

```
1 // Opgave 1A
2 // Fokus - linie med template og template parameter, korrekt brug af MyType
3 template <typename MyType>
4 void isInRange(const MyType &min, const MyType &max, const MyType &value)
5 {
6     // Fx. denne måde, men opdeling i mindre kodebidder og brug af variable er også tilladt
7     cout << value << " er" << (min <= value && value <= max ? "" : " ikke") << " mellem " << min << " og " << max;
8 }
9
10
11 // Opgave 1B
12 // Der behøves ingen særlig type angivelse, template funktionen bliver automatisk valgt ud fra værdierne
13 int main()
14 {
15     isInRange(0, 10, 5);
16     cout << endl;
17
18     isInRange(0.0, 10.0, 13.5);
19     cout << endl;
20
21     isInRange('a', 'z', 'c');
22     cout << endl;
23 }
24
25 // Opgave 2A Shape.h - fokus: protected: og pure virtual
26 class Shape
27 {
28 public:
29     Shape(int x, int y);
30     void move(int x, int y);
31     virtual void draw() const = 0;
32     virtual int area() const = 0;
33
34 protected:
35     int centerX_;
36     int centerY_;
37
38 private:
39 };
40
41 // Opgave 2A Shape.cpp - fokus, ingen implementering af pure virtual metoder
42
43 #include "Shape.h"
44
45 Shape::Shape(int x, int y)
46 {
47     centerX_ = x;
48     centerY_ = y;
49 }
50
51 void Shape::move(int x, int y)
52 {
53     centerX_ = x;
54     centerY_ = y;
55 }
56
57 // Opgave 2B Rectangle.h - fokus public nedrivning
58 #include "Shape.h"
59
60 class Rectangle : public Shape
61 {
```

```
62 public:
63     Rectangle(int x, int y, int vert, int hori);
64     void setSides(int vert, int hori);
65     void draw() const override;
66     virtual int area() const;
67
68
69 private:
70     int sideVertical_;
71     int sideHorizontal_;
72 };
73
74 // Opgave 2B Rectangle.cpp - fokus, kald af Shape constructor, validering af sides
75
76 #include "Rectangle.h"
77 #include <iostream>
78
79 using namespace std;
80
81 Rectangle::Rectangle(int x, int y, int vert, int hori)
82     : Shape(x, y)
83 {
84     setSides(vert, hori);
85 }
86
87 void Rectangle::setSides(int vert, int hori)
88 {
89     sideVertical_ = (vert > 0 ? vert : 1);
90     sideHorizontal_ = (hori > 0 ? hori : 1);
91 }
92
93 void Rectangle::draw() const
94 {
95     cout << "Rectangle med centrum i (" << centerX_ << ", " << centerY_ << ")" <<      ↗
96         endl;
97     cout << "lodret side " << sideVertical_ << " og vandret side " << sideHorizontal_ ↗
98         << endl;
99     cout << "og arealet " << area() << endl;
100 }
101
102 int Rectangle::area() const
103 {
104     return sideHorizontal_ * sideVertical_;
105 }
106
107 // Opgave 2 - Test.cpp
108 #include "Shape.h"
109 #include "Rectangle.h"
110 #include <iostream>
111 #include <vector>
112
113 using namespace std;
114
115 int main()
116 {
117     // Opgave 2A
118     // Shape s1(4, 5);
119
120     // Opgave 2B+C
121     Rectangle r1(4, 5, 10, 15);
122
123     r1.draw();
124 }
```

```
124     r1.move(7, 8);
125     r1.setSides(11, 16);
126     r1.draw();
127
128     // Opgave 2D - fokus vector med Shape *, iteratorer
129     vector<Shape *> shapeVec;
130
131     shapeVec.push_back(new Rectangle(2, 3, 4, 5));
132     shapeVec.push_back(new Rectangle(10, 11, 14, 15));
133     shapeVec.push_back(new Rectangle(22, 23, 24, 25));
134
135     for (vector<Shape *>::iterator i = shapeVec.begin(); i != shapeVec.end(); i++)
136     {
137         (*i)->draw();
138     }
139
140     // Med auto og en anden dereference
141     for (auto i = shapeVec.begin(); i != shapeVec.end(); i++)
142     {
143         (**i).draw();
144     }
145 }
146
147 // Opgave 3A+C BitSet.h
148 class BitSet
149 {
150 public:
151     BitSet(int size);
152
153     // Flot, hvis de har destructor med under 3A
154     ~BitSet();
155
156     BitSet & shiftIn(int data);
157
158     void print() const;
159
160     // Ekstra metoder til rule-of-three Opgave 3C + destructor
161     BitSet(const BitSet &copyMe); // Copy constructor
162     const BitSet & operator=(const BitSet &copyMe); // Copy Assignment
163
164 private:
165     int *dataPtr_;
166     int size_;
167 };
168
169 // Opgave 3A+C BitSet.cpp
170 #include "BitSet.h"
171 #include <iostream>
172
173 using namespace std;
174
175 BitSet::BitSet(int size) : size_(size > 0 ? size : 32)
176 {
177     // Denne metoder er også tilladt i stedet for memberinitializer
178     size_ = size > 0 ? size : 32;
179
180     dataPtr_ = new int[size_];
181     for (int i = 0; i < size_; i++)
182     {
183         dataPtr_[i] = 0;
184     }
185
186     // Alternativ initialisering tilladt - kun for nerder
187 }
```

```
188     dataPtr_ = new int[size_] {0};
189 }
190
191 BitSet & BitSet::shiftIn(int data)
192 {
193     if (data == 0 || data == 1)
194     {
195         for (int i = 0; i < size_ - 1; ++i)
196         {
197             dataPtr_[i] = dataPtr_[i + 1];
198         }
199
200         dataPtr_[size_ - 1] = data;
201     }
202
203     return *this;
204 }
205
206 void BitSet::print() const
207 {
208     for (int i = 0; i < size_; i++)
209     {
210         cout << dataPtr_[i];
211     }
212 }
213
214
215 // Opgave 3A eller 3C
216 BitSet::~BitSet()
217 {
218     delete[] dataPtr_;
219 }
220
221 // Opgave 3C
222 BitSet::BitSet(const BitSet & copyMe) : size_(copyMe.size_)
223 {
224     // Denne assignment er også tilladt i stedet for memberinitializer
225     size_ = copyMe.size_;
226
227     dataPtr_ = new int[size_];
228
229     for (int i = 0; i < size_; i++)
230     {
231         dataPtr_[i] = copyMe.dataPtr_[i];
232     }
233 }
234
235 const BitSet & BitSet::operator=(const BitSet & copyMe)
236 {
237     // Test for self assignment meget vigtig
238     if (this != &copyMe)
239     {
240         // At teste størrelsen for at undgå reallokering er ekstra flot, men ikke obligatorisk
241         if (size_ != copyMe.size_)
242         {
243             delete[] dataPtr_;
244             size_ = copyMe.size_;
245             dataPtr_ = new int[size_];
246         }
247
248         for (int i = 0; i < size_; i++)
249         {
250             dataPtr_[i] = copyMe.dataPtr_[i];
```

```
251     }
252 }
253
254     return *this;
255 }
256
257 // Opgave 3 Test.cpp
258 #include "BitSet.h"
259
260 #include <iostream>
261
262 using namespace std;
263
264 int main()
265 {
266     // Opgave 3A og 3B
267     BitSet b(8);
268
269     b.shiftIn(1).shiftIn(0).shiftIn(1).shiftIn(1);
270
271     b.print();
272     cout << endl;
273
274
275     // Opgave 3C - ikke angivet at de skal lave test
276     BitSet b2(b);
277     BitSet b3(1);
278
279     b3 = b2;
280
281     b2.print();
282     cout << endl;
283
284     b3.print();
285     cout << endl;
286
287     return 0;
288 }
289
```