

```

1      /*
      File: strelka3.cpp

      Compiler: VC++ 3.1

      MSP, 30.10.07, 14:43

      Вопрос.
      Что такое стрелка: -> ?

      Объяснение.
      Part 3, последняя.
      */

10     # include <stdio.h>
11     # include <conio.h>

12     class spisok
13     {
14     public:
15         spisok * L; // указатель на элемент списка слева
16         int data; // данные элемента списка
17         spisok * R; // указатель на элемент списка справа
18     }; // это правильно, здесь НУЖНО ставить ;

19     int main ()
20     {
21         clrscr ();

22         // Вот, теперь, только в третьем файле этого урока,
23         // я смогу тебе объяснить, что же такое -> и
24         // как ее применяют и КОГДА.

25         // Если структура-класс большие, то обычно применяют не
26         // просто объявление переменной, а УКАЗАТЕЛЬ на нее, указатель на
27         // переменную структуры-класса, и тогда для ДОСТУПА к элементам
28         // структуры применяют уже не ТОЧКУ, а СТРЕЛОЧКУ!

29         spisok * element; // создан УКАЗАТЕЛЬ на структуру element

30         // двусвязный список-дек-кольцо состоит из элементов,
31         // пока порожден ВСЕГО ОДИН только указатель на элемент
32         // списка, да и то, этот указатель МУСОРНЫЙ, имеет неопределенное
33         // значение, какое-угодно, хаотическое, случайное, что является
34         // беспорядком, который нужно устранить, и попросить САМУ ОС дать
35         // легальный адрес для хранения ПЕРВОГО элемента списка. Эта
36         // просьба осуществляется ОПЕРАТОРОМ new.

37         element = new spisok;

```

```

38 // Как в этом случае обратиться в поле L?
39 // Вот так:
40 // (* element).L
41 // Так мы делали в Part 2 этого урока
42 // Так вот, есть соглашение, по которому,
43 // вместо того, чтобы писать
44 // (* element).L
45 // будем писать
46 // element -> L.

47 // Выражения
48 // (* element).L
49 // element -> L
50 // эквивалентны!

51 // Теперь разберемся в том, как же добраться до ПОЛЕЙ СЛОЖНОЙ
52 // переменной element, и именно до тех маленьких переменных,
53 // из которых она сама состоит?

54 // Как обратиться к L, data, R?
55 // -- через СТРЕЛОЧКУ, потому что теперь element уже УКАЗАТЕЛЬ.
56 // таково ПРАВИЛО языка, если переменная УКАЗАТЕЛЬ, то
57 // до доступ к ее полям идет через стрелочку.

58 // Вот так:
59 // element -> L - обратились к указателю на
60 // предыдущий элемент списка
61 // element -> data - обратились к данному элемента списка
62 // element -> R - обратились к указателю на
63 // следующий элемент списка

64 // Проинициализируем element
65 // Выделим память для L и R, зададим значение data

66 element -> L = new spisok; // сама OS даст адресное пространство
67 // A2..(A2 + 9)

68 printf ("Введите число: ");
69 scanf ("%i", & (element -> data)); // введем 20
70 fflush (stdin);

71 element -> R = new spisok; // сама OS даст адресное пространство
72 // A3..(A3 + 9)

73 // Итак, с адреса A1 располагается в памяти PC переменной
74 // element
75 // теперь оно строго определено самой OS по требованию
76 // программиста:
77 // A2, 20, A3

```

```

78 // element == A2 + 123 + A3 - единое целое, записанное по адресу
79 // A1

80 // element имеет АДРЕС A1
81 // В element -> L записан АДРЕС A2
82 // в element -> data записано, например, число 20
83 // в element -> R записан АДРЕС A3
84 // Все адреса предоставлены OS

85 // element -> L == адрес, он имеет тип spisok *
86 // В списке дека сейчас всего 1 элемент == element
87 // Получим элемент списка СЛЕВА, то есть предыдущий
88 // Создадим его.
89 // Память под него выделена OS - 10 байт, но они все "мусорные",
90 // проинициализируем предыдущий элемент

91 // | element -> L - адрес A2
92 // |
93 // |
94 // v
95 // (L, data, R) ===== (* (element -> L)) - то,
96 // что лежит по адресу A2

97 (element -> L) -> L = new spisok; // получили адрес A4, для
98 // элемента, который находится
99 // левее, чем A2

100 printf ("Введите число: ");
101 scanf ("%i", & ((element -> L) -> data)); // введем 10
102 fflush (stdin);

103 element -> L -> R = new spisok; // получили адрес A5, для
104 // элемента, который находится
105 // правее, чем A2

106 // заполним элемент, правее element
107 element -> R -> L = new spisok;

108 printf ("Введите число: ");
109 scanf ("%i", & element -> R -> data); // введем 30
110 fflush (stdin);

111 element -> R -> R = new spisok;

112 printf ("element -> data = %i\n", element -> data);
113 printf ("element -> L -> data = %i\n", element -> L -> data);
114 printf ("element -> R -> data = %i\n", element -> R -> data);

115 // вот, теперь, все, теперь объяснено, что такое СТРЕЛОЧКА.

116 getch ();

```

```
117     return 0;
118 }
```

Listing данной задачи опубликован в сети Internet по адресу  
<http://www.Best-Listing.ru/color-3-task-20.html>

Sergey Mitrofanov, 30.10.13, 21:27

E-mail: [infostar@mail.ru](mailto:infostar@mail.ru)

© <http://www.Best-Listing.ru/>, 2006–2013