

```
1 public class MyMath {
2
3     /*
4     מקבל: שני מספרים שלמים
5     מחזיר: סכום שני המספרים
6     */
7     public static int add(int a, int b)
8     {
9         return a + b;
10    }
11
12    /*
13    מקבל: שני מספרים שלמים
14    מחזיר: הפרש מספר ראשון בשני
15    */
16    public static int sub(int a, int b)
17    {
18        return a - b;
19    }
20
21    /*
22    מקבל: שני מספרים שלמים
23    מחזיר: מכפלתם של שני המספרים
24    בהנחה שהמספרים חיוביים
25    */
26    public static int mult(int a, int b)
27    {
28        int sum = 0;
29        for (int i = 0; i < b; i++) {
30            sum += a;
31        }
32        return sum;
33    }
34
35    /*
36    מקבל: שני מספרים שלמים
37    מחזיר: חילוק מספר הראשון בשני כמספר ממשי
38    */
39    public static double divide(int a, int b)
40    {
41        return (double) a / b;
42    }
43
44    /*
45    מקבל: שני מספרים שלמים
46    מחזיר: מנת החלוקה בשיטת חיסור חוזר
47    בהנחה שהמספרים חיוביים
48    */
49    public static int div(int a, int b)
```

```

50     {
51
52         int moneah = 0;
53
54         while (a >= b)
55         {
56             a -= b;
57             moneah++;
58         }
59         return moneah;
60     }
61
62     /*
63     מקבל: שני מספרים שלמים
64     מחזיר: שארית החלוקה
65     בהנחה שהמספרים חיוביים
66     */
67     public static int mod(int a, int b)
68     {
69
70         while (a >= b)
71         {
72             a -= b;
73         }
74         return a;
75     }
76
77     /*
78     מקבל: מספר שלם חיובי
79     מחזיר: אם מספר הוא ראשוני
80     */
81     public static boolean isPrime(int num)
82     {
83         double sq = Math.sqrt(num);
84         for (int i = 2; i <= sq; i++)
85         {
86             if (num%i == 0)
87                 return false;
88         }
89         return true;
90     }
91
92     /*
93     מקבל: מספר חיובי
94     מחזיר: קירוב לשורש של 3 ספרות אחרי הנקודה
95     */
96     public static double squireRoot(int num)
97     {
98         double x = 1;

```

```

99     double y = num/x;
100    while (Math.abs(y-x) > 0.001)
101    {
102        x = (x+y)/2;
103        y = num/x;
104    }
105    return x;
106 }
107
108 /*
109  מדפיס את כל המספרים הקטנים בסדרת פיבונצ'י הקטנים ממספר זה
110  */
111 public static void fiboSmallerNum(int num)
112 {
113     int a = 0, b = 1, c = 0;
114     do {
115         c = a+b;
116         System.out.print(c + ", ");
117         a = b;
118         b = c;
119     } while (c < num);
120     System.out.println();
121 }
122
123 // מדפיס את ה-10 ספרות הראשונות בסדרת פיבונצ'י
124 public static void fibonacci()
125 {
126     System.out.print("0, 1, ");
127     int a = 0, b = 1, c = 0;
128     for (int i = 3; i <= 10 ; i++)
129     {
130         c = a+b;
131         System.out.print(c + ", ");
132         a = b;
133         b = c;
134     }
135     System.out.println();
136 }
137
138
139 /*
140  מקבלת: שני מספרים שלמים וחיובים
141  מחזירה: את המספר הכי גדול שמתחלק בניהם
142  */
143 public static int mmg(int num1, int num2)
144 {
145     while (num1!=num2)
146     {
147         if (num1>num2)

```

```
148         num1 = num1 - num2;
149         else
150             num2 = num2 - num1;
151     }
152     return num1;
153 }
154
155 }
156
```