

```

1 // Programação Orientada a Objectos 2020/2021
2 // Ana Rita Videira - 5012012218
3
4 //Exame 1920 - Época Recurso - Exercício 7
5
6
7 #include <cstdlib>
8 #include <iostream>
9 #include <sstream>
10 #include <fstream>
11 #include <string>
12 #include <ctype.h>
13 #include <vector>
14
15 using namespace std;
16
17 class Fruta{
18
19 protected:
20     static int conta;
21     int id;
22     float peso;
23
24 public:
25     Fruta(float pe): peso(pe) {
26         id = conta++;
27     };
28     float getPeso()const { return peso; };
29     int getID()const { return id; };
30     void setPeso( int p ){peso=p; };
31     void setID( int i ){ id = i; };
32     virtual float setCrescer()=0;
33 };
34
35 int Fruta::conta = 1;
36
37
38
39 class Limao: public Fruta{
40
41 public:
42     Limao() : Fruta(150){};
43     float setCrescer(){return peso = peso * 1.15;}
44 };
45
46
47
48 class Laranja: public Fruta {
49
50 public:
51     Laranja(): Fruta(100){};
52     float setCrescer(){return peso = peso * 1.10;}
53 };
54
55 class Arvore {
56     vector <Fruta*> frutas;
57
58 public:
59     Arvore() = default;
60
61     void nascer(Fruta *x) { frutas.push_back(x); }
62
63     void cair(int x) {
64         for (auto i = frutas.begin(); i != frutas.end(); i++)
65             if ((*i)->getID() == x) {
66                 i = frutas.erase(i);
67                 return;
68             }
69     }
70
71     void crescer() {
72         for (int i = 0; i < frutas.size(); i++)
73             frutas[i]->setCrescer();

```

```

74     }
75
76     int getTotal() const {
77         return frutas.size();
78     }
79
80     Fruta* getFruto(int i) const {
81         return frutas[i];
82     }
83
84 };
85
86
87
88 //Mostra o id e peso de todos os frutos da Arvore
89
90 ostream & operator<<(ostream & out, Arvore & a) {
91     for (int i = 0; i < a.getTotal(); i++)
92         out << a.getFruto(i)->getID() << " " << a.getFruto(i)->getPeso() << endl;
93
94     return out;
95 }
96
97
98 int main(int argc, char** argv) {
99
100     Arvore a;
101     a.nascer(new Laranja());
102     a.nascer(new Laranja());
103     a.nascer(new Limao());
104
105     cout << a << endl;
106
107     a.crescer();
108
109     cout << a << endl;
110
111
112     return 0;
113 }
114
115

```