

```
1 #include "Polynomial.h"
2 #include <cmath>
3
4 double Polynomial::operator()(double aX) const
5 {
6     double result = 0.0;
7
8     for (int i = 0; i <= fDegree; i++) {
9         result += fCoeffs[i] * pow(aX, i);
10    }
11    return result;
12 }
13
14 Polynomial Polynomial::getDerivative() const
15 {
16     Polynomial Result;
17
18     if (fDegree == 0) {
19         return Result;
20     }
21
22     Result.fDegree = fDegree - 1;
23
24     for (size_t i = 1; i <= fDegree; i++) {
25         Result.fCoeffs[i - 1] = fCoeffs[i] * i;
26     }
27
28     return Result;
29 }
30
31 Polynomial Polynomial::getIndefiniteIntegral() const
32 {
33     Polynomial Result;
34
35     Result.fDegree = fDegree + 1;
36
37     for (int i = fDegree; i >= 0; i--) {
38         Result.fCoeffs[i + 1] = fCoeffs[i] / (i + 1);
39     }
40
41     return Result;
42 }
43
44 double Polynomial::getDefiniteIntegral(double aXLow, double aXHigh) const
45 {
46     return this->getIndefiniteIntegral()(aXHigh) - this->getIndefiniteIntegral()(aXLow);
47 }
```