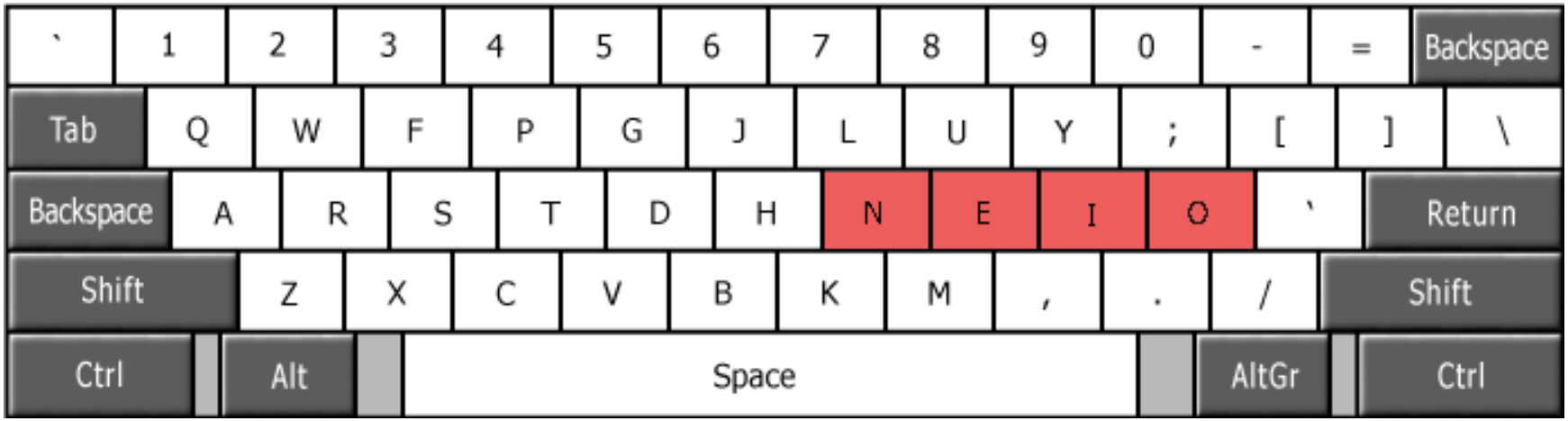
# The design and implementation of a userfriendly object-oriented markup language: Neio

Titouan Vervack
Promotor: Prof. dr. ir. Marko van Dooren
Counsellor: Dr. ir. Benoit Desouter

# Neio


` 1 2 3 4 5 6 7 8 9 0 - = Backspace
Tab Q W F P G J L U Y ; [ ] \
Backspace A R S T D H N E I O ' Return
Shift Z X C V B K M , . / Shift
Ctrl Alt Space AltGr Ctrl

# What is it

- Markup language
- Improves on LaTeX and Markdown

# The good and bad: Markdown

- Not extensible
- Too simple
- No programming model

+ Easy to read/write
+ Very easy to get started

# The good and bad: LaTeX

- – Steep initial learning curve
- – Overly complex
- – Not statically typed

- + Well suited for complex documents
- + Has a programming model
- + Lots of packages

# The good and bad: WYSIWYG

- Corruptable fileformat
- Not well suited for very complex documents
- No full control through programming model

+ Very well known
+ Easy to use for anyone
+ Immediately see the final document

# Goals

- Easy to get started with
- Has to use a modern programming model
- As simple as Markdown
- As powerful as TeX

# Neio document

```
// This is a Neio document
[Document]

/* Below we define a chapter
 * and a paragraph
 */
# Chapter 1
This is the first paragraph
```

# 1 Chapter 1

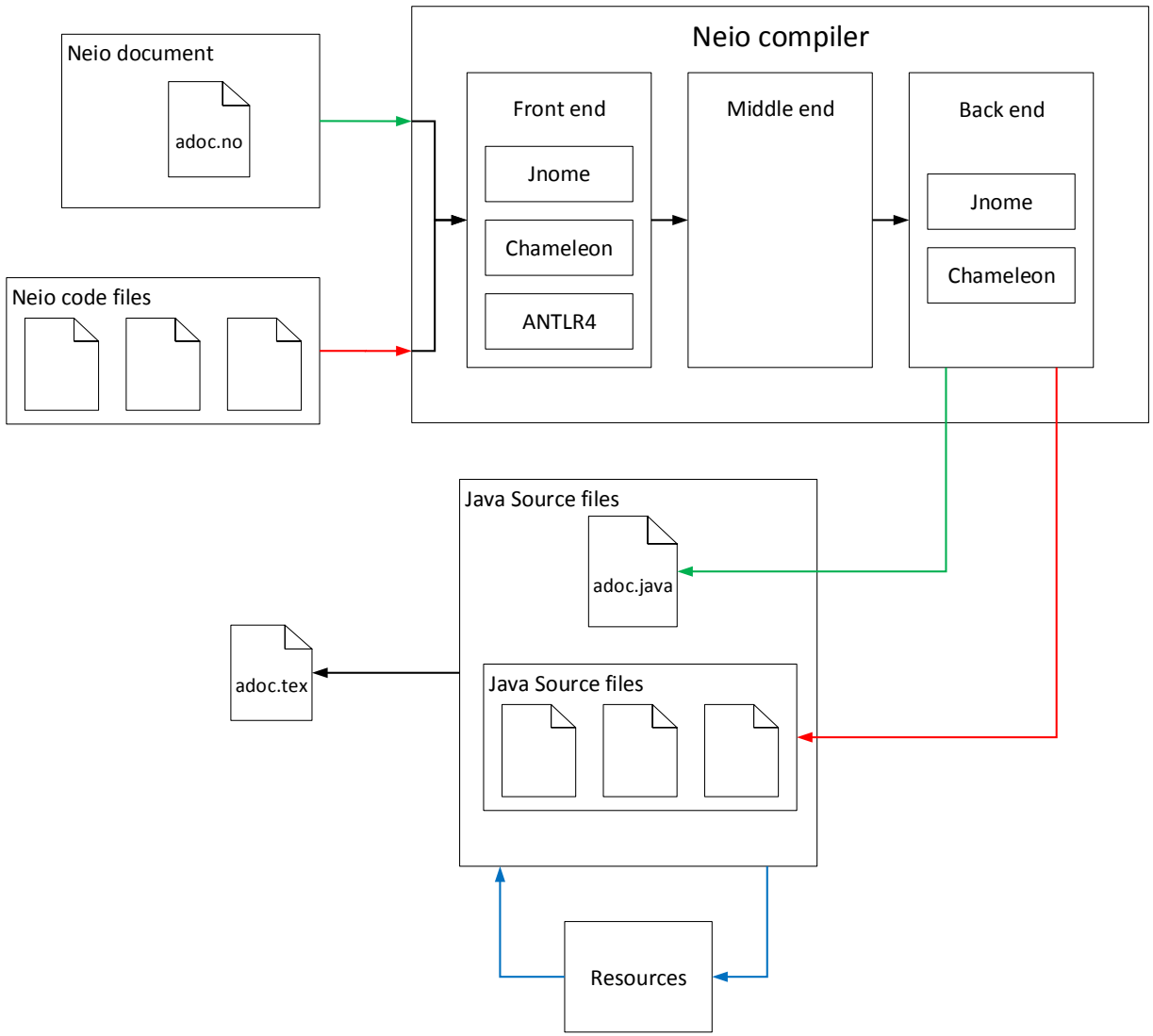
This is the first paragraph

# Neio code file

```
1  namespace neio.stdlib;
2
3  import neio.lang.*;
4
5  class Paragraph extends Content;
6
7  // Private
8  Text text;
9
10 Paragraph(Text text) {
11     this.text = text;
12 }
13
14 // Public
15 Paragraph appendLine(Text t) {
16     return appendText(new Text("\n").appendText(t));
17 }
18
19 ParNLHandler newline() {
20     return new ParNLHandler(this);
21 }
22
23 String toTex() {
24     if (text != null) {
25         return "\par " + text.toTex();
26     } else {
27         return "";
28     }
29 }
```

Fully compatible with Java

# Process flow



# Symbol methods

```
1 Chapter #(Text title) {
2     Chapter chapter = new Chapter(title, 1);
3     addContent(chapter);
4
5     return chapter;
6 }
```

```
1 ParNLHandler newline() {
2     return new ParNLHandler(this);
3 }
```

```
1 Paragraph text(Text text) {
2     new Paragraph(text) par;
3     parent().addContent(par);
4     return par;
5 }
```

Usable symbols: #, -, *, _, $, |, =, ^, ‘, **text** and **newline**

# Call chain

```
[Document]

# Chapter 1
This is the first paragraph.
This is the second line.

This is a second paragraph.
```

```
new Document()
    .newline()
    .newline()
    .#("Chapter 1")
        .newline
        .text("This is the first paragraph.");
        .newline()
        .text("This is the second line.")
        .newline()
        .newline()
        .text("This is a second paragraph.")
```

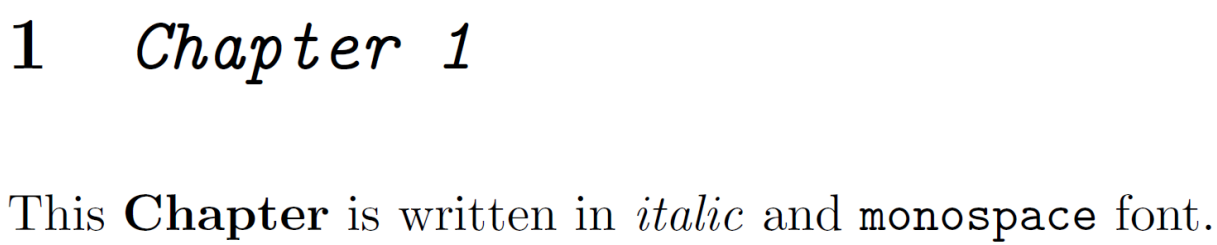
# Surround methods

```
public surround Text *(Text t) {
    return appendText(new BoldText(t));
}

public surround Text _(Text t) {
    return appendText(new ItalicText(t));
}

public surround Text `(Text t) {
    return appendText(new MonospaceText(t));
}
```

```
[Document]
# `_Chapter 1_`
This *Chapter* is written in _italic_ and `monospace` font.
```



# Nested methods

```
1 nested Chapter #(Text title, Integer level) {
2     Chapter chapter = new Chapter(title, level);
3     addContent(chapter);
4
5     return chapter;
6 }
```

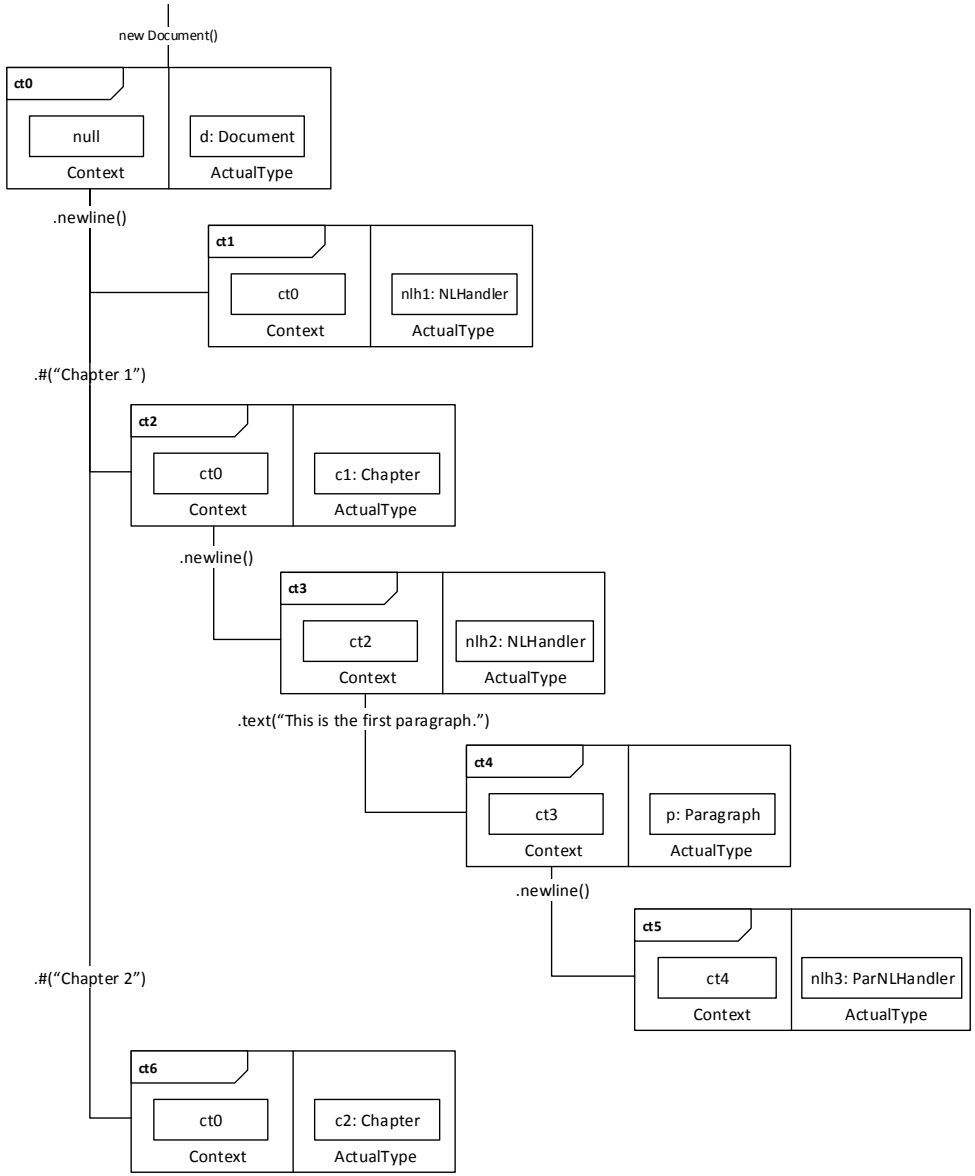
```
1 [Document]
2 # Chapter 1
3 ## Chapter 1.1
4 # Chapter 2
```

# 1 Chapter 1

## 1.1 Chapter 1.1

# 2 Chapter 2

# Context types



# Code blocks

Listing 1: Neio document

```
[Document]
# Chapter 1
{
    Chapter chapter2 = new Chapter("Chapter 2")
}
```

Listing 2: Call chain

```
new Document()
    .#("Chapter 1");
Chapter chapter2 = new Chapter("Chapter 2");
```

# Code blocks

```
[Document]
{
    new Itemize()
}
* Item 1
* Item 2
* Item 3
```

```
[Document]
{
    Itemize itemize = new Itemize();
    return itemize;
}
* Item 1
* Item 2
* Item 3
```

# This

Listing 3: Neio document

```
[Document]
{
    new Itemize()
}
* Item 1
* Item 2
* Item 3
```

Listing 4: Call chain

```
new Document();
Itemize itemize = new Itemize();
this.appendContent(itemize);
// itemize is now the new "this"
this.*("Item 1")
    .*("Item 2")
    .*("Item 3")
```

Listing 5: Java code

```
Document $var0 = new Document();

Itemize $var1 = new Itemize();
Itemize $var2 = $var0.appendContent($var1);

ItemizeItem $var3 = $var2.minus("Item 1");
ItemizeItem $var4 = $var2.minus("Item 2");
ItemizeItem $var5 = $var2.minus("Item 3");
```

# Code blocks: scoped

```
[Document]
# Chapter 1
{{
    List<String> l = new ArrayList<String>();
    l.add("upquote");
    l.add("pdfpages");
    l.add("url");
    for (int i = 0; i < l.size(); i = i + 1) {
        addPackage(l.get(i));
    }
}}
```

```
new Document()
    .#("Chapter 1");
{
    List<String> l = new ArrayList<String>();
    l.add("upquote");
    l.add("pdfpages");
    l.add("pdfpages");
    for (int i = 0; i < l.size(); i = i + 1) {
        this.addPackage(l.get(i));
    }
}
```

# Inline code

```
[Document]
{
    Text addressee = "Thomas Vanhaskel";
    Text help = "my math class";
    Text helpSubject = "math test";
    Text closings = "Kind regards,";
    Text name = "Titouan Vervack";
}

Dear {addressee},

Thank you for helping me out with {help}.

I was able to do great at the {
    helpSubject} thanks to you.

{closings}

{name}
```

Dear Thomas Vanhaskel,

Thank you for helping me out with my math class.

I was able to do great at the math test thanks to you.

Kind regards,

Titouan Vervack

# Table

| Student club | Rounds | Seconds/Round | Dist (km) |
|---|---|---|---|
| HILOK | 1030 | 42 | 298,70 |
| VTK | 1028 | 42 | 298.12 |
| VLK | 841 | 51 | 243.89 |
| VGK | 810 | 53 | 234.90 |
| HK | 771 | 56 | 223.59 |
| VRG | 764 | 57 | 221.56 |
| VEK | 757 | 57 | 219.53 |
| VPPK | 689 | 63 | 199.81 |
| SK | 647 | 67 | 187.63 |
| Zeus WPI | 567 | 76 | 164.43 |
| VBK | 344 | 126 | 99.76 |

# Table

| Student club | Rounds | Seconds/Round | Dist (km) |
|---|---|---|---|
| HILOK | 1030 | 42 | 298,70 |
| VTK | 1028 | 42 | 298.12 |
| VLK | 841 | 51 | 243.89 |
| VGK | 810 | 53 | 234.90 |
| HK | 771 | 56 | 223.59 |
| VRG | 764 | 57 | 221.56 |
| VEK | 757 | 57 | 219.53 |
| VPPK | 689 | 63 | 199.81 |
| SK | 647 | 67 | 187.63 |
| Zeus WPI | 567 | 76 | 164.43 |
| VBK | 344 | 126 | 99.76 |

# Red black tree

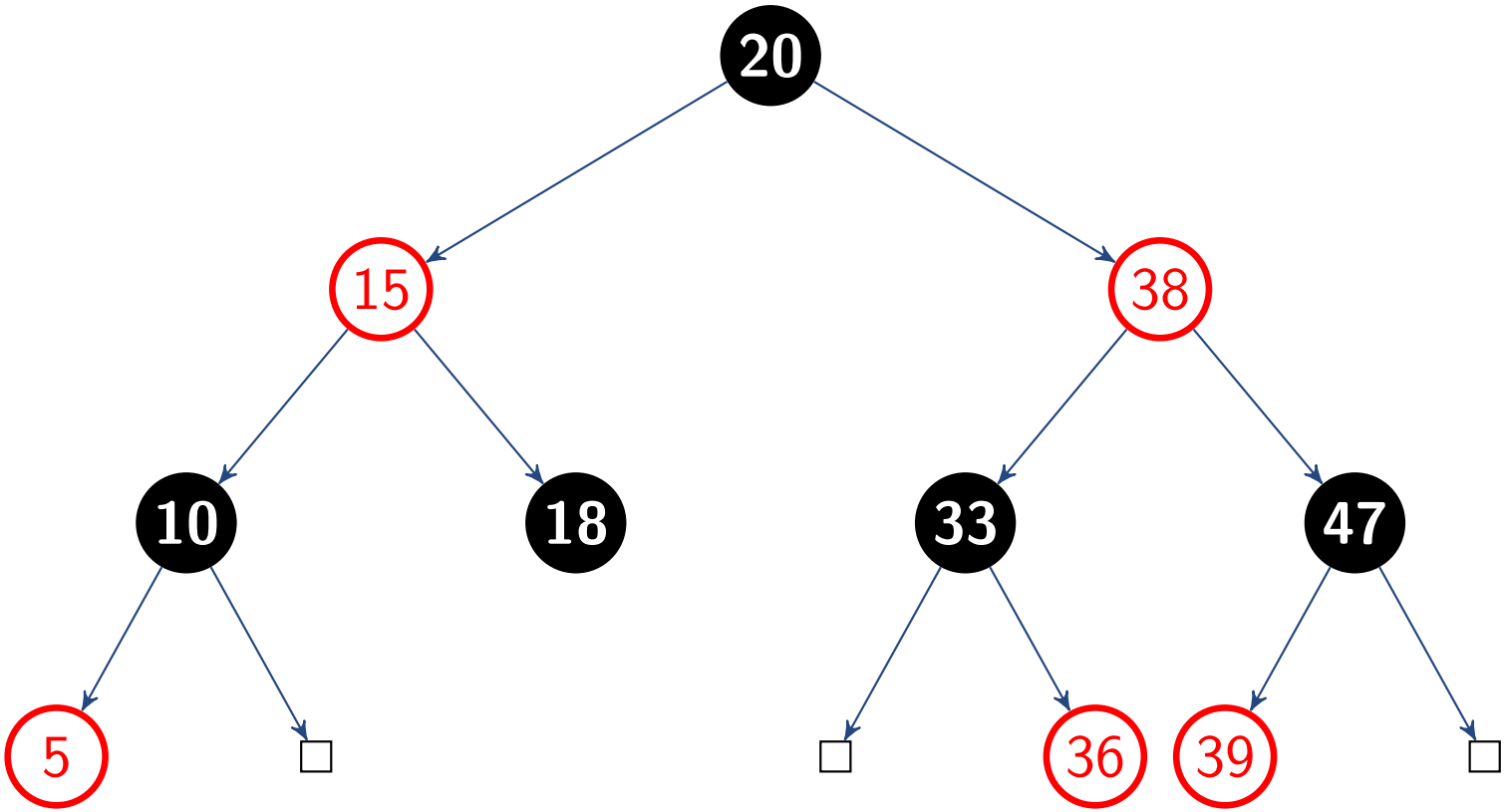
```
[Document]
{
    List<Integer> tree = new ArrayList<Integer>();
    tree.add(33);
    tree.add(15);
    tree.add(10);
    tree.add(5);
    tree.add(20);
    tree.add(18);
    tree.add(47);
    tree.add(38);
    tree.add(36);
    tree.add(39);

    String numbers = String.valueOf(tree.get(0));
    for (int i = 1; i < tree.size(); i = i + 1) {
        numbers = numbers + ''', ''' + tree.get(i);
    }
}

Given the red black tree of {numbers}
{
    RedBlackTree rbt = new RedBlackTree().insert(tree);
    return rbt;
}
```

# Red black tree

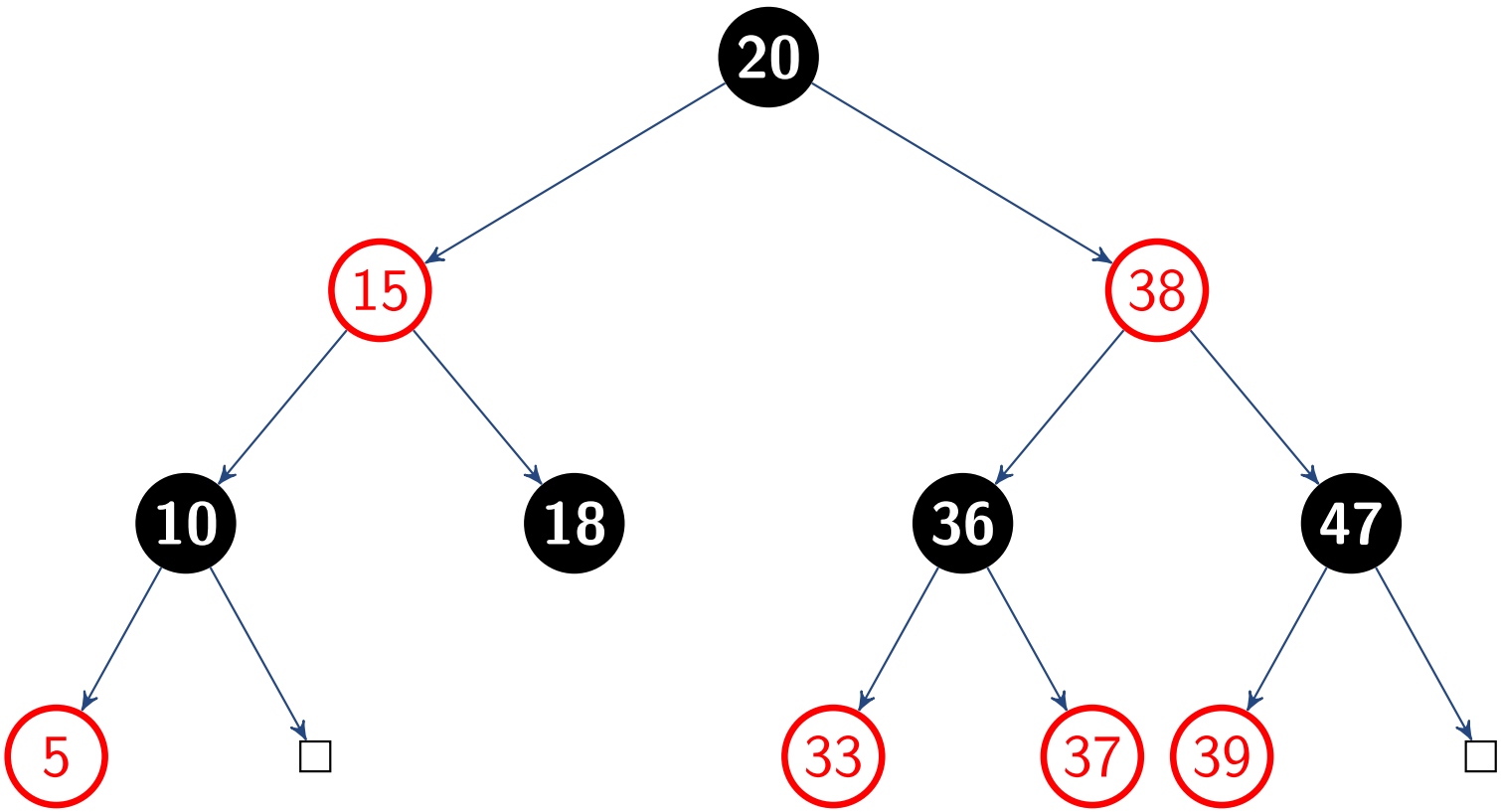
Given the red black tree of 33, 15, 10, 5, 20, 18, 47, 38, 36, 39



# Red black tree

```
1 Add 37
2 {
3     RedBlackTree rbt2 = rbt.insert(37);
4     return rbt2;
5 }
```

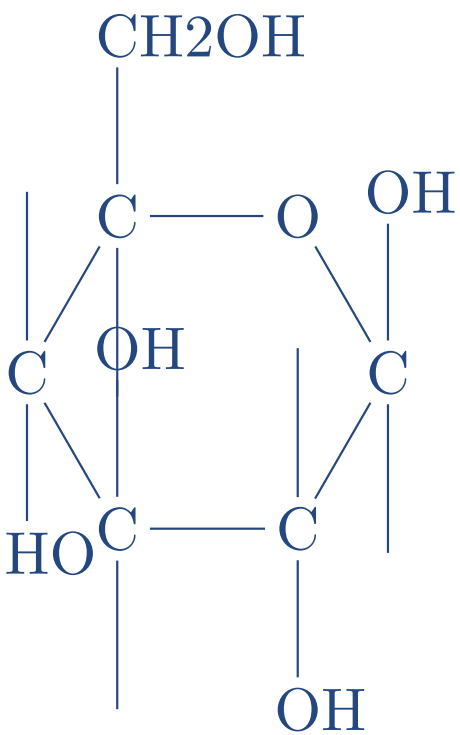
## Add 37



# Structural formulas

```
[Document]
{
    new Structure()
}

$ C*6_OH^-C_^OH-O-C_^CH2OH-C_HO^-C_^OH-
| Glucose
```



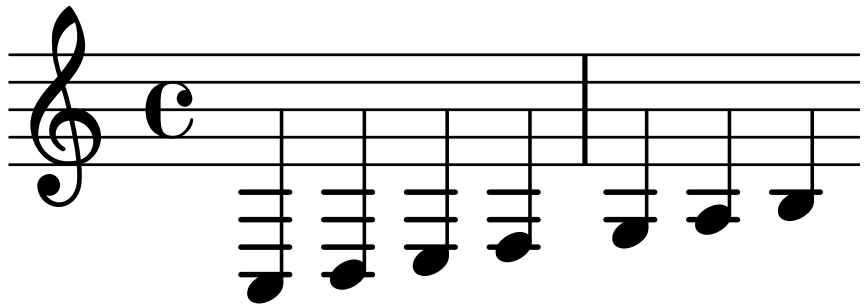

Glucose

# Musical scores

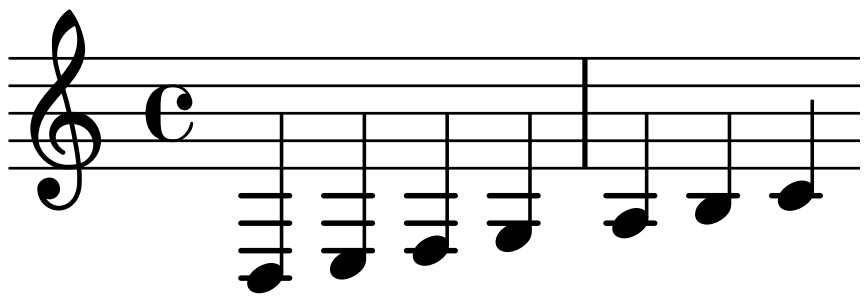
```
[Document]
{
    Score ss = new Score().c().d().e().f().g().a().b();
    return ss;
}

This score is read as {ss.print()}.
{
    Score sss = ss.shift(1);
    return sss;
}
```

# Musical scores



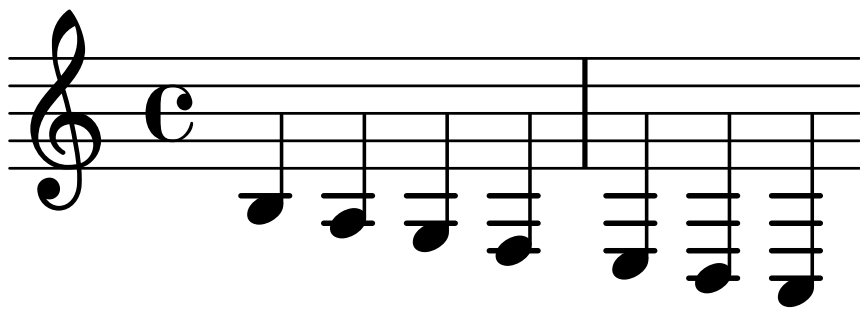
This score is read as c, d, e, f, g, a, b.



We can shift every note by one, doing so yields the following notes: d, e, f, g, a, b, c.

# Musical scores

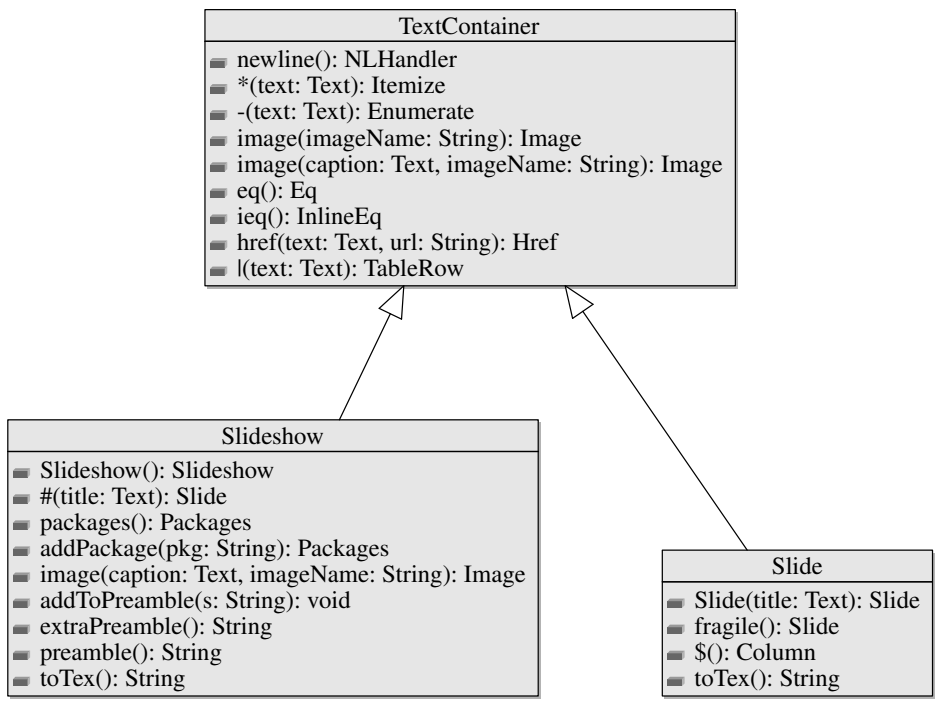
```
1 {
2     Score s4 = ss.reverse();
3     return s4;
4 }
5
6 Reversing the score results in the following sequence of nodes: {s4.print()}
```



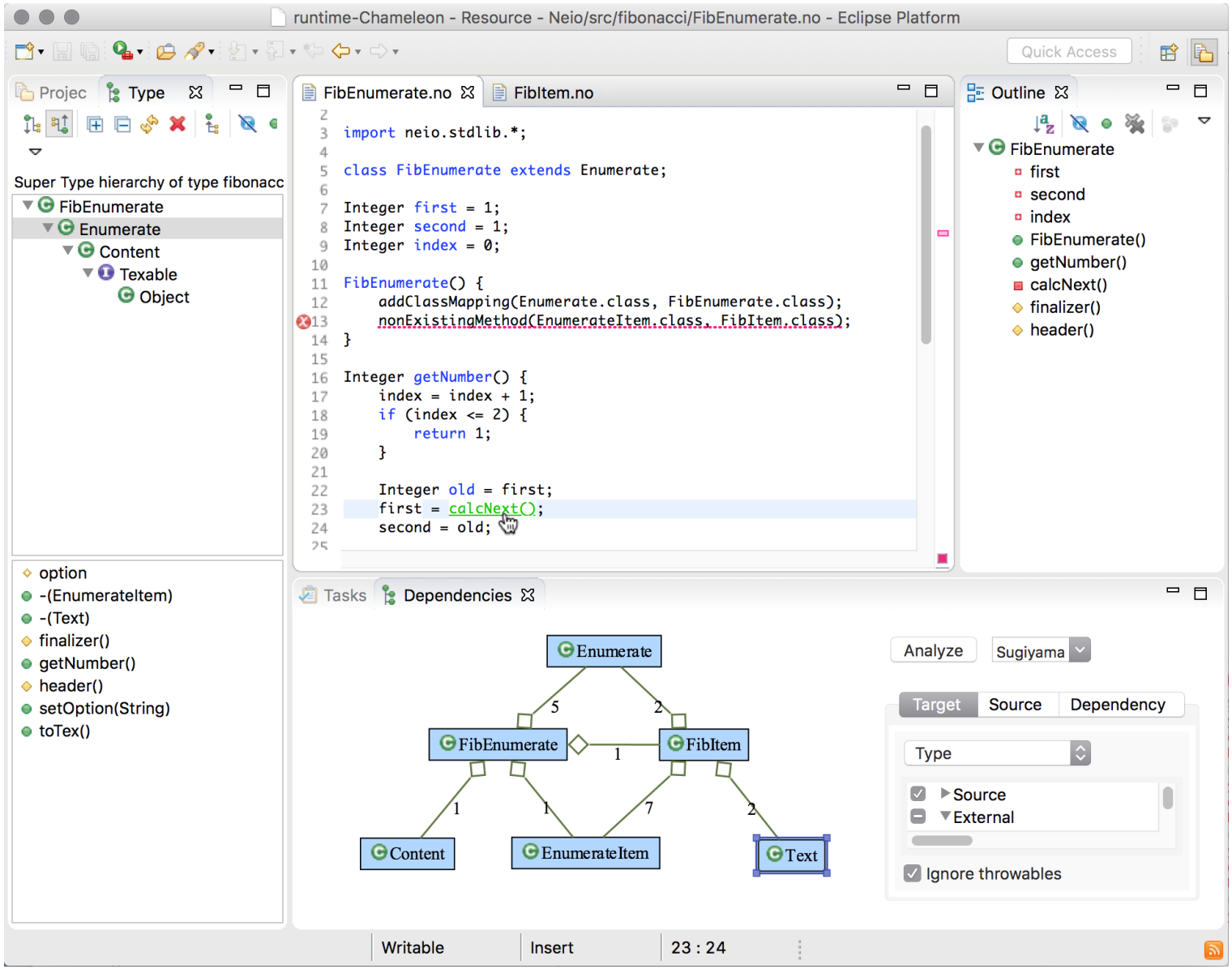
Reversing the score results in the following sequence of nodes: b, a, g, f, e, d, c

# UML diagram

```
1 [Document]
2 {
3     List<String> list = new ArrayList<String>();
4     list.add('''neio.stdlib.Slideshow''');
5     list.add('''neio.stdlib.Slide''');
6     list.add('''neio.stdlib.TextContainer''');
7     new Uml('''project.xml''', '''neio.stdlib''').scale(20).show(list).showAll();
8 }
```



# IDE


runtime-Chameleon - Resource - Neio/src/fibonacci/FibEnumerate.no - Eclipse Platform
Quick Access
Projec
Type
Super Type hierarchy of type fibonacc
FibEnumerate
Enumerate
Content
Texable
Object
option
-(EnumerateItem)
-(Text)
finalizer()
getNumber()
header()
setOption(String)
toTex()
FibEnumerate.no
FibItem.no
import neio.stdlib.*;
class FibEnumerate extends Enumerate;
Integer first = 1;
Integer second = 1;
Integer index = 0;
FibEnumerate() {
addClassMapping(Enumerate.class, FibEnumerate.class);
nonExistingMethod(EnumerateItem.class, FibItem.class);
}
Integer getNumber() {
index = index + 1;
if (index <= 2) {
return 1;
}
Integer old = first;
first = calcNext();
second = old;
Outline
FibEnumerate
first
second
index
FibEnumerate()
getNumber()
calcNext()
finalizer()
header()
Tasks
Dependencies
Enumerate
FibEnumerate
FibItem
Content
EnumerateItem
Text
5
2
1
1
1
7
2
Analyze
Sugiyama
Target
Source
Dependency
Type
Source
External
Ignore throwables
Writable
Insert
23 : 24

# Future work

- Checks on static typing
- Use
- Compiler improvement
- Better error messages

# Questions