

```
1:  /*
2:   * A2.cpp
3:   *
4:   * Created on: 04.05.2017
5:   * Author: mona & kevin
6:   */
7:
8:
9: #include <iostream>
10: #include <iomanip>
11: #include <complex>
12: #include <cstdlib>
13: #include <vector>
14: #include <cmath>
15: #include <sstream>
16: #include <utility>
17: #include <math.h>
18: #include <fstream>
19: #include <functional>
20: using namespace std;
21:
22: //*****
23: // Funktionsvorschriften
24: //*****
25:
26: double F1(double x){
27:     return exp(x)/x;
28: }
29:
30: double F1_sing(double x, double lim){
31:     return lim+0.5*lim*lim*x;
32: }
33:
34: double F2(double x){
35:     return 2 * exp(-x * x);
36: }
37:
38: double F3(double x){
39:     if(x==0){
40:         return 1;
41:     }
42:     return sin(x)/x;
43: }
44:
45: //*****
46: // Trapezregel
47: //*****
48:
49: double trapez(double a, double b, double N, double(*f)(double)){
50:     double h=(b-a)/N;
51:     double summe=(h/2)*(f(a)+f(b));
52:     for (int n=1; n<N; n++){
53:         summe=summe+h*f(a+n*h);
54:     };
55:     return summe;
56: }
57:
58: //*****
59: // Trapezregel für den singulären Bereich um 0 für Funktion 2
60: // benötigt ein zusätzliches Argument in f: lim.
61: //*****
62:
63: double trapezlim(double a, double b, double N, double(*f)(double,double), double lim){
64:     double h=(b-a)/N;
65:     double summe=(h/2)*(f(a,lim)+f(b,lim));
66:     for (int n=1; n<N; n++){
67:         summe=summe+h*f(a+n*h,lim);
68:     };
69:     return summe;
70: }
71:
72: //*****
73: // GENAUIGKEIT
74: //*****
75:
76: double eps = 1e-5;
77:
78: //*****
79: // Funktion für Aufgabenteil a)
80: //*****
81:
82: double Trapez(double a, double b){
83:     double lim = 0.00001;
84:     double Delta = 1;
85:     double N=2;
86:     //double links;
```

```

87:         double integrall1;
88:         int i = 0;
89:
90:         while(abs(Delta)>eps) {
91:             ++i;
92:             integrall1 = trapez(a,-lim,N,F1)+trapez(lim,b,N,F1)+trapezlim(-1,1,N,F1_sing,lim);
93:             Delta = abs(integrall1-2.1145018)/2.1145018; // relativer Fehler zum analytischen W
ert
94:             N = 2*N;
95:             lim=lim/2;}
96:         cout << "Anzahl der Iterationen: " << i << endl;
97:         cout << "Wert des Integrals 2a): " << integrall1 << endl;
98:         return integrall1;
99:
100:     return integrall1;
101: }
102:
103: //*****
104: // Funktion für Aufgabenteil b)
105: //*****
106:
107: double F2Trapez() {
108:     double up = 2;
109:     double Delta = 1;
110:     double N=2;
111:     double integral2;
112:     int i = 0;
113:
114:     while(abs(Delta)>eps) {
115:         ++i;
116:         integral2 = trapez(0,up,N,F2);
117:         Delta = abs(1.77245385-integral2)/1.77245385; // relativer Fehler zum "wahren" Wert
118:         N = 2*N;
119:         up *= 1.5;}
120:     cout << "\nAnzahl Iterationen: " << i << endl;
121:     cout << "Wert des Integrals 2b): " << integral2 << endl;
122:     return integral2;
123:
124:     return integral2;
125: }
126:
127: //*****
128: // Funktion für Aufgabenteil c)
129: //*****
130:
131: double F3Trapez() {
132:     double ob = 1;
133:     double Delta = 1;
134:     double N=2;
135:     double integral3;
136:     int i = 0;
137:
138:     while(abs(Delta)>eps) {
139:         ++i;
140:         integral3 = 2 * trapez(0,ob,N,F3);
141:         Delta = abs(M_PI-integral3)/M_PI; // relativer Fehler zum "wahren" Wert
142:         N = 2*N;
143:         ob *= 1.5;}
144:     cout << "\nAnzahl Iterationen: " << i << endl;
145:     cout << "Wert des Integrals 2c): " << integral3 << endl;
146:     return integral3;
147:
148:     return integral3;
149: }
150:
151: //*****
152: // main
153: //*****
154:
155: int main() {
156:     int a1 = -1;
157:     int b1 = 1;
158:     //Aufgabenteil a)
159:     cout << "*****\nAufgabe 2a)\n*****\n" << endl;
160:     cout << scientific << setprecision(8) << Trapez(a1,b1) << endl;
161:     //Aufgabenteil b)
162:     cout << "\n*****\nAufgabe 2b)\n*****" << endl;
163:     cout << scientific << setprecision(8) << F2Trapez() << endl;
164:     //Aufgabenteil c)
165:     cout << "\n*****\nAufgabe 2c)\n*****" << endl;
166:     cout << scientific << setprecision(8) << F3Trapez() << endl;
167:
168:     return 0;
169: }

```