

---- Anyag.h ----

```
1  #include <list>
2  #include <vector>
3
4  class Anyag {
5      std::list<Anyag*> szomszed;
6      int x, y;
7  public:
8      typedef std::list<Anyag*>::iterator iterator;
9
10     Anyag(int x, int y, std::vector<std::vector<Anyag*> >& test) :x(x), y(y)
11     {
12         if( x > 0 ) { ujszomszed( test[x-1][y] ); test[x-1][y]->ujszomszed(this); }
13         if( y > 0 ) { ujszomszed( test[x][y-1] ); test[x][y-1]->ujszomszed(this); }
14     }
15
16     void ujszomszed(Anyag* sz) { szomszed.push_back( sz ); }
17     iterator begin() { return szomszed.begin(); }
18     iterator end() { return szomszed.end(); }
19     int szomszedokSzama() const { return szomszed.size(); }
20
21     virtual void calcTemperature()=0;
22     virtual double getTemperature() const = 0;
23     virtual void setTemperature() = 0;
24
25     virtual ~Anyag() {}
26 };
```

---- Fem.h ----

```
27 #include "anyag.h"
28
29 class Fem :public Anyag {
30     double T; // hőmérséklet
31     double Tnew; // következő hőmérséklet
32 public:
33     Fem(double T, int i, int j, std::vector<std::vector<Anyag*> >& test) : Anyag(i, j, test), T(T) {}
34     void calcTemperature();
35     double getTemperature() const;
36     void setTemperature();
37 };
```

---- Fem.cpp ----

```
38 #include "fem.h"
39
40 double Fem::getTemperature() const { return T; }
41 void Fem::setTemperature() { T = Tnew; }
42 void Fem::calcTemperature() {
43     double deltaT=0;
44     iterator first, last=end();
45     for( first = begin(); first!=last; ++first ) deltaT+= (*first)->getTemperature() - T;
46     Tnew = T + 0.2*deltaT;
47 }
```

---- Fa.h ----

```
48 #include "anyag.h"
49
50 class Fa :public Anyag {
51     double T; // hőmérséklet
52     double Tnew; // következő hőmérséklet
53 public:
54     Fa(double T, int i, int j, std::vector<std::vector<Anyag*> >& test) : Anyag(i, j, test), T(T) {}
55     void calcTemperature();
56     double getTemperature() const;
57     void setTemperature();
58 };
```

---- Fa.cpp ----

```
59 #include "fa.h"
60
61 double Fa::getTemperature() const { return T; }
62 void Fa::setTemperature() { T = Tnew; }
63 void Fa::calcTemperature()
64 {
65     double deltaT=0;
66     iterator first, last=end();
67     for( first = begin(); first!=last; ++first )
68         deltaT+= (*first)->getTemperature() - T;
69     Tnew = T + 0.01*deltaT;
70 }
```

``` ----- Futo.h ----- ```

```
71 #include "anyag.h"
72
73 class Futo : public Anyag {
74     const double T;    // hőmérséklet
75 public:
76     Futo(double T, int i, int j, std::vector<std::vector<Anyag*> >& test) : Anyag(i, j, test), T(T) {}
77     void calcTemperature() {};
78     double getTemperature() const { return T; }
79     void setTemperature() {};
80 };
```

``` ----- Test.cpp ----- ```

```
81 #include <iostream>
82 #include <fstream>
83 #include <sstream>
84 #include <iomanip>
85 #include <cstdlib>
86 #include <vector>
87 #include <cmath>
88
89 #include "fem.h"
90 #include "fa.h"
91 #include "futo.h"
92
93 using namespace std;
94
95 class Test {
96     vector<vector<Anyag*> > test;
97 public:
98     Test(size_t, size_t);
99     void hoterjedes();
100    void plot(int);
101 };
102
103 const double R = 15;
104
105 double distance( double y1, double x1, double y2, double x2 ) {
106     return sqrt( (x1-x2)*(x1-x2) + (y1-y2)*(y1-y2) );
107 }
108
109 Test::Test(size_t sor, size_t oszlop) :test(sor, vector<Anyag*>(oszlop)) {
110     for (size_t i = 0; i < sor; i++) {
111         for (size_t j = 0; j < oszlop; j++) {
112             if( 15 <= i && i <=35 && 10<=j && j<=20 ) test[i][j] = new Futo(100.0, i, j, test);
113             else if( distance( i, j, 25, 70 ) <= R ) test[i][j] = new Fa( 5 , i, j, test);
114             else test[i][j] = new Fem( 50 , i, j, test);
115         }
116     }
117 }
118
119 void Test::hoterjedes() {
120     size_t sor = test.size();
121     size_t oszlop = test[0].size();
122
123     for (size_t i = 0; i < sor; i++)
124         for (size_t j = 0; j < oszlop; j++) test[i][j]->calcTemperature();
125
126     for (size_t i = 0; i < sor; i++)
127         for (size_t j = 0; j < oszlop; j++) test[i][j]->setTemperature();
128 }
129
130 void Test::plot(int i) {
131     ostringstream fname;
132     fname << "hoterjedes" << setw(3) << setfill('0') << i << ".pgm";
133     fstream of(fname.str().c_str(), ofstream::out|ofstream::binary);
134     size_t sor = test.size();
135     size_t oszlop = test[0].size();
136
137     of << "P5" << endl << sor << endl << oszlop << endl << 255 << endl;
138     for (size_t i = 0; i < sor; i++) {
139         for (size_t j = 0; j < oszlop; j++) {
140             char b = (int) test[i][j]->getTemperature()*2;
141             of.write(&b, 1);
142         }
143     }
144 }
145
146 int main() {
147     Test test(100, 100);
148     int kep = 0;
149     test.plot(kep++);
150     for (int i = 0; i < 200000; i++) {
151         test.hoterjedes();
152         if (i % 1000 == 0) test.plot(kep++);
153     }
154     return 0;
155 }
156
```