

```
1 // Copyright (c) 2022 Swopblock LLC
2 // See https://github.com/swopblock
3
4 namespace Swopblock.Liquidity
5 {
6     public interface IMain : IBranches, ILiquidity
7     {
8         public decimal CashBalance { get; set; }
9
10        public decimal CashVolume { get; set; }
11    }
12
13    public interface IBranches
14    {
15        public IBranch[] Branches { get; set; }
16    }
17
18    public interface IBranch : IAccounts, ILiquidity
19    {
20        public IMain Supply { get; set; }
21
22        public IMain Demand { get; set; }
23
24        public decimal CashBalance { get; set; }
25
26        public decimal CashVolume { get; set; }
27
28        public decimal AssetBalance { get; set; }
29
30        public decimal AssetVolume { get; set; }
31    }
32
33    public interface IAccounts
34    {
35        public IAccount[] Accounts { get; set; }
36    }
37
38    public interface IAccount : IAddresses, ILiquidity
39    {
40        public IBranch Supply { get; set; }
41
42        public IBranch Demand { get; set; }
43
44        public decimal CashBalance { get; set; }
45
46        public decimal CashVolume { get; set; }
47
48        public decimal AssetBalance { get; set; }
49    }
```

```
50     public decimal AssetVolume { get; set; }
51 }
52
53 public interface IAddresses
54 {
55     public IAddress[] Addresses { get; set; }
56 }
57
58 public interface IAddress : IOutputs, ILiquidity
59 {
60     public IAccount Supply { get; set; }
61
62     public IAccount Demand { get; set; }
63
64     public decimal CashBalance { get; set; }
65
66     public decimal CashVolume { get; set; }
67
68     public decimal AssetBalance { get; set; }
69
70     public decimal AssetVolume { get; set; }
71 }
72
73 public interface IOutputs
74 {
75     public IOutput[] Outputs { get; set; }
76 }
77
78 public interface IOutput : Inputs, ILiquidity
79 {
80     public IAddress Supply { get; set; }
81
82     public IAddress Demand { get; set; }
83
84     public decimal CashBalance { get; set; }
85
86     public decimal CashVolume { get; set; }
87
88     public decimal AssetBalance { get; set; }
89
90     public decimal AssetVolume { get; set; }
91 }
92
93 public interface Inputs
94 {
95     public IInput[] Inputs { get; set; }
96 }
97
98 public interface IInput : IMatches, ILiquidity
```

```
99     {
100         public IOutput Supply { get; set; }
101
102         public IOutput Demand { get; set; }
103
104         public decimal CashBalance { get; set; }
105
106         public decimal CashVolume { get; set; }
107
108         public decimal AssetBalance { get; set; }
109
110         public decimal AssetVolume { get; set; }
111     }
112
113     public interface IMatches
114     {
115         public IMatch[] Matches { get; set; }
116     }
117
118     public interface IMatch : ILiquidity
119     {
120         public IInput Supply { get; set; }
121
122         public IInput Demand { get; set; }
123
124         public decimal CashBalance { get; set; }
125
126         public decimal CashVolume { get; set; }
127
128         public decimal AssetBalance { get; set; }
129
130         public decimal AssetVolume { get; set; }
131
132         public IMatch SwopEqualibrium()
133         {
134             // Swopblock LLC software license to be available.
135
136             return (IMatch)this;
137         }
138
139         public void SwopCustody(IMatch Match)
140         {
141             // Swopblock LLC software license to be available.
142
143             return;
144         }
145     }
146 }
147
```

```
148     public interface ISupply : ILiquidity
149     {
150         public IMatch Supply { get; set; }
151
152         public IMatch Demand { get; set; }
153     }
154
155     public interface IDemand : ILiquidity
156     {
157         public decimal CashBalance { get; set; }
158
159         public decimal CashVolume { get; set; }
160
161         public decimal AssetBalance { get; set; }
162
163         public decimal AssetVolume { get; set; }
164     }
165
166     public interface ILiquidity { }
167 }
168
```