|  |  |
| --- | --- |
| **编号:** | **1-1** |

****

信息科学与工程学院实验报告

《面向对象程序设计》

**Object-Oriented Programming**

|  |  |
| --- | --- |
| 姓名： | 姬彬荃 |
| 学号： | 201911020125 |
| 班级： | 计联培1901 |
| 时间： | 2020年10月20日 |

**《面向对象程序设计》实验报告**

**基本要求：**请围绕实验目的、实验内容、实验过程、实验结果（附图）、实验总结（重点阐述）五个部分进行撰写。若报告中若涉及源代码内容，请在附录部分提供完整源码及GitHub源码托管地址。报告撰写完毕后请提交PDF格式版本到云班课。

1. **实验目的**

* 综合运用c++知识解决实际应用问题
* 掌握visual studio 代码调试方法

1. **实验内容**

请采用C++语言编程实现下列3个任务,要求程序代码:清晰简洁易读, 良好的开闭性, 编程风格规范统一。

【编程: 基础题】编写C++风格程序,采用动态内存分配空间的方法,计算斐波那契数列的前N项,并将其存储在动态分配空间中。

【编程: 应用题】编写C++风格程序,采用二分方法计算函数f(x)=0的根,要求采用指针或引用实现, 精度误差控制在10^-6。

【编程: 创新题】基于C++语言设计一款控制台窗口游戏,游戏名称自拟,游戏规则自定义.

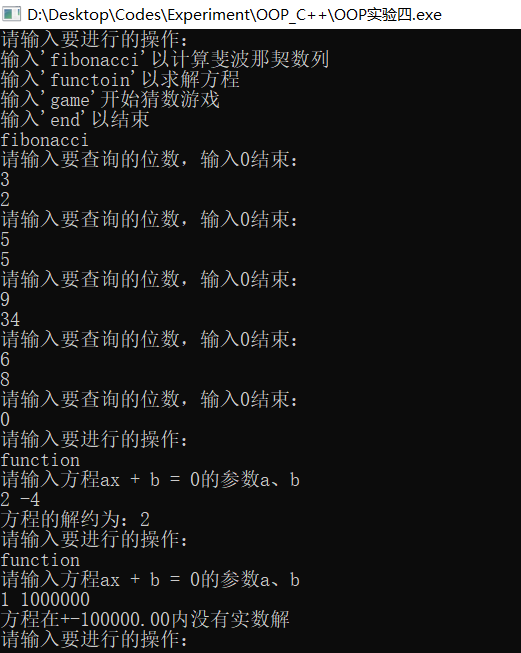
1. **实验过程**

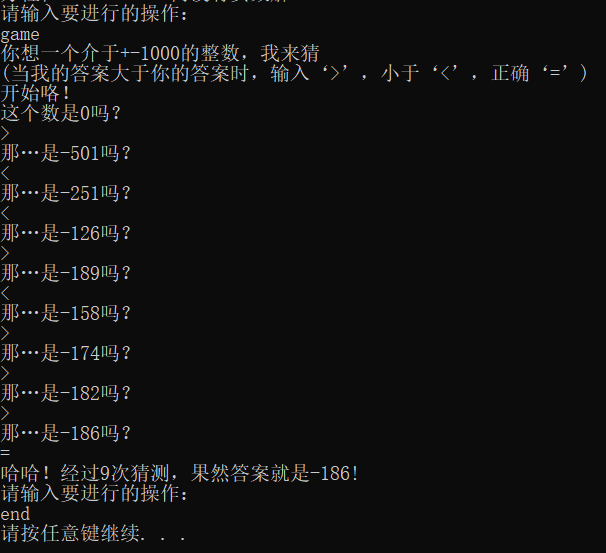
上机时间： 2020年10月20日 晚 19:00 -- 20:45

上机地点： E315机房 (按组序入座)

实验环境： Visual Studio 2019

1. **实验结果**





1. **实验总结**

对于任务一：“,采用动态内存分配空间的方法,计算斐波那契数列的前N项,并将其存储在动态分配空间中”。不难想到使用双链表的方法去对斐波那契数列的前n项进行存储，当查询项数小于历史查询的的最高项数时，遍历链表即可查询到相应项的值。当查询项数大于历史查询的最高项数是，对链表进行更新，在双链表尾加入新结点，新节点的值即为前两个结点的值之和，直到更新到查询所在项。最后直接输出答案即可。

对于任务二：“编写C++风格程序,采用二分方法计算函数f(x)=0的根,要求采用指针或引用实现, 精度误差控制在10^-6”。这个由于在一定区间内如果函数是非单调函数，仅使用零点存在定理很有可能在该区间内无法正确找到函数零点的存在位置，于是仅考虑一元一次方程这种单调性和零点唯一的简单初等函数的求解。我们假设解的区间为+-10000.00然后开始二分查找，查找区间的大小小于给定的精度值时，我们就可以输出其中一个边界作为方程的一个解。

对于任务三：“基于C++语言设计一款控制台窗口游戏,游戏名称自拟,游戏规则自定义.”。受到第二题的启发，就写了一个二分猜数字的小交互游戏。计算机最多在11次以内猜出玩家心中想的+-1000以内的整数。

* **附录：程序源码（建议基于Highlight软件导入）**



1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. \*\* 功能 : OOP课程上机实验课：第4次实验
3. \*\* 作者 : 2019Jibinquan
4. \*\* 创建 : 2020-10-20 / 19:01
5. \*\* 版权 : 431263064@qq.com
6. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
8. #include <iostream>
9. #include <cstdio>
10. #include <cmath>
11. #include <algorithm>
12. #include <cstring>
13. #include <queue>
14. #include <stack>
15. #include <vector>
16. #include <map>
17. #include <set>
18. #define ios ios::sync\_with\_stdio(false);cin.tie(0);cout.tie(0);
19. #define debug(a) cout << #a << " " << a << endl
20. **using** **namespace** std;
21. **typedef** **long** **long** ll;
22. **const** **double** pi = acos(-1);
23. **const** **double** eps = 1e-6;
24. **const** **int** inf = 0x3f3f3f3f;
25. **const** **int** maxn = 100007;//1e5+7
26. **const** ll mod = 1000000007;//1e9+7
28. /\*----------------------------------\*
29. Class Definition
30. \*-----------------------------------\*/
32. **typedef** **struct** LNode {
33. **int** data;
34. LNode\* next;
35. LNode\* pre;
36. }LNode, \* Linklist;
38. Linklist initList() //初始化链表
39. {
40. Linklist L = **new** LNode;
41. L->data = 0;
42. L->next = NULL;
43. L->pre = NULL;
44. **return** L;
45. }

48. **void** insert\_back(Linklist& L, **int** e) //尾插法插入元素
49. {
50. LNode\* p = L;
51. **while** (p->next != NULL) {
52. p = p->next;
53. }
54. LNode\* s = **new** LNode;
55. s->data = e;
56. s->next = NULL;
57. s->pre = p;
58. p->next = s;
59. L->data++;
60. }
62. **void** distorylist(Linklist& L)//销毁链表
63. {
64. LNode\* p = L;
65. **while** (p->next != NULL) {
66. LNode\* s = p;
67. p = p->next;
68. **delete** s;
69. }
70. **delete** p;
71. }
73. **void** Fbi()//斐波那契查询
74. {
75. **int** n;
76. Linklist L = initList();
77. insert\_back(L, 1);
78. insert\_back(L, 1);
79. cout << "请输入要查询的位数，输入0结束：" << endl;
80. **while** (cin >> n && n) {
81. LNode\* p = L;
82. **if** (n <= L->data) {
83. **for** (**int** i = 1; i <= n; i++) {
84. p = p->next;
85. }
86. cout << p->data << endl;
87. }
88. **else** {
89. **while** (p->next != NULL) {
90. p = p->next;
91. }
92. **while** (L->data < n) {
93. **int** a = p->data;
94. **int** b = p->pre->data;
95. //cout << a << ' ' << b << endl;
96. LNode\* s = **new** LNode;
97. s->data = a + b;
98. s->pre = p;
99. s->next = NULL;
100. p->next = s;
101. p = p->next;
102. L->data++;
103. }
104. cout << p->data << endl;
105. }
106. //cout << L->data << endl;
107. cout << "请输入要查询的位数，输入0结束：" << endl;
108. }
109. distorylist(L);
110. }
112. **class** Fun{
113. **private**:
114. **double** a,b;
115. **public**:
116. **double** f(**double** x)
117. {
118. //cout<<" \*\* "<<a<<" "<<b<<" "<<x<<" "<<a \* x + b<<endl;
119. **return** (a \* x + b);
120. }
121. **void** slove()
122. {
123. **double** l = -100000.000,r = 100000.00;
124. cout<<"请输入方程ax + b = 0"<<endl;
125. cin>>a>>b;
126. **if**(f(l) == 0){
127. cout<<"方程的解为："<<l<<endl;
128. }
129. **else** **if**(f(r) == 0){
130. cout<<"方程的解为："<<r<<endl;
131. }
132. **else** **if**(f(l) \* f(r) > 0){
133. cout<<"方程在+-100000.00内没有实数解"<<endl;
134. }
135. **else**{
136. **bool** flag = 0;
137. **while**(r - l > eps){
138. **double** mid = (l + r) / 2.0;
139. **double** fm = f(mid);
140. **double** fl = f(l);
141. **double** fr = f(r);
142. //cout<<" mid "<<mid<<" f(mid) "<<fm<<endl<<" l "<<l<<" f(l) "<<fl<<endl<<" r "<<r<<" f(r) "<<fr<<endl<<endl;
143. **if**(fm == 0){
144. cout<<"方程的解为："<<mid<<endl;
145. flag = 1;
146. **break** ;
147. }
148. **else** **if**(fm \* fl < 0){
149. r = mid;
150. }
151. **else**{
152. l = mid;
153. }
154. }
155. **if**(!flag){
156. //cout<<"方程的解介于："<<l<<" 和 "<<r<<" 之间"<<endl;
157. cout<<"方程的解约为："<<l<<endl;
158. }
159. }
160. }
161. };
163. **class** Game{
164. **public**:
165. **void** guess(){
166. cout<<"你想一个介于+-1000的整数，我来猜"<<endl;
167. cout<<"(当我的答案大于你的答案时，输入‘>’，小于‘<’，正确‘=’)"<<endl;
168. cout<<"开始咯！"<<endl;
169. **int** l = -1001,r = 1001;
170. **char** ch;
171. **int** cnt = 1;
172. **int** mid = 0;
173. cout<<"这个数是0吗？"<<endl;
174. **while**(cin >> ch && ch != '='){
175. **if**(ch == '>'){
176. r = mid - 1;
177. }
178. **else**{
179. l = mid;
180. }
181. mid = (r + l) / 2;
182. //cout<<" mid "<<mid<<endl<<" l "<<l<<endl<<" r "<<r<<endl;
183. cout<<"那…是"<<mid<<"吗？"<<endl;
184. cnt++;
185. }
186. cout<<"哈哈！经过"<<cnt<<"次猜测，果然答案就是"<<mid<<"!"<<endl;
187. }
188. };
190. /\*----------------------------------\*
191. Main Function
192. \*-----------------------------------\*/
194. **int** main()
195. {
196. string op;
197. cout << "请输入要进行的操作：" << endl;
198. cout << "输入'fibonacci'以计算斐波那契数列 " << endl;
199. cout << "输入'functoin'以求解方程 " << endl;
200. cout << "输入'game'开始猜数游戏 " << endl;
201. cout << "输入'end'以结束 " << endl;
202. **while** (cin >> op) {
203. **if** (op == "end") {
204. **break**;
205. }
206. **else** **if** (op == "fibonacci") {
207. Fbi();
208. }
209. **else** **if** (op == "function") {
210. Fun fun;
211. fun.slove();
212. }
213. **else** **if** (op == "game") {
214. Game game;
215. game.guess();
216. }
217. **else** {
218. cout << "指令错误！" << endl;
219. }
220. cout << "请输入要进行的操作：" << endl;
221. }
222. system("pause");
223. **return** 0;
224. }

GitHub地址：