```csharp
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Perceptron
8  {
9      class Program
10     {
11
12
13         class Input
14         {
15             public double x0;
16             public double x1;
17             public double y;
18             public Input(double x0, double x1, double y)
19             {
20                 this.x0 = x0;
21                 this.x1 = x1;
22                 this.y = y;
23             }
24         }
25
26
27
28
29         class Perceptron
30         {
31             public List<double> W;
32             public double Bias;
33             public List<Input> data;
34             public double N;
35             int licznikEpok;
36             string[] output;
37             int accuratness;
38
39
40             public void Run()
41             {
42                 List<Input> inputLearn = new List<Input>();      //Ustawia dane ⇗
                       uczące
43                 inputLearn.Add(new Input(0, 0, 0));
44                 inputLearn.Add(new Input(1, 0, 0));
45                 inputLearn.Add(new Input(0, 1, 0));
46                 inputLearn.Add(new Input(1, 1, 1));
47                 List<Input> inputCheck = new List<Input>();      //Ustawia dane ⇗
                       walidujące
48                 inputCheck.Add(new Input(0, 0, 0));
49                 inputCheck.Add(new Input(1, 0, 0));
50                 inputCheck.Add(new Input(0, 1, 0));
51                 inputCheck.Add(new Input(1, 1, 1));
52
53                 for (int i = 0; i < 1000; i++)                   //Każdy obieg ⇗
                       pętli to kolejna epoka
```

```csharp
54                {
55                    licznikEpok++;
56                    LoadInput(inputLearn);
57                    Learn();
58                    LoadInput(inputCheck);
59                    Evaluate();
60                    if (licznikEpok % 1000 == 0)
61                        Console.WriteLine(licznikEpok);
62                }
63                System.IO.File.WriteAllLines(@"OUTPUT.txt", output);
64                Console.WriteLine("Waga 0: " + W[0]);
65                Console.WriteLine("Waga 1: " + W[1]);
66            }
67
68            private int activateFunction(double x)
69            {
70                if (x < 1)
71                    return 0;
72                return 1;
73            }
74
75            public void LoadInput(List<Input> input)
76            {
77                data = input;
78            }
79
80
81
82            public void Learn()
83            {
84                double sum;
85                foreach (Input element in data)
86                {
87                    sum = 0;
88                    //sum = element.x0 * W[0] + element.x1 * W[1] + Bias;
89                    sum = element.x0 * W[0] + element.x1 * W[1];
90                    int result = activateFunction(sum);
91                    if (element.y == result)
92                    {
93                        //neuron działa poprawnie
94                    }
95                    else
96                    {
97                        //trzeba poprawić wagi
98                        W[0] += N * (element.y - result) * element.x0;
99                        W[1] += N * (element.y - result) * element.x1;
100                        Bias += N * (1 - result) * 1;
101                    }
102
103                }
104            }
105
106
107            public void Evaluate()
108            {
109                accuratness = 0;
```

```csharp
110                    double sum;
111                    foreach (Input element in data)
112                    {
113                        sum = 0;
114                        //sum = element.x0 * W[0] + element.x1 * W[1] + Bias;
115                        sum = element.x0 * W[0] + element.x1 * W[1] ;
116                        double result = activateFunction(sum);
117                        if (element.y == result)
118                        {
119                            //neuron działa poprawnie
120                            accuratness++;
121                        }
122                        else
123                        {
124                            //błąd
125                        }
126
127                    }
128                    output[licznikEpok - 1] = accuratness.ToString();
129                }
130
131            public Perceptron()
132            {
133                W = new List<double>();      //Lista wag
134
135                Random random = new Random();
136                W.Add(random.NextDouble());
137                W.Add(random.NextDouble());
138                Bias = random.NextDouble();
139                N = 0.01;                    //Współczynnik uczenia
140                licznikEpok = 0;
141                output = new string[1000];  //rozmiar tabicy na dane wyjściowe
142            }
143
144        }
145
146
147
148        static void Main(string[] args)
149        {
150            Perceptron p1 = new Perceptron();
151            p1.Run();
152
153            Console.WriteLine("Press any key");
154            Console.ReadLine();
155        }
156    }
157 }
158
```