## 多种方式读取文件

使用C语言读取：

1. int main(){
2. FILE\* fp = fopen("data.txt", "r");
3. if(fp == NULL) return -1;
4. char buffer[10][20] = {'\0'};
5. for(int i=0; i<10; i++){
6. fgets(buffer[i], sizeof(buffer[i]), fp);
7. size\_t len = strcspn(buffer[i], "\n");
8. buffer[i][len] = '\0';
9. std::cout<<buffer[i]<<std::endl;
10. }
12. fclose(fp);
13. return 0;
14. }

另一种直接在下面的程序中实现了

## C++实现抽奖程序

1. #include <iostream>
2. #include <fstream>
3. #include <vector>
4. #include <set>
5. #include <cstdlib>
6. #include <ctime>
7. #include <iomanip>
8. #include <conio.h>
9. #include <windows.h>
11. using namespace std;
12. class Prize {
13. protected:
14. string name;
15. int total;
16. vector<string> winners;
17. public:
18. Prize(const string& n, int t) : name(n), total(t) {}
19. virtual ~Prize() {}
21. bool canDraw() const { return winners.size() < total; }
22. void addWinner(const string& num) { winners.push\_back(num); }
23. void showWinners() const {
24. cout << name << "获奖者：" << endl;
25. for (const auto& w : winners) cout << w << endl;
26. }
27. int remain() const { return total - winners.size(); }
28. const string& getName() const { return name; }
29. };
30. class FirstPrize : public Prize {
31. public:
32. FirstPrize() : Prize("一等奖", 1) {}
33. };
34. class SecondPrize : public Prize {
35. public:
36. SecondPrize() : Prize("二等奖", 2) {}
37. };
38. class ThirdPrize : public Prize {
39. public:
40. ThirdPrize() : Prize("三等奖", 5) {}
41. };
42. class MenuSystem {
43. private:
44. vector<string> participants;
45. set<string> drawnNumbers;
46. FirstPrize fp;
47. SecondPrize sp;
48. ThirdPrize tp;
50. vector<string> loadParticipants(const string& filename) {
51. vector<string> nums;
52. ifstream infile(filename);
53. string line;
54. while (getline(infile, line)) {
55. nums.push\_back(line);
56. }
57. return nums;
58. }
60. void moveCursorToLine(int line) {*// 移动光标到指定行*
61. std::cout << "\033[" << line << ";0H";
62. }
63. void refreshLines(const std::vector<std::string>& contents) {
64. std::cout << "\033[s"; *// 保存光标位置*
65. for (size\_t i = 0; i < contents.size(); ++i) {
66. moveCursorToLine(i + 1);   *// 第i行*
67. std::cout << "\033[K" << contents[i]; *// 清行+输出*
68. }
69. std::cout << "\033[u"; *// 恢复光标位置*
70. std::cout.flush();
71. }
72. void printRandom(Prize& pr, int n){
73. string name = pr.getName()+": ";
74. for(int i=0; i<n; i++){
75. vector<string> content{name};
76. int index = rand()%10;
77. content.push\_back(participants[index]);
78. Sleep(30);
79. refreshLines(content);
80. }
81. }
82. string drawRandom() {
83. vector<string> available;
84. for (const auto& p : participants) {
85. if (drawnNumbers.find(p) == drawnNumbers.end())
86. available.push\_back(p);
87. }
88. if (available.empty()) return "";
90. int idx = rand() % available.size();
91. drawnNumbers.insert(available[idx]);
92. return available[idx];
93. }
94. public:
95. MenuSystem(const char\* file) {
96. participants = loadParticipants(file);
97. srand(time(NULL));
98. }
99. void startDrawing() {
100. cout << "\n按任意键开始抽奖..." << endl;
101. \_getch();
102. printRandom(fp, 10);
103. while (fp.canDraw()) {
104. string winner = drawRandom();
105. fp.addWinner(winner);
106. system("cls");
107. cout << fp.getName() << ": " << winner << endl;
108. }
109. cout << "\n按任意键开始抽奖..." << endl;
110. \_getch();
111. system("cls");
112. printRandom(sp, 15);
113. while(sp.canDraw()) {
114. string winner = drawRandom();
115. sp.addWinner(winner);
116. cout << sp.getName() << ": " << winner << endl;
117. }
119. cout << "\n按任意键开始抽奖..." << endl;
120. \_getch();
121. system("cls");
122. printRandom(tp, 20);
123. while (tp.canDraw()) {
124. string winner = drawRandom();
125. tp.addWinner(winner);
126. cout << tp.getName() << ": " << winner << endl;
127. }
129. cout << "\n按任意键查看结果..." << endl;
130. \_getch();
131. system("cls");
132. cout<<"  结果汇总中...\n";
133. Sleep(500);
134. cout << "\n=== 抽奖结果 ===" << endl;
135. fp.showWinners();
136. cout<<"\n";
137. sp.showWinners();
138. cout<<"\n";
139. tp.showWinners();
140. }
141. };
142. int main() {
143. MenuSystem ms("data.txt");
144. ms.startDrawing();
145. return 0;
146. }