12
13    private Te...
14
15    private Button addButton = new Button("+");
16
17    private Label resultLabel = new Label();
18
19    @Override
20    public void start(Stage stage) throws Exception {
21        stage.setTitle("SE1 Calculator");
22        errorLabel.setTextFill(Color.web("#AA0000"));
23        VBox layout = new VBox(10, errorLabel, inputField, addButton, resultLabel);
24    }
25
26    private void handleButtonAction(ActionEvent event) {
27        String text = ((Button) event.getSource()).getText();
28        if ("C".equals(text)) {
29            clear();
30        } else if (".".equals(text)) {
31            if (!inputField.getText().contains(".")) {
32                inputField.appendText(".");
33            }
34        } else if ("=".equals(text)) {
35            calculate();
36        } else if ("0".equals(text)) {
37            if (!inputField.getText().contains("0")) {
38                inputField.appendText("0");
39            }
40        } else if ("1".equals(text)) {
41            if (!inputField.getText().contains("1")) {
42                inputField.appendText("1");
43            }
44        } else if ("2".equals(text)) {
45            if (!inputField.getText().contains("2")) {
46                inputField.appendText("2");
47            }
48        } else if ("3".equals(text)) {
49            if (!inputField.getText().contains("3")) {
50                inputField.appendText("3");
51            }
52        } else if ("4".equals(text)) {
53            if (!inputField.getText().contains("4")) {
54                inputField.appendText("4");
55            }
56        } else if ("5".equals(text)) {
57            if (!inputField.getText().contains("5")) {
58                inputField.appendText("5");
59            }
60        } else if ("6".equals(text)) {
61            if (!inputField.getText().contains("6")) {
62                inputField.appendText("6");
63            }
64        } else if ("7".equals(text)) {
65            if (!inputField.getText().contains("7")) {
66                inputField.appendText("7");
67            }
68        } else if ("8".equals(text)) {
69            if (!inputField.getText().contains("8")) {
70                inputField.appendText("8");
71            }
72        } else if ("9".equals(text)) {
73            if (!inputField.getText().contains("9")) {
74                inputField.appendText("9");
75            }
76        } else if ("+".equals(text)) {
77            if (!inputField.getText().contains("+")) {
78                inputField.appendText("+");
79            }
80        } else if ("-".equals(text)) {
81            if (!inputField.getText().contains("-")) {
82                inputField.appendText("-");
83            }
84        } else if ("*".equals(text)) {
85            if (!inputField.getText().contains("*")) {
86                inputField.appendText("*");
87            }
88        } else if ("/".equals(text)) {
89            if (!inputField.getText().contains("/")) {
90                inputField.appendText("/");
91            }
92        }
93    }
94
95    private void calculate() {
96        try {
97            double value = Double.parseDouble(inputField.getText());
98            resultLabel.setText(String.valueOf(value));
99        } catch (Exception e) {
100            errorLabel.setText("Error");
101        }
102    }
103
104    private void clear() {
105        inputField.setText("");
106        resultLabel.setText("0");
107    }
108
109    private void appendText(String text) {
110        inputField.appendText(text);
111    }
112
113    private void setErrorText(String text) {
114        errorLabel.setText(text);
115    }
116
117    private void setLabelText(String text) {
118        resultLabel.setText(text);
119    }
120
121    private void setAddButton() {
122        addButton.setOnAction(this::handleButtonAction);
123    }
124
125    private void setClearButton() {
126        clearButton.setOnAction(this::handleButtonAction);
127    }
128
129    private void setDotButton() {
130        dotButton.setOnAction(this::handleButtonAction);
131    }
132
133    private void setEqualButton() {
134        equalButton.setOnAction(this::handleButtonAction);
135    }
136
137    private void setMinusButton() {
138        minusButton.setOnAction(this::handleButtonAction);
139    }
140
141    private void setMultiplyButton() {
142        multiplyButton.setOnAction(this::handleButtonAction);
143    }
144
145    private void setPlusButton() {
146        plusButton.setOnAction(this::handleButtonAction);
147    }
148
149    private void setZeroButton() {
150        zeroButton.setOnAction(this::handleButtonAction);
151    }
152
153    private void setOneButton() {
154        oneButton.setOnAction(this::handleButtonAction);
155    }
156
157    private void setTwoButton() {
158        twoButton.setOnAction(this::handleButtonAction);
159    }
160
161    private void setThreeButton() {
162        threeButton.setOnAction(this::handleButtonAction);
163    }
164
165    private void setFourButton() {
166        fourButton.setOnAction(this::handleButtonAction);
167    }
168
169    private void setFiveButton() {
170        fiveButton.setOnAction(this::handleButtonAction);
171    }
172
173    private void setSixButton() {
174        sixButton.setOnAction(this::handleButtonAction);
175    }
176
177    private void setSevenButton() {
178        sevenButton.setOnAction(this::handleButtonAction);
179    }
180
181    private void setEightButton() {
182        eightButton.setOnAction(this::handleButtonAction);
183    }
184
185    private void setNineButton() {
186        nineButton.setOnAction(this::handleButtonAction);
187    }
188
189    private void setVBox() {
190        VBox.setVBox();
191    }
192
193    private void setInputField() {
194        inputField.setEditable(true);
195    }
196
197    private void setErrorLabel() {
198        errorLabel.setVisible(true);
199    }
200
201    private void setResultLabel() {
202        resultLabel.setVisible(true);
203    }
204
205    private void setAddLabel() {
206        addLabel.setVisible(true);
207    }
208
209    private void setClearLabel() {
210        clearLabel.setVisible(true);
211    }
212
213    private void setDotLabel() {
214        dotLabel.setVisible(true);
215    }
216
217    private void setEqualLabel() {
218        equalLabel.setVisible(true);
219    }
220
221    private void setMinusLabel() {
222        minusLabel.setVisible(true);
223    }
224
225    private void setMultiplyLabel() {
226        multiplyLabel.setVisible(true);
227    }
228
229    private void setPlusLabel() {
230        plusLabel.setVisible(true);
231    }
232
233    private void setZeroLabel() {
234        zeroLabel.setVisible(true);
235    }
236
237    private void setOneLabel() {
238        oneLabel.setVisible(true);
239    }
240
241    private void setTwoLabel() {
242        twoLabel.setVisible(true);
243    }
244
245    private void setThreeLabel() {
246        threeLabel.setVisible(true);
247    }
248
249    private void setFourLabel() {
250        fourLabel.setVisible(true);
251    }
252
253    private void setFiveLabel() {
254        fiveLabel.setVisible(true);
255    }
256
257    private void setSixLabel() {
258        sixLabel.setVisible(true);
259    }
260
261    private void setSevenLabel() {
262        sevenLabel.setVisible(true);
263    }
264
265    private void setEightLabel() {
266        eightLabel.setVisible(true);
267    }
268
269    private void setNineLabel() {
270        nineLabel.setVisible(true);
271    }
272
273    private void setVBox() {
274        VBox.setVBox();
275    }
276
277    private void setInputField() {
278        inputField.setEditable(true);
279    }
280
281    private void setErrorLabel() {
282        errorLabel.setVisible(true);
283    }
284
285    private void setResultLabel() {
286        resultLabel.setVisible(true);
287    }
288
289    private void setAddLabel() {
290        addLabel.setVisible(true);
291    }
292
293    private void setClearLabel() {
294        clearLabel.setVisible(true);
295    }
296
297    private void setDotLabel() {
298        dotLabel.setVisible(true);
299    }
300
301    private void setEqualLabel() {
302        equalLabel.setVisible(true);
303    }
304
305    private void setMinusLabel() {
306        minusLabel.setVisible(true);
307    }
308
309    private void setMultiplyLabel() {
310        multiplyLabel.setVisible(true);
311    }
312
313    private void setPlusLabel() {
314        plusLabel.setVisible(true);
315    }
316
317    private void setZeroLabel() {
318        zeroLabel.setVisible(true);
319    }
320
321    private void setOneLabel() {
322        oneLabel.setVisible(true);
323    }
324
325    private void setTwoLabel() {
326        twoLabel.setVisible(true);
327    }
328
329    private void setThreeLabel() {
330        threeLabel.setVisible(true);
331    }
332
333    private void setFourLabel() {
334        fourLabel.setVisible(true);
335    }
336
337    private void setFiveLabel() {
338        fiveLabel.setVisible(true);
339    }
340
341    private void setSixLabel() {
342        sixLabel.setVisible(true);
343    }
344
345    private void setSevenLabel() {
346        sevenLabel.setVisible(true);
347    }
348
349    private void setEightLabel() {
350        eightLabel.setVisible(true);
351    }
352
353    private void setNineLabel() {
354        nineLabel.setVisible(true);
355    }
356
357    private void setVBox() {
358        VBox.setVBox();
359    }
360
361    private void setInputField() {
362        inputField.setEditable(true);
363    }
364
365    private void setErrorLabel() {
366        errorLabel.setVisible(true);
367    }
368
369    private void setResultLabel() {
370        resultLabel.setVisible(true);
371    }
372
373    private void setAddLabel() {
374        addLabel.setVisible(true);
375    }
376
377    private void setClearLabel() {
378        clearLabel.setVisible(true);
379    }
380
381    private void setDotLabel() {
382        dotLabel.setVisible(true);
383    }
384
385    private void setEqualLabel() {
386        equalLabel.setVisible(true);
387    }
388
389    private void setMinusLabel() {
390        minusLabel.setVisible(true);
391    }
392
393    private void setMultiplyLabel() {
394        multiplyLabel.setVisible(true);
395    }
396
397    private void setPlusLabel() {
398        plusLabel.setVisible(true);
399    }
400
401    private void setZeroLabel() {
402        zeroLabel.setVisible(true);
403    }
404
405    private void setOneLabel() {
406        oneLabel.setVisible(true);
407    }
408
409    private void setTwoLabel() {
410        twoLabel.setVisible(true);
411    }
412
413    private void setThreeLabel() {
414        threeLabel.setVisible(true);
415    }
416
417    private void setFourLabel() {
418        fourLabel.setVisible(true);
419    }
420
421    private void setFiveLabel() {
422        fiveLabel.setVisible(true);
423    }
424
425    private void setSixLabel() {
426        sixLabel.setVisible(true);
427    }
428
429    private void setSevenLabel() {
430        sevenLabel.setVisible(true);
431    }
432
433    private void setEightLabel() {
434        eightLabel.setVisible(true);
435    }
436
437    private void setNineLabel() {
438        nineLabel.setVisible(true);
439    }
440
441    private void setVBox() {
442        VBox.setVBox();
443    }
444
445    private void setInputField() {
446        inputField.setEditable(true);
447    }
448
449    private void setErrorLabel() {
450        errorLabel.setVisible(true);
451    }
452
453    private void setResultLabel() {
454        resultLabel.setVisible(true);
455    }
456
457    private void setAddLabel() {
458        addLabel.setVisible(true);
459    }
460
461    private void setClearLabel() {
462        clearLabel.setVisible(true);
463    }
464
465    private void setDotLabel() {
466        dotLabel.setVisible(true);
467    }
468
469    private void setEqualLabel() {
470        equalLabel.setVisible(true);
471    }
472
473    private void setMinusLabel() {
474        minusLabel.setVisible(true);
475    }
476
477    private void setMultiplyLabel() {
478        multiplyLabel.setVisible(true);
479    }
480
481    private void setPlusLabel() {
482        plusLabel.setVisible(true);
483    }
484
485    private void setZeroLabel() {
486        zeroLabel.setVisible(true);
487    }
488
489    private void setOneLabel() {
490        oneLabel.setVisible(true);
491    }
492
493    private void setTwoLabel() {
494        twoLabel.setVisible(true);
495    }
496
497    private void setThreeLabel() {
498        threeLabel.setVisible(true);
499    }
500
501    private void setFourLabel() {
502        fourLabel.setVisible(true);
503    }
504
505    private void setFiveLabel() {
506        fiveLabel.setVisible(true);
507    }
508
509    private void setSixLabel() {
510        sixLabel.setVisible(true);
511    }
512
513    private void setSevenLabel() {
514        sevenLabel.setVisible(true);
515    }
516
517    private void setEightLabel() {
518        eightLabel.setVisible(true);
519    }
520
521    private void setNineLabel() {
522        nineLabel.setVisible(true);
523    }
524
525    private void setVBox() {
526        VBox.setVBox();
527    }
528
529    private void setInputField() {
530        inputField.setEditable(true);
531    }
532
533    private void setErrorLabel() {
534        errorLabel.setVisible(true);
535    }
536
537    private void setResultLabel() {
538        resultLabel.setVisible(true);
539    }
540
541    private void setAddLabel() {
542        addLabel.setVisible(true);
543    }
544
545    private void setClearLabel() {
546        clearLabel.setVisible(true);
547    }
548
549    private void setDotLabel() {
550        dotLabel.setVisible(true);
551    }
552
553    private void setEqualLabel() {
554        equalLabel.setVisible(true);
555    }
556
557    private void setMinusLabel() {
558        minusLabel.setVisible(true);
559    }
560
561    private void setMultiplyLabel() {
562        multiplyLabel.setVisible(true);
563    }
564
565    private void setPlusLabel() {
566        plusLabel.setVisible(true);
567    }
568
569    private void setZeroLabel() {
570        zeroLabel.setVisible(true);
571    }
572
573    private void setOneLabel() {
574        oneLabel.setVisible(true);
575    }
576
577    private void setTwoLabel() {
578        twoLabel.setVisible(true);
579    }
580
581    private void setThreeLabel() {
582        threeLabel.setVisible(true);
583    }
584
585    private void setFourLabel() {
586        fourLabel.setVisible(true);
587    }
588
589    private void setFiveLabel() {
590        fiveLabel.setVisible(true);
591    }
592
593    private void setSixLabel() {
594        sixLabel.setVisible(true);
595    }
596
597    private void setSevenLabel() {
598        sevenLabel.setVisible(true);
599    }
599
600    private void setEightLabel() {
601        eightLabel.setVisible(true);
602    }
603
604    private void setNineLabel() {
605        nineLabel.setVisible(true);
606    }
607
608    private void setVBox() {
609        VBox.setVBox();
610    }
611
612    private void setInputField() {
613        inputField.setEditable(true);
614    }
615
616    private void setErrorLabel() {
617        errorLabel.setVisible(true);
618    }
619
620    private void setResultLabel() {
621        resultLabel.setVisible(true);
622    }
623
624    private void setAddLabel() {
625        addLabel.setVisible(true);
626    }
627
628    private void setClearLabel() {
629        clearLabel.setVisible(true);
630    }
631
632    private void setDotLabel() {
633        dotLabel.setVisible(true);
634    }
635
636    private void setEqualLabel() {
637        equalLabel.setVisible(true);
638    }
639
640    private void setMinusLabel() {
641        minusLabel.setVisible(true);
642    }
643
644    private void setMultiplyLabel() {
645        multiplyLabel.setVisible(true);
646    }
647
648    private void setPlusLabel() {
649        plusLabel.setVisible(true);
650    }
651
652    private void setZeroLabel() {
653        zeroLabel.setVisible(true);
654    }
655
656    private void setOneLabel() {
657        oneLabel.setVisible(true);
658    }
659
660    private void setTwoLabel() {
661        twoLabel.setVisible(true);
662    }
663
664    private void setThreeLabel() {
665        threeLabel.setVisible(true);
666    }
667
668    private void setFourLabel() {
669        fourLabel.setVisible(true);
670    }
671
672    private void setFiveLabel() {
673        fiveLabel.setVisible(true);
674    }
675
676    private void setSixLabel() {
677        sixLabel.setVisible(true);
678    }
679
680    private void setSevenLabel() {
681        sevenLabel.setVisible(true);
682    }
683
684    private void setEightLabel() {
685        eightLabel.setVisible(true);
686    }
687
688    private void setNineLabel() {
689        nineLabel.setVisible(true);
690    }
691
692    private void setVBox() {
693        VBox.setVBox();
694    }
695
696    private void setInputField() {
697        inputField.setEditable(true);
698    }
699
700    private void setErrorLabel() {
701        errorLabel.setVisible(true);
702    }
703
704    private void setResultLabel() {
705        resultLabel.setVisible(true);
706    }
707
708    private void setAddLabel() {
709        addLabel.setVisible(true);
710    }
711
712    private void setClearLabel() {
713        clearLabel.setVisible(true);
714    }
715
716    private void setDotLabel() {
717        dotLabel.setVisible(true);
718    }
719
720    private void setEqualLabel() {
721        equalLabel.setVisible(true);
722    }
723
724    private void setMinusLabel() {
725        minusLabel.setVisible(true);
726    }
727
728    private void setMultiplyLabel() {
729        multiplyLabel.setVisible(true);
730    }
731
732    private void setPlusLabel() {
733        plusLabel.setVisible(true);
734    }
735
736    private void setZeroLabel() {
737        zeroLabel.setVisible(true);
738    }
739
740    private void setOneLabel() {
741        oneLabel.setVisible(true);
742    }
743
744    private void setTwoLabel() {
745        twoLabel.setVisible(true);
746    }
747
748    private void setThreeLabel() {
749        threeLabel.setVisible(true);
750    }
751
752    private void setFourLabel() {
753        fourLabel.setVisible(true);
754    }
755
756    private void setFiveLabel() {
757        fiveLabel.setVisible(true);
758    }
759
760    private void setSixLabel() {
761        sixLabel.setVisible(true);
762    }
763
764    private void setSevenLabel() {
765        sevenLabel.setVisible(true);
766    }
767
768    private void setEightLabel() {
769        eightLabel.setVisible(true);
770    }
771
772    private void setNineLabel() {
773        nineLabel.setVisible(true);
774    }
775
776    private void setVBox() {
777        VBox.setVBox();
778    }
779
780    private void setInputField() {
781        inputField.setEditable(true);
782    }
783
784    private void setErrorLabel() {
785        errorLabel.setVisible(true);
786    }
787
788    private void setResultLabel() {
789        resultLabel.setVisible(true);
790    }
791
792    private void setAddLabel() {
793        addLabel.setVisible(true);
794    }
795
796    private void setClearLabel() {
797        clearLabel.setVisible(true);
798    }
799
800    private void setDotLabel() {
801        dotLabel.setVisible(true);
802    }
803
804    private void setEqualLabel() {
805        equalLabel.setVisible(true);
806    }
807
808    private void setMinusLabel() {
809        minusLabel.setVisible(true);
810    }
811
812    private void setMultiplyLabel() {
813        multiplyLabel.setVisible(true);
814    }
815
816    private void setPlusLabel() {
817        plusLabel.setVisible(true);
818    }
819
820    private void setZeroLabel() {
821        zeroLabel.setVisible(true);
822    }
823
824    private void setOneLabel() {
825        oneLabel.setVisible(true);
826    }
827
828    private void setTwoLabel() {
829        twoLabel.setVisible(true);
830    }
831
832    private void setThreeLabel() {
833        threeLabel.setVisible(true);
834    }
835
836    private void setFourLabel() {
837        fourLabel.setVisible(true);
838    }
839
840    private void setFiveLabel() {
841        fiveLabel.setVisible(true);
842    }
843
844    private void setSixLabel() {
845        sixLabel.setVisible(true);
846    }
847
848    private void setSevenLabel() {
849        sevenLabel.setVisible(true);
850    }
851
852    private void setEightLabel() {
853        eightLabel.setVisible(true);
854    }
855
856    private void setNineLabel() {
857        nineLabel.setVisible(true);
858    }
859
860    private void setVBox() {
861        VBox.setVBox();
862    }
863
864    private void setInputField() {
865        inputField.setEditable(true);
866    }
867
868    private void setErrorLabel() {
869        errorLabel.setVisible(true);
870    }
871
872    private void setResultLabel() {
873        resultLabel.setVisible(true);
874    }
875
876    private void setAddLabel() {
877        addLabel.setVisible(true);
878    }
879
880    private void setClearLabel() {
881        clearLabel.setVisible(true);
882    }
883
884    private void setDotLabel() {
885        dotLabel.setVisible(true);
886    }
887
888    private void setEqualLabel() {
889        equalLabel.setVisible(true);
890    }
891
892    private void setMinusLabel() {
893        minusLabel.setVisible(true);
894    }
895
896    private void setMultiplyLabel() {
897        multiplyLabel.setVisible(true);
898    }
899
900    private void setPlusLabel() {
901        plusLabel.setVisible(true);
902    }
903
904    private void setZeroLabel() {
905        zeroLabel.setVisible(true);
906    }
907
908    private void setOneLabel() {
909        oneLabel.setVisible(true);
910    }
911
912    private void setTwoLabel() {
913        twoLabel.setVisible(true);
914    }
915
916    private void setThreeLabel() {
917        threeLabel.setVisible(true);
918    }
919
920    private void setFourLabel() {
921        fourLabel.setVisible(true);
922    }
923
924    private void setFiveLabel() {
925        fiveLabel.setVisible(true);
926    }
927
928    private void setSixLabel() {
929        sixLabel.setVisible(true);
930    }
931
932    private void setSevenLabel() {
933        sevenLabel.setVisible(true);
934    }
935
936    private void setEightLabel() {
937        eightLabel.setVisible(true);
938    }
939
940    private void setNineLabel() {
941        nineLabel.setVisible(true);
942    }
943
944    private void setVBox() {
945        VBox.setVBox();
946    }
947
948    private void setInputField() {
949        inputField.setEditable(true);
950    }
951
952    private void setErrorLabel() {
953        errorLabel.setVisible(true);
954    }
955
956    private void setResultLabel() {
957        resultLabel.setVisible(true);
958    }
959
960    private void setAddLabel() {
961        addLabel.setVisible(true);
962    }
963
964    private void setClearLabel() {
965        clearLabel.setVisible(true);
966    }
967
968    private void setDotLabel() {
969        dotLabel.setVisible(true);
970    }
971
972    private void setEqualLabel() {
973        equalLabel.setVisible(true);
974    }
975
976    private void setMinusLabel() {
977        minusLabel.setVisible(true);
978    }
979
980    private void setMultiplyLabel() {
981        multiplyLabel.setVisible(true);
982    }
983
984    private void setPlusLabel() {
985        plusLabel.setVisible(true);
986    }
987
988    private void setZeroLabel() {
989        zeroLabel.setVisible(true);
990    }
991
992    private void setOneLabel() {
993        oneLabel.setVisible(true);
994    }
995
996    private void setTwoLabel() {
997        twoLabel.setVisible(true);
998    }
999
1000    private void setThreeLabel() {
1001        threeLabel.setVisible(true);
1002    }
1003
1004    private void setFourLabel() {
1005        fourLabel.setVisible(true);
1006    }
1007
1008    private void setFiveLabel() {
1009        fiveLabel.setVisible(true);
1010    }
1011
1012    private void setSixLabel() {
1013        sixLabel.setVisible(true);
1014    }
1015
1016    private void setSevenLabel() {
1017        sevenLabel.setVisible(true);
1018    }
1019
1020    private void setEightLabel() {
1021        eightLabel.setVisible(true);
1022    }
1023
1024    private void setNineLabel() {
1025        nineLabel.setVisible(true);
1026    }
1027
1028    private void setVBox() {
1029        VBox.setVBox();
1030    }
1031
1032    private void setInputField() {
1033        inputField.setEditable(true);
1034    }
1035
1036    private void setErrorLabel() {
1037        errorLabel.setVisible(true);
1038    }
1039
1040    private void setResultLabel() {
1041        resultLabel.setVisible(true);
1042    }
1043
1044    private void setAddLabel() {
1045        addLabel.setVisible(true);
1046    }
1047
1048    private void setClearLabel() {
1049        clearLabel.setVisible(true);
1050    }
1051
1052    private void setDotLabel() {
1053        dotLabel.setVisible(true);
1054    }
1055
1056    private void setEqualLabel() {
1057        equalLabel.setVisible(true);
1058    }
1059
1060    private void setMinusLabel() {
1061        minusLabel.setVisible(true);
1062    }
1063
1064    private void setMultiplyLabel() {
1065        multiplyLabel.setVisible(true);
1066    }
1067
1068    private void setPlusLabel() {
1069        plusLabel.setVisible(true);
1070    }
1071
1072    private void setZeroLabel() {
1073        zeroLabel.setVisible(true);
1074    }
1075
1076    private void setOneLabel() {
1077        oneLabel.setVisible(true);
1078    }
1079
1080    private void setTwoLabel() {
1081        twoLabel.setVisible(true);
1082    }
1083
1084    private void setThreeLabel() {
1085        threeLabel.setVisible(true);
1086    }
1087
1088    private void setFourLabel() {
1089        fourLabel.setVisible(true);
1090    }
1091
1092    private void setFiveLabel() {
1093        fiveLabel.setVisible(true);
1094    }
1095
1096    private void setSixLabel() {
1097        sixLabel.setVisible(true);
1098    }
1099
1100    private void setSevenLabel() {
1101        sevenLabel.setVisible(true);
1102    }
1103
1104    private void setEightLabel() {
1105        eightLabel.setVisible(true);
1106    }
1107
1108    private void setNineLabel() {
1109        nineLabel.setVisible(true);
1110    }
1111
1112    private void setVBox() {
1113        VBox.setVBox();
1114    }
1115
1116    private void setInputField() {
1117        inputField.setEditable(true);
1118    }
1119
1120    private void setErrorLabel() {
1121        errorLabel.setVisible(true);
1122    }
1123
1124    private void setResultLabel() {
1125        resultLabel.setVisible(true);
1126    }
1127
1128    private void setAddLabel() {
1129        addLabel.setVisible(true);
1130    }
1131
1132    private void setClearLabel() {
1133        clearLabel.setVisible(true);
1134    }
1135
1136    private void setDotLabel() {
1137        dotLabel.setVisible(true);
1138    }
1139
1140    private void setEqualLabel() {
1141        equalLabel.setVisible(true);
1142    }
1143
1144    private void setMinusLabel() {
1145        minusLabel.setVisible(true);
1146    }
1147
1148    private void setMultiplyLabel() {
1149        multiplyLabel.setVisible(true);
1150    }
1151
1152    private void setPlusLabel() {
1153        plusLabel.setVisible(true);
1154    }
1155
1156    private void setZeroLabel() {
1157        zeroLabel.setVisible(true);
1158    }
1159
1160    private void setOneLabel() {
1161        oneLabel.setVisible(true);
1162    }
1163
1164    private void setTwoLabel() {
1165        twoLabel.setVisible(true);
1166    }
1167
1168    private void setThreeLabel() {
1169        threeLabel.setVisible(true);
1170    }
1171
1172    private void setFourLabel() {
1173        fourLabel.setVisible(true);
1174    }
1175
1176    private void setFiveLabel() {
1177        fiveLabel.setVisible(true);
1178    }
1179
1180    private void setSixLabel() {
1181        sixLabel.setVisible(true);
1182    }
1183
1184    private void setSevenLabel() {
1185        sevenLabel.setVisible(true);
1186    }
1187
1188    private void setEightLabel() {
1189        eightLabel.setVisible(true);
1190    }
1191
1192    private void setNineLabel() {
1193        nineLabel.setVisible(true);
1194    }
1195
1196    private void setVBox() {
1197        VBox.setVBox();
1198    }
1199
1200    private void setInputField() {
1201        inputField.setEditable(true);
1202    }
1203
1204    private void setErrorLabel() {
1205        errorLabel.setVisible(true);
1206    }
1207
1208    private void setResultLabel() {
1209        resultLabel.setVisible(true);
1210    }
1211
1212    private void setAddLabel() {
1213        addLabel.setVisible(true);
1214    }
1215
1216    private void setClearLabel() {
1217        clearLabel.setVisible(true);
1218    }
1219
1220    private void setDotLabel() {
1221        dotLabel.setVisible(true);
1222    }
1223
1224    private void setEqualLabel() {
1225        equalLabel.setVisible(true);
1226    }
1227
1228    private void setMinusLabel() {
1229        minusLabel.setVisible(true);
1230    }
1231
1232    private void setMultiplyLabel() {
1233        multiplyLabel.setVisible(true);
1234    }
1235
1236    private void setPlusLabel() {
1237        plusLabel.setVisible(true);
1238    }
1239
1240    private void setZeroLabel() {
1241        zeroLabel.setVisible(true);
1242    }
1243
1244    private void setOneLabel() {
1245        oneLabel.setVisible(true);
1246    }
1247
1248    private void setTwoLabel() {
1249        twoLabel.setVisible(true);
1250    }
1251
1252    private void setThreeLabel() {
1253        threeLabel.setVisible(true);
1254    }
1255
1256    private void setFourLabel() {
1257        fourLabel.setVisible(true);
1258    }
1259
1260    private void setFiveLabel() {
1261        fiveLabel.setVisible(true);
1262    }
1263
1264    private void setSixLabel() {
1265        sixLabel.setVisible(true);
1266    }
1267
1268    private void setSevenLabel() {
1269        sevenLabel.setVisible(true);
1270    }
1271
1272    private void setEightLabel() {
1273        eightLabel.setVisible(true);
1274    }
1275
1276    private void setNineLabel() {
1277        nineLabel.setVisible(true);
1278    }
1279
1280    private void setVBox() {
1281        VBox.setVBox();
1282    }
1283
1284    private void setInputField() {
1285        inputField.setEditable(true);
1286    }
1287
1288    private void setErrorLabel() {
1289        errorLabel.setVisible(true);
1290    }
1291
1292    private void setResultLabel() {
1293        resultLabel.setVisible(true);
1294    }
1295
1296    private void setAddLabel() {
1297        addLabel.setVisible(true);
1298    }
1299
1300    private void setClearLabel() {
1301        clearLabel.setVisible(true);
1302    }
1303
1304    private void setDotLabel() {
1305        dotLabel.setVisible(true);
1306    }
1307
1308    private void setEqualLabel() {
1309        equalLabel.setVisible(true);
1310    }
1311
1312    private void setMinusLabel() {
1313        minusLabel.setVisible(true);
1314    }
1315
1316    private void setMultiplyLabel() {
1317        multiplyLabel.setVisible(true);
1318    }
1319
1320    private void setPlusLabel() {
1321        plusLabel.setVisible(true);
1322    }
1323
1324    private void setZeroLabel() {
1325        zeroLabel.setVisible(true);
1326    }
1327
1328    private void setOneLabel() {
1329        oneLabel.setVisible(true);
1330    }
1331
1332    private void setTwoLabel() {
1333        twoLabel.setVisible(true);
1334    }
1335
1336    private void setThreeLabel() {
1337        threeLabel.setVisible(true);
1338    }
1339
1340    private void setFourLabel() {
1341        fourLabel.setVisible(true);
1342    }
1343
1344    private void setFiveLabel() {
1345        fiveLabel.setVisible(true);
1346    }
1347
1348    private void setSixLabel() {
1349        sixLabel.setVisible(true);
1350    }
1351
1352    private void setSevenLabel() {
1353        sevenLabel.setVisible(true);
1354    }
1355
1356    private void setEightLabel() {
1357        eightLabel.setVisible(true);
1358    }
1359
1360    private void setNineLabel() {
1361        nineLabel.setVisible(true);
1362    }
1363
1364    private void setVBox() {
1365        VBox.setVBox();
1366    }
1367
1368    private void setInputField() {
1369        inputField.setEditable(true);
1370    }
1371
1372    private void setErrorLabel() {
1373        errorLabel.setVisible(true);
1374    }
1375
1376    private void setResultLabel() {
1377        resultLabel.setVisible(true);
1378    }
1379
1380    private void setAddLabel() {
1381        addLabel.setVisible(true);
1382    }
1383
1384    private void setClearLabel() {
1385        clearLabel.setVisible(true);
1386    }
1387
1388    private void setDotLabel() {
1389        dotLabel.setVisible(true);
1390    }
1391
1392    private void setEqualLabel() {
1393        equalLabel.setVisible(true);
1394    }
1395
1396    private void setMinusLabel() {
1397        minusLabel.setVisible(true);
1398    }
1399
1400    private void setMultiplyLabel() {
1401        multiplyLabel.setVisible(true);
1402    }
1403
1404    private void setPlusLabel() {
1405        plusLabel.setVisible(true);
1406    }
1407
1408    private void setZeroLabel() {
1409        zeroLabel.setVisible(true);
1410    }
1411
1412    private void setOneLabel() {
1413        oneLabel.setVisible(true);
1414    }
1415
1416    private void setTwoLabel() {
1417        twoLabel.setVisible(true);
1418    }
1419
1420    private void setThreeLabel() {
1421        threeLabel.setVisible(true);
1422    }
1423
1424    private void setFourLabel() {
1425        fourLabel.setVisible(true);
1426    }
1427
1428    private void setFiveLabel() {
1429        fiveLabel.setVisible(true);
1430    }
1431
1432    private void setSixLabel() {
1433        sixLabel.setVisible(true);
1434    }
1435
1436    private void setSevenLabel() {
1437        sevenLabel.setVisible(true);
1438    }
1439
1440    private void setEightLabel() {
1441        eightLabel.setVisible(true);
1442    }
1443
1444    private void setNineLabel() {
1445        nineLabel.setVisible(true);
1446    }
1447
1448    private void setVBox() {
1449        VBox.setVBox();
1450    }
1451
1452    private void setInputField() {
1453        inputField.setEditable(true);
1454    }
1455
1456    private void setErrorLabel() {
1457        errorLabel.setVisible(true);
1458    }
1459
1460    private void setResultLabel() {
1461        resultLabel.setVisible(true);
1462    }
1463
1464    private void setAddLabel() {
1465        addLabel.setVisible(true);
1466    }
1467
1468    private void setClearLabel() {
1469        clearLabel.setVisible(true);
1470    }
1471
1472    private void setDotLabel() {
1473        dotLabel.setVisible(true);
1474    }
1475
1476    private void setEqualLabel() {
1477        equalLabel.setVisible(true);
1478    }
1479
1480    private void setMinusLabel() {
1481        minusLabel.setVisible(true);
1482    }
1483
1484    private void setMultiplyLabel() {
1485        multiplyLabel.setVisible(true);
1486    }
1487
1488    private void setPlusLabel() {
1489        plusLabel.setVisible(true);
1490    }
1491
1492    private void setZeroLabel() {
1493        zeroLabel.setVisible(true);
1494    }
1495
1496    private void setOneLabel() {
1497        oneLabel.setVisible(true);
1498    }
1499
1500    private void setTwoLabel() {
1501        twoLabel.setVisible(true);
1502    }
1503
1504    private void setThreeLabel() {
1505        threeLabel.setVisible(true);
1506    }
1507
1508    private void setFourLabel() {
1509        fourLabel.setVisible(true);
1510    }
1511
1512    private void setFiveLabel() {
1513        fiveLabel.setVisible(true);
1514    }
1515
1516    private void setSixLabel() {
1517        sixLabel.setVisible(true);
1518    }
1519
1520    private void setSevenLabel() {
1521        sevenLabel.setVisible(true);
1522    }
1523
1524    private void setEightLabel() {
1525        eightLabel.setVisible(true);
1526    }
1527
1528    private void setNineLabel() {
1529        nineLabel.setVisible(true);
1530    }
1531
1532    private void setVBox() {
1533        VBox.setVBox();
1534    }
1535
1536    private void setInputField() {
1537        inputField.setEditable(true);
1538    }
1539
1540    private void setErrorLabel() {
1541        errorLabel.setVisible(true);
1542    }
1543
1544    private void setResultLabel() {
1545        resultLabel.setVisible(true);
1546    }
1547
1548    private void setAddLabel() {
1549        addLabel.setVisible(true);
1550    }
1551
1552    private void setClearLabel() {
1553        clearLabel.setVisible(true);
1554    }
1555
1556    private void setDotLabel() {
1557        dotLabel.setVisible(true);
1558    }
1559
1560    private void set
```

Coding Conventions

Lessons Learned

- Coding conventions ([Programmierrichtlinien](#))
- Formatting, naming, comments, and documentation
- Further Reading: [Google's Java Style Guide](#)

Practice

- See [Moodle](#)
- Optional: inspect [changes of the live coding](#)
- The [resulting code](#) does not contain any comments. Give an example comment.
- Argue for one other comment whether it is useful or not?

Lecture Contents

1. Programming Languages
2. Coding Conventions
3. Tools and Environments
 - Computer-Aided Software Engineering
 - Overview on Development Tools
 - Excursion: Extending Thomas' Calculator
 - Lessons Learned

Computer-Aided Software Engineering

Terms

[adapted from Ghezzi/Jazayeri/Mandrioli]

A **tool** is an application that supports a particular activity. An **environment** is a collection of related tools. Tools and environments aim at automating some of the activities that are involved in software engineering. The generic term for this field of study is **computer-aided software engineering**.

nano? REAL
PROGRAMMERS
USE emacs



HEY. REAL
PROGRAMMERS
USE vim.



WELL, REAL
PROGRAMMERS
USE ed.



NO, REAL
PROGRAMMERS
USE cat.



REAL PROGRAMMERS
USE A MAGNETIZED
NEEDLE AND A
STEADY HAND.



EXCUSE ME, BUT
REAL PROGRAMMERS
USE BUTTERFLIES.



THEY OPEN THEIR
HANDS AND LET THE
DELICATE WINGS FLAP ONCE.

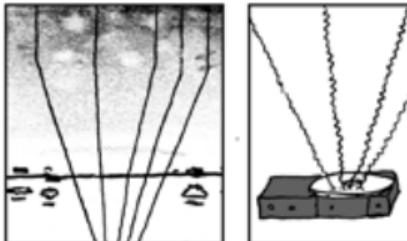


THE DISTURBANCE RIPPLES
OUTWARD, CHANGING THE FLOW
OF THE EDDY CURRENTS
IN THE UPPER ATMOSPHERE.



THESE CAUSE MOMENTARY POCKETS
OF HIGHER-PRESSURE AIR TO FORM,

WHICH ACT AS LENSES THAT
DEFLECT INCOMING COSMIC
RAYS, FOCUSING THEM TO
STRIKE THE DRIVE PLATTER
AND FLIP THE DESIRED BIT.



NICE.
'COURSE, THERE'S AN EMACS
COMMAND TO DO THAT.
OH YEAH! GOOD OL'
C-x M-c M-butterfly...



DAMMIT, EMACS.

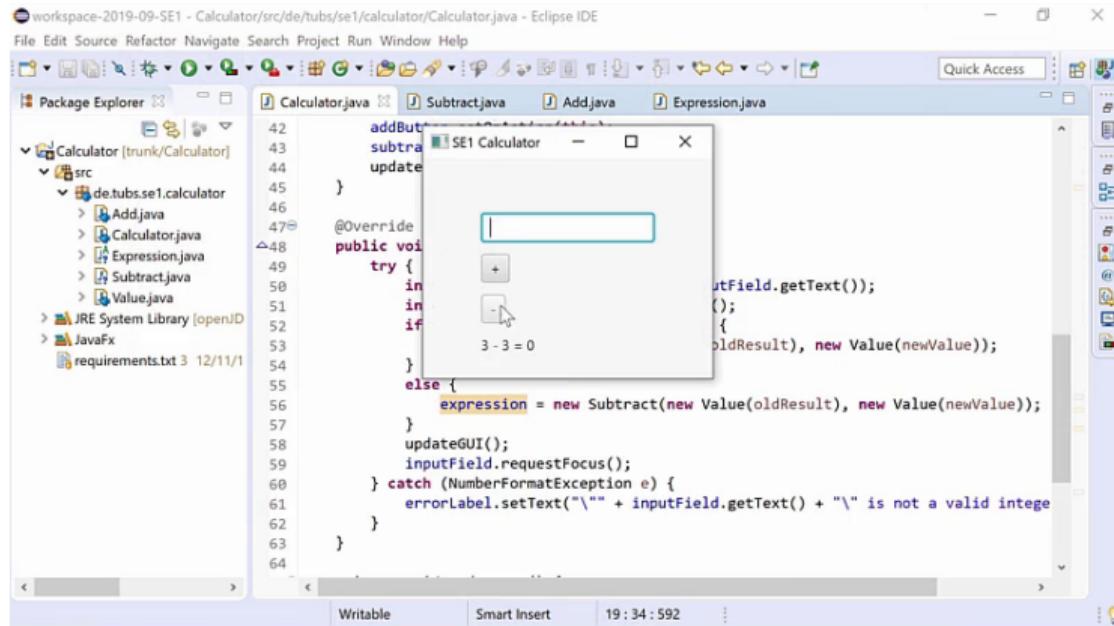
Overview on Development Tools

Variety of Tools

[Ghezzi/Jazayeri/Mandrioli]

- text(ual) editors: emacs, vim, ed, Word, ...
- graphical editors: UML editors, Powerpoint, ...
- assembler, compiler, interpreter
- configuration management tools: git, SVN, CVS, ...
- tracking tools (issue trackers): Github, Gitlab, ...
- tools for code navigation and refactoring
- tools for test specification, generation, execution, reporting
- tools for static and dynamic code analysis (e.g., debugger), reverse/reengineering, project management
- integrated development environments (IDEs): Eclipse, IntelliJ, Android Studio, Visual Studio

Excursion: Extending Thomas' Calculator



Tools and Environments

Lessons Learned

- Tool supports by means of tools, environments, and IDEs
- Further Reading: [Ghezzi/Jazayeri/Mandrioli](#), Chapter 9 Software Engineering Tools and Environments

Practice

- See [Moodle](#)
- Choose a tool or an IDE.
- Tryout tooling that you have learned about in this video but never used before (e.g., automated code formatting or code navigation).
- Post a screenshot and brief description in Moodle.