Exercises week 3: Advanced Templates I - Revision

Klaas Isaac Bijlsma s2394480
David Vroom s2309939

March 7, 2018

Exercise 18

Learn to design a small template meta program

In the first attempt we used an extra struct to set the initial values of the template parameters that are necessary to do the computations. We solved this by adding default template parameters to the template header of `Bin`. We also noticed that we did not include the `cstddef` headerfile in `bin.h`. The program compiled properly, because of the `iostream` headerfile in `main`, but of course we cannot assume this. Therefore the `cstddef` headerfile is now included.

We used the following code,

bin.h

```
1  #ifndef INCLUDED_BIN_H_
2  #define INCLUDED_BIN_H_
3
4  #include <cstddef>
5
6  template <size_t Decimal,
7            size_t accBin = 0,
8            size_t accDec = 0,
9            size_t idx = 0,
10           size_t tens = 1,
11           size_t twos = 1>
12 struct Bin
```

```
13 {
14     enum { value = Bin<Decimal,
15                        accBin + ((Decimal >> idx) & 1) * tens,
16                        accDec + ((Decimal >> idx) & 1) * twos,
17                        idx + 1,
18                        tens * 10,
19                        twos * 2>::value
20          };
21 };
22
23 template <size_t Decimal,
24           size_t accBin,
25           size_t idx,
26           size_t tens,
27           size_t twos>
28 struct Bin<Decimal, accBin, Decimal, idx, tens, twos>
29 {
30     enum { value = accBin + ((Decimal >> idx) & 1) * tens };
31 };
32
33 #endif
```

main.cc

```
1  #include <iostream>
2  #include "bin.h"
3
4  using namespace std;
5
6  int main()
7  {
8      cout << Bin<5>::value  << '\n'
9           << Bin<27>::value << '\n';
10 }
```