

```
1  /*-----*/
2  /*Navn: Oliver Malling      Studie nr.: 202005265      Dato: 14-06-2021*/
3  /*-----*/
4
5  /* - - - - - Opgave 1 - - - - - */
6
7  /*1a) - header fil for flyttebil*/
8  #pragma once
9  #include "Flyttekasse.h"
10 #include <list>
11 #include <iostream>
12 using namespace std;
13
14 class Flyttebil
15 {
16 public:
17     Flyttebil(int kapacitet = 10);
18     int beregnAntalFlyttekasser(const Flyttekasse& flyttekasse);
19     void fyldTomFlyttebil(const Flyttekasse& flyttekasse, int antal);
20     void toemFlyttebil();
21     void operator+=(const Flyttekasse& flyttekasseRef);
22
23 private:
24     int kapacitet_;
25     list<const Flyttekasse*> indhold_;
26 };
27
28
29 /*1b) - source kode for flyttebil*/
30 #include "Flyttebil.h"
31
32 Flyttebil::Flyttebil(int kapacitet)
33 {
34     kapacitet_ = (10 <= kapacitet && kapacitet <= 40 ? kapacitet : 10);
35 }
36
37 int Flyttebil::beregnAntalFlyttekasser(const Flyttekasse& flyttekasse)
38 {
39     int space = 0;
40     if (indhold_.empty() != 0)
41     {
42         space = flyttekasse.beregnVolumen() * 1000;
43     }
44     return space;
45 }
46
47 void Flyttebil::fyldTomFlyttebil(const Flyttekasse& flyttekasse, int antal)
48 {
49     if (indhold_.empty() != 0 && antal <= beregnAntalFlyttekasser      ↗
50         (flyttekasse))
51     {
52         indhold_.assign(antal, &flyttekasse);
53     }
54     else
55     {
56         cout << "Flyttebilen er ikke tom eller der er ikke plads til " <<      ↗
```

```
        antal << " flyttekasser. Der fyldes ingen flyttekasser i
        flyttebilen." << endl;
56     cout << endl;
57 }
58 cout << "Der er nu " << indhold_.size() << " flyttekasser i flyttebilen." << endl;
59 cout << endl;
60 }
61
62 void Flyttebil::toemFlyttebil()
63 {
64     indhold_.clear();
65     cout << "Flyttebilen er tom" << endl;
66     cout << endl;
67 }
68
69
70 /*1c) - Test program for flyttebil*/
71 #include "Flyttebil.h"
72
73 int main()
74 {
75     Flyttekasse kasse1(10, 10, 10);
76     Flyttebil bil1(20);
77     cout << "Der er plads til " << bil1.beregnAntalFlyttekasser(kasse1) << "
        flyttekasser i flyttebilen" << endl;
78     bil1.fyldTomFlyttebil(kasse1, 500); //for lavt antal
79     bil1.toemFlyttebil();
80     bil1.fyldTomFlyttebil(kasse1, 2000); //for højt antal
81
82     bil1 += kasse1; //+= operator
83
84     return 0;
85 }
86
87
88 /*1d) - prototype for += operator*/
89 void operator+=(const Flyttekasse& flyttekasseRef);
90
91
92 /*1e) - implementering af += operator*/
93 void Flyttebil::operator+=(const Flyttekasse& flyttekasseRef)
94 {
95     if (indhold_.size() != beregnAntalFlyttekasser(flyttekasseRef))
96     {
97         indhold_.push_back(&flyttekasseRef);
98     }
99     else
100     {
101         cout << "Der er ikke plads til flere flyttekasser i flyttebilen." << endl;
102     }
103     cout << "Der er nu " << indhold_.size() << " flyttekasser i flyttebilen." << endl;
104     cout << endl;
105 }
```

```
106
107
108 /*1f) - test af += operator*/
109 #include "Flyttebil.h"
110
111 int main()
112 {
113     Flyttekasse kasse1(10, 10, 10);
114     Flyttebil bil1(20);
115     cout << "Der er plads til " << bil1.beregnAntalFlyttekasser(kasse1) << "  ↗
         flyttekasser i flyttebilen" << endl;
116     bil1.fyldTomFlyttebil(kasse1, 500);    //for lavt antal
117     bil1.toemFlyttebil();
118     bil1.fyldTomFlyttebil(kasse1, 2000);    //for højt antal
119
120     bil1 += kasse1;                        //+= operator
121
122     return 0;
123 }
124
125
126 /* - - - - - Opgave 2 - - - - - */
127
128 /*2a) - skriv kode for header-fil Feriebolig*/
129 #pragma once
130 #include "Dato.h"
131
132 class Feriebolig
133 {
134 public:
135     Feriebolig(const Dato& købsdato, int købspris, int antalSengepladser);
136     virtual ~Feriebolig();
137     Dato getKøbsdato() const;
138     void setKøbsdato(const Dato& købsdato);
139     int getKøbspris() const;
140     void setKøbspris(int købspris);
141     int getAntalSengepladser() const;
142     void setAntalSengepladser(int antalSengepladser);
143     virtual int beregnAfgift() = 0;
144     virtual void print() const;
145
146 protected:
147     Dato købsdato_;
148
149 private:
150     int købspris_;
151     int antalSengepladser_;
152 };
153
154
155 /*2b) - headerfil Sommerhus*/
156 #pragma once
157 #include "Feriebolig.h"
158 #include <iostream>
159 using namespace std;
160
```

```
161 class Sommerhus : public Feriebolig
162 {
163 public:
164     Sommerhus(const char* navn, int areal, float afstandTilStrand, const Dato&
        købsdato, int købspris, int antalSengepladser);
165     void setAreal(int areal);
166     void setNavn(const char* navn);
167     int beregnAfgift() const;
168     void print() const;
169
170 private:
171     int areal_;
172     float afstandTilStrand_;
173     char* navn_;
174 };
175
176
177 /*2c) - source fil for Sommerhus*/
178 #include "Sommerhus.h"
179
180 Sommerhus::Sommerhus(const char* navn, int areal, float afstandTilStrand,
        const Dato& købsdato, int købspris, int antalSengepladser) : Feriebolig
        (købsdato, købspris, antalSengepladser)
181 {
182     setNavn(navn);
183     setAreal(areal);
184     afstandTilStrand_ = (afstandTilStrand >= 0 ? afstandTilStrand : 0);
185 }
186
187 void Sommerhus::setAreal(int areal)
188 {
189     areal_ = (areal >= 25 ? areal : 25);
190 }
191
192 void Sommerhus::setNavn(const char* navn)
193 {
194     navn_ = new char[strlen(navn) + 1];
195
196     if (navn_ == nullptr)
197     {
198         exit(1);
199     }
200     strcpy(navn_, navn);
201 }
202
203 int Sommerhus::beregnAfgift() const
204 {
205     int afgift = 0;
206     afgift = getKøbspris() / areal_;
207     return afgift * 12;
208 }
209
210 void Sommerhus::print() const
211 {
212     cout << navn_ << endl;
213     cout << areal_ << endl;
```

```
214     cout << "Naermeste strand: " << afstandTilStrand_ << " km" << endl;
215     Feriebolig::print();
216     cout << "Ejendomsskat " << beregnAfgift() << " kr. pr. aar" << endl;
217 }
218
219
220 /*2d) - test program for Sommerhus*/
221 #include "Sommerhus.h"
222
223 Sommerhus::Sommerhus(const char* navn, int areal, float afstandTilStrand,
224     const Dato& købsdato, int købspris, int antalSengepladser) : Feriebolig
225     (købsdato, købspris, antalSengepladser)
226 {
227     setNavn(navn);
228     setAreal(areal);
229     afstandTilStrand_ = (afstandTilStrand >= 0 ? afstandTilStrand : 0);
230 }
231
232 void Sommerhus::setAreal(int areal)
233 {
234     areal_ = (areal >= 25 ? areal : 25);
235 }
236
237 void Sommerhus::setNavn(const char* navn)
238 {
239     navn_ = new char[strlen(navn) + 1];
240     if (navn_ == nullptr)
241     {
242         exit(1);
243     }
244     strcpy(navn_, navn);
245 }
246
247 int Sommerhus::beregnAfgift() const
248 {
249     int afgift = 0;
250     afgift = getKøbspris() / areal_;
251     return afgift * 12;
252 }
253
254 void Sommerhus::print() const
255 {
256     cout << navn_ << endl;
257     cout << areal_ << endl;
258     cout << "Naermeste strand: " << afstandTilStrand_ << " km" << endl;
259     Feriebolig::print();
260     cout << "Ejendomsskat " << beregnAfgift() << " kr. pr. aar" << endl;
261 }
```