**Практическая работа №16**

**Вариант №6 – Реализовать задачу о рюкзаке методом ветвей и границ.**

**Автор: Николаев-Аксенов И. С.**

**Группа: ИКБО-20-19**

**Код программы:**

1. #include <iostream>
2. #include <algorithm>
3. #include <queue>
4. using namespace std;
6. struct Item
7. {
8. float weight;
9. int value;
10. };
12. struct Node
13. {
14. int level, profit, bound;
15. float weight;
16. };
18. bool cmp(Item a, Item b)
19. {
20. double r1 = (double) a.value / a.weight;
21. double r2 = (double) b.value / b.weight;
22. return r1 > r2;
23. }
25. int bound(Node u, int n, int knapsackWeight, Item arr[])
26. {
27. if (u.weight >= knapsackWeight)
28. return 0;
30. int profit\_bound = u.profit;
32. int j = u.level + 1;
33. int totalWeight = u.weight;
35. while ((j < n) && (totalWeight + arr[j].weight <= knapsackWeight))
36. {
37. totalWeight += arr[j].weight;
38. profit\_bound += arr[j].value;
39. j++;
40. }
42. if (j < n)
43. profit\_bound += (knapsackWeight - totalWeight) \* arr[j].value / arr[j].weight;
45. return profit\_bound;
46. }
48. int knapsack(int W, Item arr[], int n)
49. {
50. sort(arr, arr + n, cmp);
52. queue<Node> Q;
53. Node u, v;
55. u.level = -1;
56. u.profit = u.weight = 0;
57. Q.push(u);
59. int maxProfit = 0;
60. while (!Q.empty())
61. {
62. u = Q.front();
63. Q.pop();
65. if (u.level == -1)
66. v.level = 0;
68. if (u.level == n - 1)
69. continue;
71. v.level = u.level + 1;
72. v.weight = u.weight + arr[v.level].weight;
73. v.profit = u.profit + arr[v.level].value;
75. if (v.weight <= W && v.profit > maxProfit)
76. maxProfit = v.profit;
78. v.bound = bound(v, n, W, arr);
80. if (v.bound > maxProfit)
81. Q.push(v);
83. v.weight = u.weight;
84. v.profit = u.profit;
85. v.bound = bound(v, n, W, arr);
86. if (v.bound > maxProfit)
87. Q.push(v);
88. }
90. return maxProfit;
91. }
93. int main() {
94. setlocale(LC\_ALL, "Russian");
96. int W = 10;
97. Item arr[] = {{2,    40}, {3.14, 50}, {1.98, 120}, {5,    95}, {3,    30}};
98. int n = sizeof(arr) / sizeof(arr[0]);
100. cout << "Маскимальная стоимость: " << knapsack(W, arr, n);
102. return 0;
103. }

**Результат выполнения программы:**

