

Абстрактные типы данных С++

Алгоритмы
и алгоритмические языки

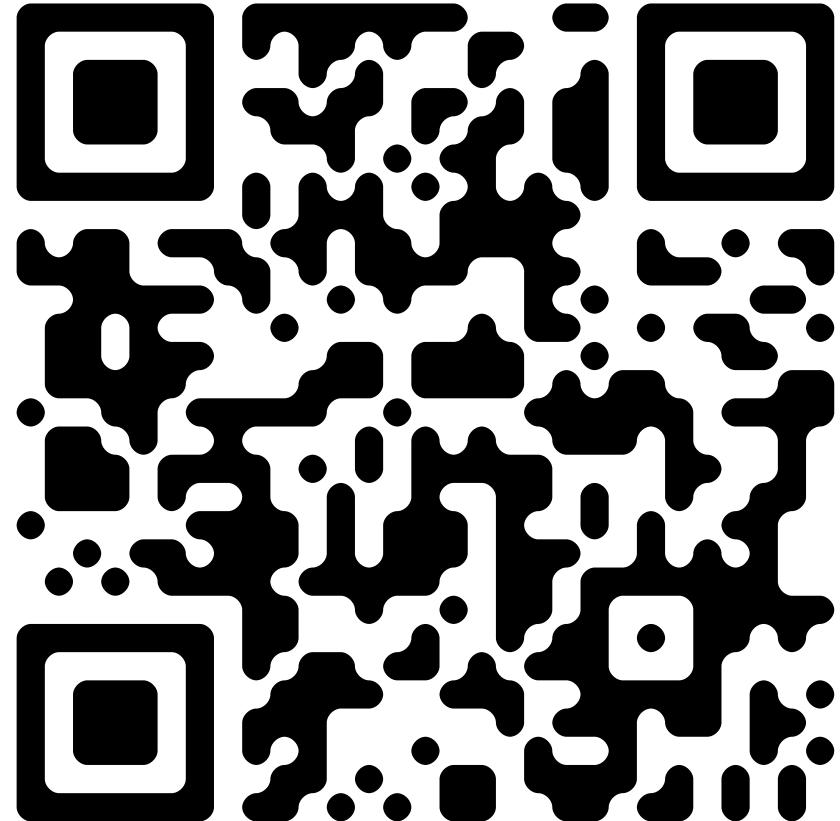
goo.gl/c8pyqX

Лекция 6, 12 октября, 2018

Лектор:

Дмитрий Северов, кафедра информатики 608 КПМ

dseverov@mail.mipt.ru



http://cs.mipt.ru/wp/?page_id=6077

«ЧТО ПОЗВОЛЕНО ЮПИТЕРУ ...»

```
#include <stdio.h>
#define N ...
struct { int a[N];
} a,b;
int main( ) {
...
a=b;      //разрешено
...
return 0; }
```

```
#include <stdio.h>
#define N ...
int a[N],b[N];
int main( ) {
...
a=b;      // ошибка
...
return 0; }
```

Dev-C++ 4.9.9.2

Файл Правка Поиск Вид Проект Выполнить Отладка Сервис CVS Окно Справка

Создать Вставить Переключить Перейти

Проект Классы Отладка tst9.cpp

```
#include <iostream>

const int N=30;
struct { int a[N]; } a,b;

int main() {
    for(int i=0; i<N; i++) { b.a[i]=i; a.a[i]=0; }
    a=b;
    for(int i=0; i<N; i++) std::cout << a.a[i] << ' '; std::cout << std::endl;

    return 0;
}
```

d:\A1\Lect2013\Exmp\4\tst9.exe

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска Закрыть

Отладка Отслеживать Вывод

Следующий шаг Продолжить выполнение Отладка Добавить в наблюдаемые
Шаг внутрь Выполнить до курсора Остановить выполнение Удалить объект наблюдения

11:9 Вставка 13 строк в файле

Пример 4: односвязный список

- Определение формата узла списка

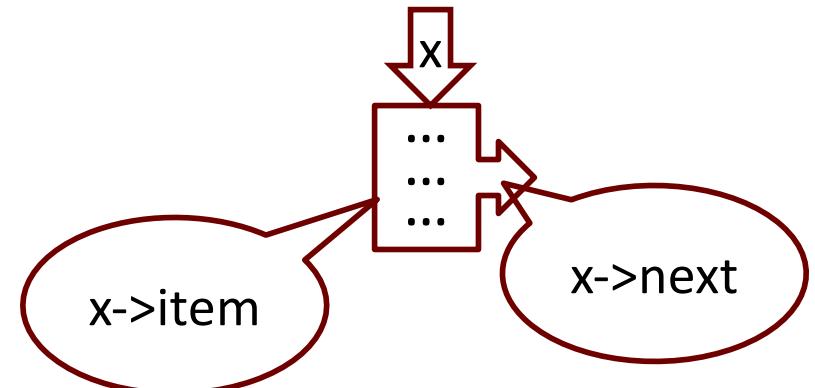
```
struct node {  
    My_type item;  
    struct node *next;  
};  
  
typedef struct node *link;
```

- Выделение памяти для узла СПИСКА

```
link x = (link)malloc(sizeof(struct node));
```

- Инициализация элементов узла

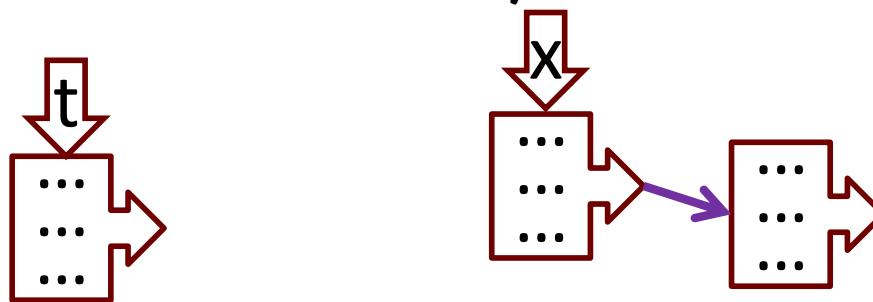
```
( *x ).item = ... , x->next = NULL;
```



Объединение элементов в список (1)

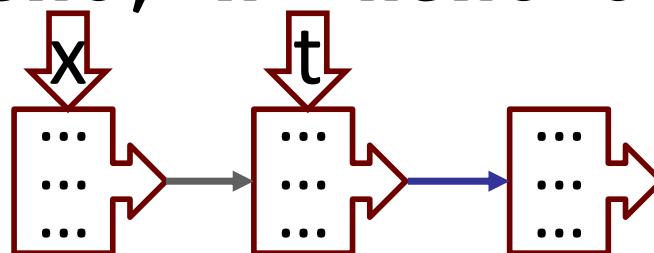
■ Вставка узла t за узлом x

```
t->next = x->next, x->next = t;
```



■ Удаление узла t за узлом x

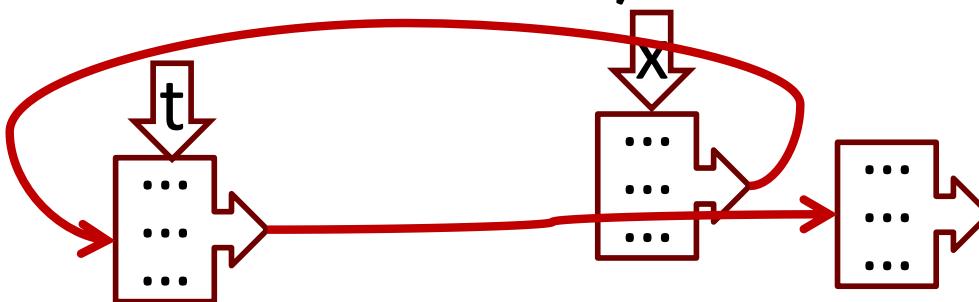
```
t = x->next; x->next=t->next; free(t);
```



Объединение элементов в список (1)

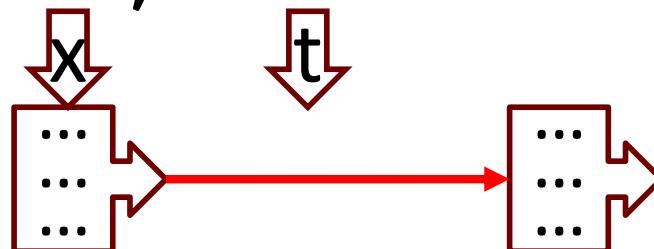
- Вставка узла t за текущим узлом x

```
 $t->next = x->next, x->next = t;$ 
```



- Удаление узла, следующего за x

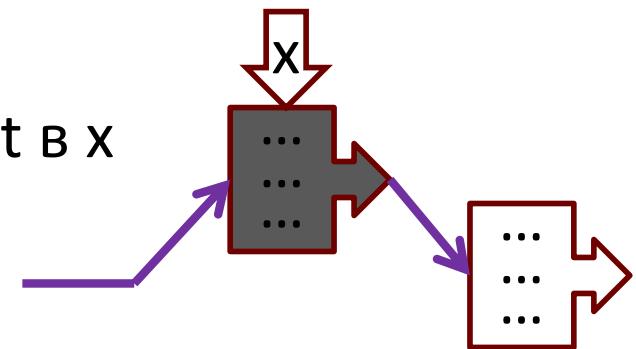
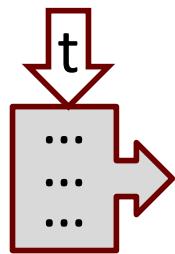
```
 $t = x->next, x->next=t->next, free(t);$ 
```



Объединение элементов в список (2)

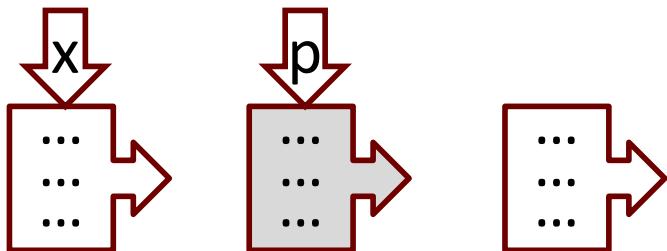
- Вставка узла t перед текущим узлом x со сменой текущего

```
link g = new node;      // новый узел g  
*g = *x;                // g дубль x, перед у  
x->next = g;            // g после x  
x->item = t->item;      // содержимое t в x
```



- Удаление узла x со сменой текущего

```
link p = x->next;        // к удалению  
*x = *x->next;          // перенос значения  
delete p;                 // удаление
```



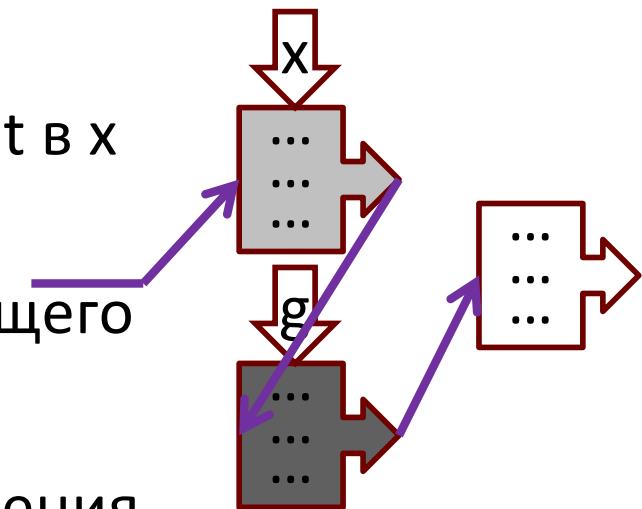
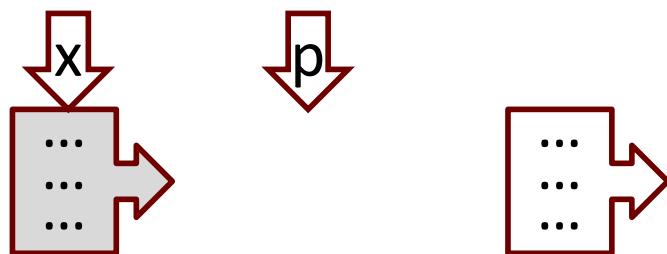
Объединение элементов в список (2)

- Вставка узла t перед текущим узлом x со сменой текущего

```
link g = new node;      // новый узел g  
*g = *x;                // g дубль x, перед у  
x->next = g;            // g после x  
x->item = t->item;      // содержимое t в x
```

- Удаление текущего узла x со сменой текущего

```
link p = x->next;        // к удалению  
*x = *x->next;          // перенос значения  
delete p;                // удаление
```



D:\Disk_R\tst\tst.c - Dev-C++ 5.11

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

TDM-GCC 4.9.2 64-bit Release

(globals)

Проект Классы 0 < > [*] tst.c

```
1 #include <stdio.h>
2 #include <malloc.h>
3 #define new(x) (x*)malloc(sizeof(x))
4
5 typedef struct node *link;      // связка - указатель на узел
6 typedef struct node { int Num; link Next; } item;    // узел или элемент списка
7
8 int main() {
9     int n,m;
10    link top, p;           // вершина списка и бегунок
11
12    scanf("%d%d",&n,&m);
13    top = p = new(item);
14    for(int i=1;i<n;i++) { p->Num = i; p = (p->Next = new(item)); }
15    p->Num = n; p = (p->Next = top);
16
17    for (int k=m%n;n>1;k=m%--n)
18        if(k==1) *p = *p->Next;
19    else { if(!k) k=n;
20          for(int i=1;i<k-1;i++) p=p->Next;
21          p = (p->Next = p->Next->Next); }
22
23    printf("%d\n",p->Num);
24
25    return 0;
26 }
```

D:\Disk_R\tst\tst.exe

1000 2000
113

Process exited after 4.041 seconds with return value 0

Для продолжения нажмите любую клавишу . . .

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

Line: 10 Col: 60 Sel: 0 Lines: 26 Length: 649 Вставка Done parsing in 0 seconds

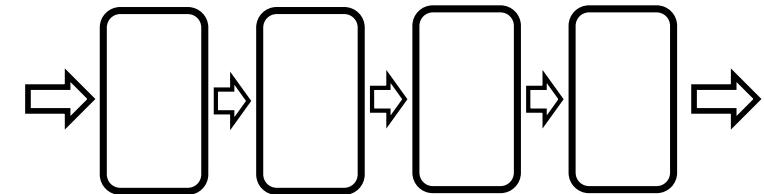
Введение

- Почему типы «абстрактные»?
- Реализации массивами и структурами

СПИСОК, ОЧЕРЕДЬ, СТЕК

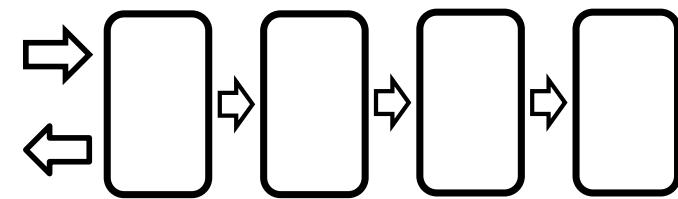
Очередь

FIFO – First In First Out



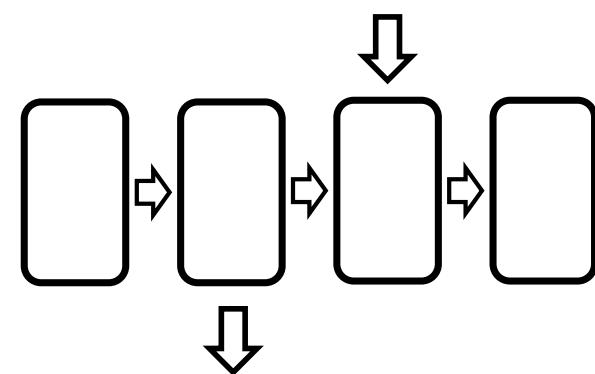
Стек

LIFO – Last In First Out



Список

Casual In Casual Out



РЕАЛИЗАЦИЯ МАССИВАМИ

Стек. Компоненты

```
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>

typedef int T;
typedef struct Stack *Link_Stack;

inline void ERR(const char* s) {
    fputs(s,stderr); exit(1); }
```

Стек. Конструкция

```
struct Stack { int NumEl, Size; T Items[ ]; };
```

```
Link_Stack Stack_ini(int N) {
    Link_Stack a = (Link_Stack)malloc(
        sizeof(struct Stack) + N*sizeof(T));
    a->NumEl=0; a->Size=N; return a; }
```

```
Link_Stack Stack_free(Link_Stack a) {
    free(a); return NULL; }
```

Стек. Состояние

```
int Stack_Is_Empty(Link_Stack a) {  
    return a->NumEl==0; }
```

```
int Stack_Is_Full(Link_Stack a) {  
    return a->NumEl==a->Size; }
```

Стек. Действия

```
void Stack_push(Link_Stack a, T New_el) {  
    if(Stack_Is_Full(a))  
        ERR("Stack::push: stack is full");  
    a->Items[a->NumEl++]=New_el; }  
  
T Stack_pop(Link_Stack a) {  
    if(Stack_Is_Empty(a))  
        ERR("Stack::pop: stack is empty");  
    return a->Items[--a->NumEl]; }
```

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка



TDM-GCC 4.9.2 64-bit Release

(globals)

t1.c

```
1 #include <stdio.h>
2 #include <malloc.h>
3 #include <stdlib.h>
4 inline void ERR(const char* s) { fputs(s,stderr); exit(1); }
5
6 typedef int T;
7 typedef struct Stack *Link_Stack;
8 struct Stack { int NumEl, Size; T Items[]; };
9
10 Link_Stack Stack_ini(int N) { Link_Stack a = (Link_Stack)malloc(sizeof(struct Stack) + N*sizeof(T));
11     a->NumEl=0; a->Size=N; return a; }
12 Link_Stack Stack_free(Link_Stack a) { free(a); return NULL; }
13 int Stack_Is_Empty(const Link_Stack a) { return a->NumEl==0; }
14 int Stack_Is_Full(const Link_Stack a) { return a->NumEl==a->Size; }
15 void Stack_push(Link_Stack a, T New_el) { if(Stack_Is_Full(a)) ERR("Stack::push: stack is full");
16     a->Items[a->NumEl++]=New_el; }
17 T Stack_pop(Link_Stack a) { if(Stack_Is_Empty(a)) ERR("Stack::pop: stack is empty"); return a->Items [--a->NumEl]; }
18
19 int main() {
20     int k,N;
21     scanf("%d",&N);
22     Link_Stack a = Stack_ini(N);
23     while(scanf("%d",&k)!=EOF) Stack_push(a,k);
24     while(!Stack_Is_Empty(a)) { k=Stack_pop(a); printf("%d ",k); }
25     a=Stack_free(a);
26
27     return 0;
28 }
```



(globals)

t1.c

```

1 #include <stdio.h>
2 #include <malloc.h>
3 #include <stdlib.h>
4 inline void ERR(const char* s) { fputs(s,stderr); exit(1);
5
6 typedef int T;
7 typedef struct Stack *Link_Stack;
8 struct Stack { int NumEl, Size; T Items[]; };
9
10 Link_Stack Stack_ini(int N) { Link_Stack a = (Link_Stack)malloc(sizeof(struct Stack) + N*sizeof(T));
11     a->NumEl=0; a->Size=N; return a; }
12 Link_Stack Stack_free(Link_Stack a) { free(a); return NULL; }
13 int Stack_Is_Empty(const Link_Stack a) { return a->NumEl==0; }
14 int Stack_Is_Full(const Link_Stack a) { return a->NumEl==a->Size; }
15 void Stack_push(Link_Stack a, T New_el) { if(Stack_Is_Full(a)) ERR("Stack::push: stack is full");
16     a->Items[a->NumEl++]=New_el; }
17 T Stack_pop(Link_Stack a) { if(Stack_Is_Empty(a)) ERR("Stack::pop: stack is empty"); return a->Items [--a->NumEl]; }
18
19 int main() {
20     int k,N;
21     scanf("%d",&N);
22     Link_Stack a = Stack_ini(N);
23     while(scanf("%d",&k)!=EOF) Stack_push(a,k);
24     while(!Stack_Is_Empty(a)) { k=Stack_pop(a); printf("%d\n",k);
25     a=Stack_free(a);
26
27     return 0;
28 }
```

D:\Disk_S\Lect2016\Exmp\9\mas_c\t1.exe

5
1 2 3 4 5 6
Stack::push: stack is full

Process exited after 15.57 seconds with return value 1
Для продолжения нажмите любую клавишу . . .

D:\Disk_S\Lect2016\Exmp\9\mas_c\t1.exe

10
1 2 3 4 5 6 7 8 9 0
^Z
0 9 8 7 6 5 4 3 2 1

Process exited after 14.26 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

Очередь. Компоненты

```
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>

typedef int T;

typedef struct Queue *Link_Queue;

inline void ERR(const char* s) {
    fputs(s, stderr); exit(1); }
```

Очередь. Конструкция

```
struct Queue { int NumEl, Size, P; T Items[ ] ; };
```

```
Link_Queue Queue_ini(int N) { Link_Queue a =  
    (Link_Queue)malloc(  
        sizeof(struct Queue) + N*sizeof(T));  
    a->Size=N; a->NumEl=a->P=0; return a; }
```

```
Link_Queue Queue_free(Link_Queue a) { free(a);  
    return NULL; }
```

Очередь. Состояние

```
int Queue_Is_Empty(Link_Queue a) {  
    return a->NumEl==0; }
```

```
int Queue_Is_Full(Link_Queue a) {  
    return a->NumEl==a->Size; }
```

Очередь. Действия

```
void Queue_put(Link_Queue a, T New_el) {
    if(Queue_Is_Full(a))
        ERR("Queue::put: queue is full");
    a->Items[ a->P++ ]=New_el; a->NumEl++;
    if( a->P==a->Size) a->P=0; }
```

```
T Queue_get(Link_Queue a) {
    if(Queue_Is_Empty(a))
        ERR("Queue::get: queue is empty");
    return a->Items[ a->P - a->NumEl-- +
        ( a->P <= a->NumEl?a->Size:0 )]; }
```

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка



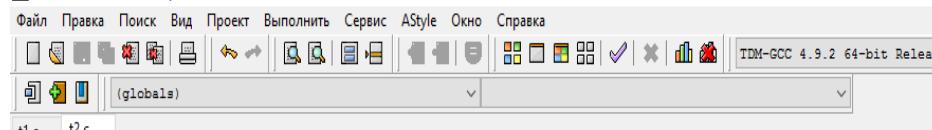
TDM-GCC 4.9.2 64-bit Release



t1.c t2.c

```
1 #include <stdio.h>
2 #include <malloc.h>
3 #include <stdlib.h>
4 typedef int T;
5 typedef struct Queue *Link_Queue;
6 inline void ERR(const char* s) { fputs(s,stderr); exit(1); }
7
8 struct Queue { int NumEl, Size, P; T Items[]; };
9
10 Link_Queue Queue_ini(int N) { Link_Queue a = (Link_Queue)malloc(sizeof(struct Queue) + N*sizeof(T));
11     a->Size=N; a->NumEl=a->P=0; return a; }
12 Link_Queue Queue_free(Link_Queue a) { free(a); return NULL; }
13 int Queue_Is_Empty(const Link_Queue a) { return a->NumEl==0; }
14 int Queue_Is_Full(const Link_Queue a) { return a->NumEl==a->Size; }
15 void Queue_put(Link_Queue a, T New_el) { if(Queue_Is_Full(a)) ERR("Queue::put: queue is full");
16     a->Items[a->P++]=New_el; a->NumEl++; if(a->P==a->Size) a->P=0; }
17 T Queue_get(Link_Queue a) { if(Queue_Is_Empty(a)) ERR("Queue::get: queue is empty");
18     return a->Items[a->P - a->NumEl-- + (a->P<=a->NumEl?a->Size:0)]; }
19
20 int main() {
21     int k,N;
22     scanf("%d",&N);
23     Link_Queue a = Queue_ini(N);
24     while(scanf("%d",&k)!=EOF) Queue_put(a,k);
25     while(!Queue_Is_Empty(a)) { k=Queue_get(a); printf("%d ",k); }
26     a=Queue_free(a);
27     return 0;
28 }
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска



```

1 #include <stdio.h>
2 #include <malloc.h>
3 #include <stdlib.h>
4 typedef int T;
5 typedef struct Queue *Link_Queue;
6 inline void ERR(const char* s) { fputs(s,stderr); exit(1); }
7
8 struct Queue { int NumEl, Size, P; T Items[]; };
9
10 Link_Queue Queue_ini(int N) { Link_Queue a = (Link_Queue)malloc(sizeof(struct Queue) + N*sizeof(T));
11     a->Size=N; a->NumEl=a->P=0; return a; }
12 Link_Queue Queue_free(Link_Queue a) { free(a); return NULL; }
13 int Queue_Is_Empty(const Link_Queue a) { return a->NumEl==0; }
14 int Queue_Is_Full(const Link_Queue a) { return a->NumEl==a->Size; }
15 void Queue_put(Link_Queue a, T New_el) { if(Queue_Is_Full(a)) ERR("Queue::put: queue is full");
16     a->Items[a->P++]=New_el; a->NumEl++; if(a->P==a->Size) a->P=0; }
17 T Queue_get(Link_Queue a) { if(Queue_Is_Empty(a)) ERR("Queue::get: queue is empty");
18     return a->Items[a->P - a->NumEl-- + (a->P<=a->NumEl?a->Size:0)]; }
19
20 int main() {
21     int k,N;
22     scanf("%d",&N);
23     Link_Queue a = Queue_ini(N);
24     while(scanf("%d",&k)!=EOF) Queue_put(a,k);
25     while(!Queue_Is_Empty(a)) { k=Queue_get(a); printf("%d",k); }
26     a=Queue_free(a);
27     return 0;
28 }
```

D:\Disk_S\Lect2016\Exmp\9\mas_c\t2.exe

```

5
1 2 3 4 5 6
Queue::put: queue is full
-----
Process exited after 6.362 seconds with return value 1
Для продолжения нажмите любую клавишу . . .

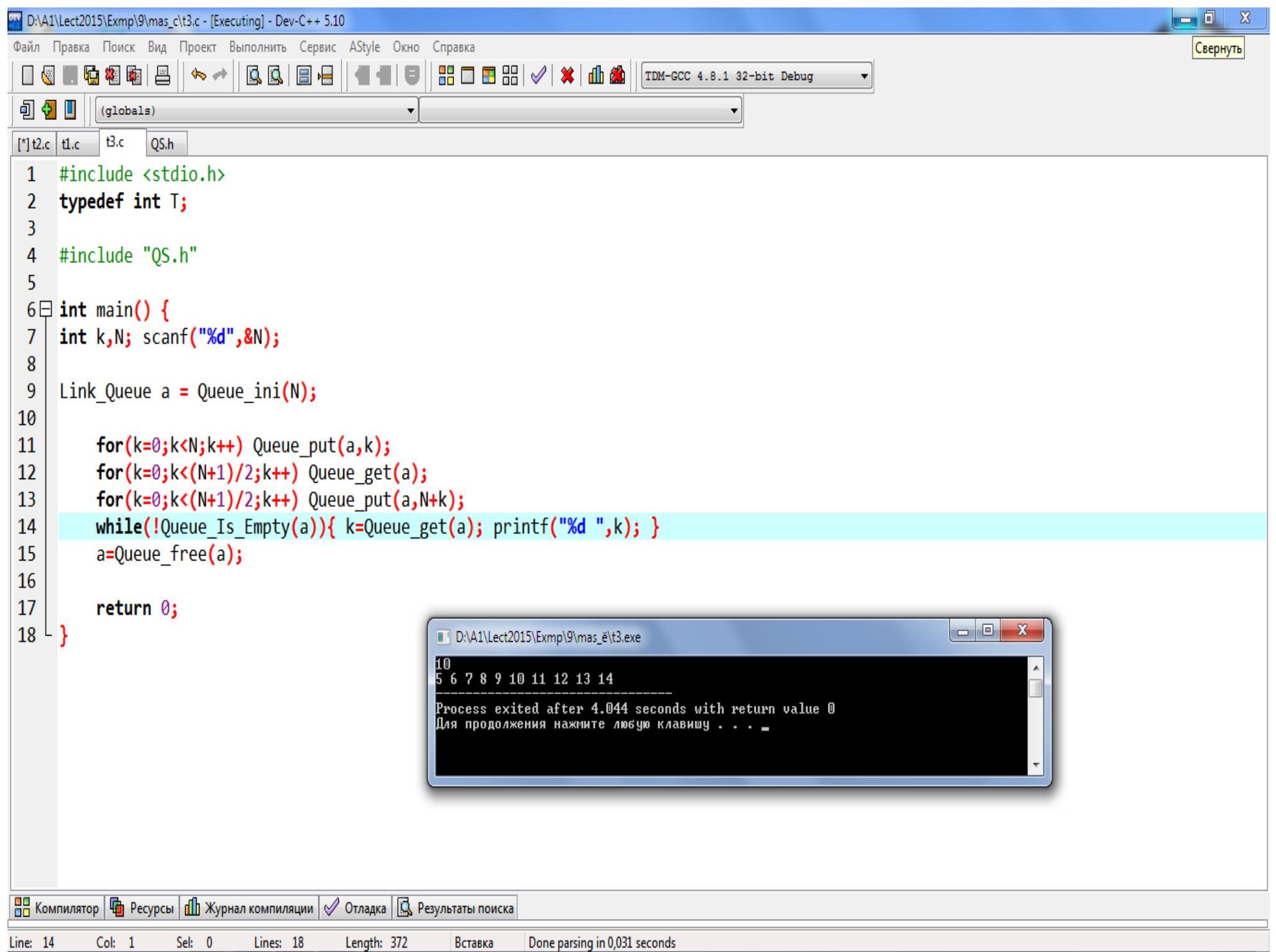
```

D:\Disk_S\Lect2016\Exmp\9\mas_c\t2.exe

```

10
1 2 3 4 5 6 7 8 9 0
^Z
1 2 3 4 5 6 7 8 9 0
-----
Process exited after 14.86 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

```



Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка



(globals)

t_char.c

```
2 #include <string.h>
3 typedef char* T;
4 #include "QS.h"
5
6 char* fget_str(char *s, FILE* a) { const int N=10;
7     char c[N]; int b=1; int ls=0,lc; if(fgets(c,N,a)==NULL) return NULL;
8     s = (char*)calloc(1,1);
9     do { lc=strlen(c); if(c[lc-1]=='\n') { b=0; c[--lc]='\0'; }
10        s = (char*)realloc(s,(ls+lc)+1);
11        strcat(s,c); } while(b && fgets(c,N,a));
12     return s;
13 }
14
15 int main() {
16     Link_Stack a;
17     Link_Queue b;
18     char* k;
19
20     a = Stack_ini(100);
21     b = Queue_ini(100);
22     while(k=fget_str(k,stdin)) { Stack_push(a,k); Queue_put(b,k); }
23     while(!Queue_Is_Empty(b)) { k=Queue_get(b); puts(k); }
24     b = Queue_free(b);
25     while(!Stack_Is_Empty(a)) { k=Stack_pop(a); puts(k); }
26     a = Stack_free(a);
27
28     return 0;
29 }
```

D:\A1\Lect2015\Exmp\9\mas_c\t_compl.c - [Executing] - Dev-C++ 5.10

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

Инструменты: (globals)

[*] t_char.c t_compl.c

```
1 #include <stdio.h>
2
3 struct complex { double Re; double Im; };
4 void print_complex(const struct complex *a) { printf("(%lf,%lf)\n", a->Re, a->Im); }
5
6 typedef struct complex T;
7
8 #include "QS.h"
9
10 int main() {
11     Link_Stack a;
12     Link_Queue b;
13     T k;
14     a = Stack_ini(100);
15     b = Queue_ini(100);
16     while (scanf("%lf%lf", &k.Re, &k.Im) != EOF) { Stack_push(a, k); Queue_put(b, k); }
17     while (!Stack_Is_Empty(a)) { k = Stack_pop(a); print_complex(&k); }
18     a = Stack_free(a); puts("");
19     while (!Queue_Is_Empty(b)) { k = Queue_get(b); print_complex(&k); }
20     b = Queue_free(b);
21
22     return 0;
23 }
```

D:\A1\Lect2015\Exmp\9\mas_c\t_compl.exe

```
1 2 3 4 5 6
^Z
<5.000000,6.000000> <3.000000,4.000000> <1.000000,2.000000>
<1.000000,2.000000> <3.000000,4.000000> <5.000000,6.000000>
-----
Process exited after 7.953 seconds with return value 0
Для продолжения нажмите любую клавишу . . .
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

Line: 23 Col: 2 Sel: 0 Lines: 23 Length: 599 Вставка Done parsing in 0,031 seconds

РЕАЛИЗАЦИЯ СТРУКТУРАМИ

Компоненты

```
#include <stdlib.h>
#include <malloc.h>

typedef struct item *Link_Item;
typedef struct list *Link_List;
typedef struct stack *Link_Stack;
typedef struct queue *Link_Queue;

void ERR(const char* s) {
    fputs(s,stderr); exit (1); }
```

Узел. Конструкция

```
typedef struct item { T node; Link_Item next; }
Item;

Link_Item Item_Create(T elem, Link_Item n) {
    Link_Item rab =
        (Link_Item)malloc(1,sizeof(Item));
    rab->node=elem; rab->next=n; return rab; }

void Item_Delete(Link_Item a) { free(a); }
```

Узел. Действия

```
T Item_get_node(const Link_Item a) {  
    return a->node; }
```

```
Link_Item Item_get_next(Link_Item a) {  
    return a->next; }
```

Стек. Конструкция и состояние

```
typedef struct stack { Link_Item Top; } Stack;

Link_Stack Stack_ini(Link_Stack a) {
    return a =
        (Link_Stack)calloc(1,sizeof(Stack)); }

Link_Stack Stack_Delete(Link_Stack a) {
    free(a); return NULL; }

int Stack_Is_Empty(const Link_Stack a) {
    return a->Top==NULL; }
```

Стек. Действия

```
void Stack_push(Link_Stack a, T elem) {  
    a->Top=Item_Create(elem,a->Top); }  
  
T Stack_pop(Link_Stack a) {  
    if(Stack_Is_Empty(a))  
        ERR("Stack::pop: empty stack");  
    Link_Item p = a->Top;  
    T rab = Item_get_node(p);  
    a->Top = Item_get_next(p);  
    Item_Delete(p);  
    return rab; }
```

Очередь. Конструкция и состояние

```
typedef struct queue {
    Link_Item Head; Link_Item Tail; } Queue;

Link_Queue Queue_ini(Link_Queue a) { return a =
    (Link_Queue)malloc(1,sizeof(struct Queue)); }

Link_Queue Queue_Delete(Link_Queue a) {
    free(a); return NULL; }

int Queue_Is_Empty(const Link_Queue a) {
    return a->Head==NULL; }
```

Очередь. Действия

```
void Queue_put(Link_Queue a, T elem) {  
    if(Queue_Is_Empty(a))  
        a->Tail=a->Head=Item_Create(elem,NULL);  
    else a->Tail =  
        (a->Tail->next = Item_Create(elem,NULL)); }
```

```
T Queue_get(Link_Queue a) {  
    if(Queue_Is_Empty(a))  
        ERR("Queue::get: queue is empty");  
    Link_Item p = a->Head;  
    T rab = Item_get_node(p);  
    a->Head = Item_get_next(p);  
    Item_Delete(p);  
    if(Queue_Is_Empty(a)) a->Tail=NULL;  
    return rab; }
```

Список. Конструкция и состояние

```
typedef struct list {  
    Link_Item Front;  
    Link_Item Back;  
} List;
```

```
Link_List List_ini(Link_List a) {  
    return a = (Link_List)malloc(1,sizeof(List));}
```

```
int List_Is_Empty(Link_List a) {  
    return a->Front==NULL; }
```

Список. Операции. Поиск

```
Link_Item List_Find(
    Link_List a,
    Link_Item *F,
    T key ) {
    if(List_Is_Empty(a)) return (*F=NULL);
    Link_Item ptr = *F = a->Front;
    if(Item_get_node(*F)==key) return NULL;
    while(( *F=Item_get_next(ptr))!=NULL) {
        if(Item_get_node(*F)==key) break; ptr=*F; }
    return ptr; }
```

Список. Операции. Добавление 1

```
void List_Insert_front(Link_List a, T elem) {  
    a->Front = Item_Create(elem,a->Front);  
    if(a->Back==NULL) a->Back = a->Front; }  
  
int List_Insert_back(Link_List a, T elem) {  
    if(List_Is_Empty(a))  
        a->Front = a->Back = Item_Create(elem,NULL);  
    else  
        a->Back =(a->Back->next =  
                  Item_Create(elem,NULL)); }
```

Список. Операции. Добавление 2

```
int List_Insert_after(
    Link_List a,
    T elem,
    T after) {
Link_Item c;
List_Find(a,&c,after);
if(c==NULL) return 0;
c->next=Item_Create(elem,Item_get_next(c));
return 1; }
```

Список. Операции. Удаление 1

```
T List_Remove_front(Link_List a) {
    if(List_Is_Empty(a))
        ERR("List::Remove_front: list is empty");
    Link_Item p=a->Front; T rab=Item_get_node(p);
    a->Front=Item_get_next(p); Item_Delete(p);
    if(a->Front==NULL) a->Back=NULL; return rab; }
```

```
int List_Remove(Link_List a, T key) {
    Link_Item b; Link_Item c;
    b=List_Find(a,&c,key);
    if(c==NULL) return 0;
    if(b==NULL) a->Front=Item_get_next(a->Front);
    else b->next=Item_get_next(c);
    Item_Delete(c); return 1; }
```

Список. Операции. Удаление 2

```
T List_Remove_back(Link_List a) {
    if(List_Is_Empty(a))
        ERR("List::Remove_back: list is empty");
    Link_Item p=a->Front;
    T rab = Item_get_node(a->Back);
    if(a->Front==a->Back)
        a->Front = a->Back = NULL;
    else { while(p->next!=a->Back) p=p->next;
            a->Back=p; p=p->next; a->Back->next=NULL; }
    Item_Delete(p); return rab; }
```

Список. Операции. Разворот

```
void List_Revers(Link_List a) {  
    Link_Item p=NULL;  
    while(a->Front!=NULL)  
        p=Item_Create(List_Remove_front(a),p);  
    a->Front=p;  
    while(Item_get_next(p)!=NULL)  
        p=Item_get_next(p);  
    a->Back=p; }
```

Список. Операции. Сортировка

```
void List_Sort (
    Link_List a,
    int (*f)(const T a, const T b)) {
Link_Item F=0;
while(a->Front!=NULL){
    Link_Item p = a->Front;
    T rab = Item_get_node(p);
    while(p=Item_get_next(p))
        if(f(p->node,rab)>0)rab=Item_get_node(p);
    F=Item_Create(rab,F); List_Remove(a,rab); }
a->Front = F;
while(F->next) F=F->next;
a->Back=F; }
```

D:\Disk_S\Lect2016\Exmp\9\src_c\SQL.h - Dev-C++ 5.11

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

TDM-GCC 4.9.2 64-bit Release

(globals)

SQL.h t_int.c t_char.c

```
1 #include <stdlib.h>
2 #include <malloc.h>
3
4 typedef struct item *Link_Item;
5 typedef struct list *Link_List;
6 typedef struct stack *Link_Stack;
7 typedef struct queue *Link_Queue;
8
9 inline void ERR(const char* s) { fputs(s,stderr); exit (1); }
10
11 typedef struct item { T node; Link_Item next; } Item;
12 Link_Item Item_Create(const T elem, const Link_Item n) {
13     Link_Item rab = (Link_Item)calloc(1,sizeof(Item));
14     rab->node=(T)elem; rab->next=n; return rab; }
15 T Item_get_node(const Link_Item a) { return a->node; }
16 Link_Item Item_get_next(const Link_Item a) { return a->next; }
17 void Item_Delete(Link_Item a) { free(a); }
18
19 typedef struct stack { Link_Item Top; } Stack;
20 Link_Stack Stack_ini(Link_Stack a) { return a = (Link_Stack)calloc(1,sizeof(Stack)); }
21 int Stack_Is_Empty(const Link_Stack a) { return a->Top==NULL; }
22 void Stack_push(Link_Stack a, const T elem) { a->Top=Item_Create(elem,a->Top); }
23 Stack Stack_pop(Link_Stack a) {
24     if(Stack_Is_Empty(a)) ERR("Stack::pop: empty stack");
25     Link_Item p = a->Top; T rab = Item_get_node(p); a->Top = Item_get_next(p);
26     Item_Delete(p); return rab; }
27 Link_Stack Stack_Delete(Link_Stack a) { free(a); return NULL; }
28
```

SQL.H (1)

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

Line: 28 Col: 1 Sel: 0 Lines: 74 Length: 4312 Вставка Done parsing in 0,015 seconds

D:\Disk_S\Lect2016\Exmp\9\src_c\SQL.h - Dev-C++ 5.11

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

TDM-GCC 4.9.2 64-bit Release

(globals)

SQL.h t_int.c

28

```
29 typedef struct queue { Link_Item Head; Link_Item Tail; } Queue;
30 Link_Queue Queue_ini(Link_Queue a) { return a = (Link_Queue)calloc(1,sizeof(Queue)); }
31 int Queue_Is_Empty(const Link_Queue a) { return a->Head==NULL; }
32 void Queue_put(Link_Queue a, const T elem) {
33     if(Queue_Is_Empty(a)) a->Tail = a->Head = Item_Create(elem,NULL);
34     else a->Tail= (a->Tail->next = Item_Create(elem,NULL));
35 T Queue_get(Link_Queue a) {
36     if(Queue_Is_Empty(a)) ERR("Queue::get: queue is empty");
37     Link_Item p = a->Head; T rab = Item_get_node(p); a->Head=Item_get_next(p);
38     Item_Delete(p); if(Queue_Is_Empty(a)) a->Tail=NULL; return rab;
39 Link_Queue Queue_Delete(Link_Queue a) { free(a); return NULL; }
40
41
```

42 **typedef struct** list { Link_Item Front; Link_Item Back; } List;
43 Link_List List_ini(Link_List a) { **return** a = (Link_List)**calloc**(1,**sizeof**(List)); }
44 **int** List_Is_Empty(**const** Link_List a) { **return** a->Front==NULL; }
45 **Link_Item** List_Find(**const** Link_List a, Link_Item *F, **const** T key) { **if**(List_Is_Empty(a)) **return** (*F=NULL);
46 Link_Item ptr = *F = a->Front; **if**(Item_get_node(*F)==key) **return** NULL;
47 **while**((*F=Item_get_next(ptr))!=NULL) { **if**(Item_get_node(*F)==key) **break**; ptr=*F; }
48 **return** ptr;
49 **void** List_Insert_front(Link_List a, **const** T elem) { a->Front = Item_Create(elem,a->Front);
50 **if**(a->Back==NULL) a->Back = a->Front; }
51 **int** List_Insert_after(Link_List a, **const** T elem, **const** T after) { Link_Item c; List_Find(a,&c,after);
52 **if**(c==NULL) **return** 0; c->next = Item_Create(elem,Item_get_next(c)); **return** 1;
53 **void** List_Insert_back(Link_List a, **const** T elem) {
54 **if**(List_Is_Empty(a)) a->Front = a->Back = Item_Create(elem,NULL);
55 **else** a->Back = (a->Back->next = Item_Create(elem,NULL)); }

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

Line: 41 Col: 1 Sel: 0 Lines: 75 Length: 4311 Вставка Done parsing in 1,953 seconds

SQL.H (2)

D:\Disk_S\Lect2016\Exmp\9\src_c\SQL.h - Dev-C++ 5.11

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

TDM-GCC 4.9.2 64-bit Release

(globals)

SQL.h

```
56 T List_Remove_front(Link_List a) { if(List_Is_Empty(a)) ERR("List::Remove_front: list is empty");  
57     Link_Item p=a->Front; T rab=Item_get_node(p); a->Front=Item_get_next(p); Item_Delete(p);  
58     if(a->Front==NULL) a->Back=NULL; return rab; }  
59 int List_Remove(Link_List a, const T key) { Link_Item b; Link_Item c; b=List_Find(a,&c,key); if(c==NULL) return 0;  
60     if(b==NULL) a->Front=Item_get_next(a->Front); else b->next=Item_get_next(c); Item_Delete(c);  
61     return 1; }  
62 T List_Remove_back(Link_List a) { if(List_Is_Empty(a)) ERR("List::Remove_back: list is empty");  
63     Link_Item p=a->Front; T rab = Item_get_node(a->Back);  
64     if(a->Front==a->Back) a->Front = a->Back = NULL;  
65     else { while(p->next!=a->Back) p=p->next; a->Back=p; p=p->next; a->Back->next=NULL; }  
66     Item_Delete(p); return rab; }  
67 void List_Revers(Link_List a) { Link_Item p=NULL; while(a->Front!=NULL) p=Item_Create(List_Remove_front(a),p);  
68     a->Front=p; while(Item_get_next(p)!=NULL) p=Item_get_next(p); a->Back=p; }  
69 void List_Sort(Link_List a, int (*f)(const T, const T)) { Link_Item F=NULL;  
70     while(a->Front!=NULL) { Link_Item p = a->Front; T rab = Item_get_node(p);  
71         while(p=Item_get_next(p)) if(f(p->node,rab)>0) rab=Item_get_node(p);  
72         F=Item_Create(rab,F); List_Remove(a,rab); }  
73     a->Front = F; while(F->next) F=F->next; a->Back=F; }  
74 Link_List List_Delete(Link_List a) { free(a); return NULL; }  
75  
76  
77  
78  
79  
80  
81  
82  
83
```

SQL.H (3)

D:\A1\Lect2015\Exmp\9\strc_c\t_int.c - Dev-C++ 5.10

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

Иконки: (globals)

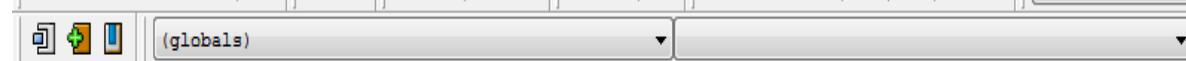
[*] t_int.c SQL.h

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <windows.h>
4
5 typedef int T;
6
7 #include "SQL.h"
8
9 void List_print(Link_List a) { Link_Item p = a->Front; if(!p) return;
10    do printf("%d ", Item_get_node(p)); while(p=Item_get_next(p)); puts(""); }
11
12 int cmp(T a, T b) { if(a>b) return 1; else if(a<b) return -1; else return 0; }
13
14 int main() {
15 Link_Stack a;
16 Link_Queue b;
17 Link_List c;
18
19 int k;
20
21 HANDLE hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);
22
23 a = Stack_ini(a);
24 b = Queue_ini(b);
25 c = List_ini(c);
26 while(scanf("%d",&k)!=EOF) { List_Insert_back(c,k); Stack_push(a,k); Queue_put(b,k); }
27 }
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

Line: 47 Col: 5 Sel: 0 Lines: 54 Length: 1539 Вставка Done parsing in 0,031 seconds

t_int.c (1)



[*] t_int.c SQL.h

```
28 SetConsoleTextAttribute(hStdOut, FOREGROUND_BLUE | FOREGROUND_INTENSITY);
29     while(!Stack_Is_Empty(a)) { k=Stack_pop(a); printf("%d ",k); } puts("");
30     a = Stack_Delete(a);
31
32 SetConsoleTextAttribute(hStdOut, FOREGROUND_GREEN | FOREGROUND_INTENSITY);
33     while(!Queue_Is_Empty(b)) { k=Queue_get(b); printf("%d ",k); } puts("");
34     b = Queue_Delete(b);
35
36 SetConsoleTextAttribute(hStdOut, FOREGROUND_RED | FOREGROUND_INTENSITY);
37     List_Insert_after(c,-1,5);
38     List_Remove(c,4);
39     List_Remove(c,1);
40     List_Remove(c,6);
41     List_print(c);
42
43 SetConsoleTextAttribute(hStdOut, FOREGROUND_GREEN | FOREGROUND_RED | FOREGROUND_INTENSITY);
44     List_Revers(c);
45     List_Insert_back(c,7);
46     List_print(c);
47 |
48 SetConsoleTextAttribute(hStdOut, FOREGROUND_GREEN | FOREGROUND_BLUE | FOREGROUND_INTENSITY);
49     List_Sort(c,cmp);
50     List_print(c);
51     c = List_Delete(c);
52
53     return 0;
54 }
```

t_int.c (2)

D:\A1\Lect2015\Exmp\9\strc_c\t_int.c - Dev-C++ 5.10

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

IDL HDB XLS XLSX XLSM XLSB XLSX5 XLSM5 XLSB5 XLSX9 XLSM9 XLSB9 XLSX10 XLSM10 XLSB10 XLSX11 XLSM11 XLSB11 XLSX12 XLSM12 XLSB12 XLSX13 XLSM13 XLSB13 XLSX14 XLSM14 XLSB14 XLSX15 XLSM15 XLSB15 XLSX16 XLSM16 XLSB16 XLSX17 XLSM17 XLSB17 XLSX18 XLSM18 XLSB18 XLSX19 XLSM19 XLSB19 XLSX20 XLSM20 XLSB20 XLSX21 XLSM21 XLSB21 XLSX22 XLSM22 XLSB22 XLSX23 XLSM23 XLSB23 XLSX24 XLSM24 XLSB24 XLSX25 XLSM25 XLSB25 XLSX26 XLSM26 XLSB26 XLSX27 XLSM27 XLSB27 XLSX28 XLSM28 XLSB28 XLSX29 XLSM29 XLSB29 XLSX30 XLSM30 XLSB30 XLSX31 XLSM31 XLSB31 XLSX32 XLSM32 XLSB32 XLSX33 XLSM33 XLSB33 XLSX34 XLSM34 XLSB34 XLSX35 XLSM35 XLSB35 XLSX36 XLSM36 XLSB36 XLSX37 XLSM37 XLSB37 XLSX38 XLSM38 XLSB38 XLSX39 XLSM39 XLSB39 XLSX40 XLSM40 XLSB40 XLSX41 XLSM41 XLSB41 XLSX42 XLSM42 XLSB42 XLSX43 XLSM43 XLSB43 XLSX44 XLSM44 XLSB44 XLSX45 XLSM45 XLSB45 XLSX46 XLSM46 XLSB46 XLSX47 XLSM47 XLSB47 XLSX48 XLSM48 XLSB48 XLSX49 XLSM49 XLSB49 XLSX50 XLSM50 XLSB50 XLSX51 XLSM51 XLSB51 XLSX52 XLSM52 XLSB52 XLSX53 XLSM53 XLSB53 XLSX54 }

(globals)

[*] t_int.c SQL.h

```
28 SetConsoleTextAttribute(hStdOut, FOREGROUND_BLUE | FOREGROUND_INTENSITY);
29     while(!Stack_Is_Empty(a)) { k=Stack_pop(a); printf("%d ",k); } puts("");
30     a = Stack_Delete(a);
31
32 SetConsoleTextAttribute(hStdOut, FOREGROUND_GREEN | FOREGROUND_INTENSITY);
33     while(!Queue_Is_Empty(b)) { k=Queue_get(b); printf("%d ",k); } puts("");
34     b = Queue_Delete(b);
35
36 SetConsoleTextAttribute(hStdOut, FOREGROUND_RED | FOREGROUND_INTENSITY);
37     List_Insert_after(c,-1,5);
38     List_Remove(c,4);
39     List_Remove(c,1);
40     List_Remove(c,6);
41     List_print(c);
42
43 SetConsoleTextAttribute(hStdOut, FOREGROUND_GREEN | FOREGROUND_INTENSITY);
44     List_Revers(c);
45     List_Insert_back(c,7);
46     List_print(c);
47
48 SetConsoleTextAttribute(hStdOut, FOREGROUND_GREEN | FOREGROUND_INTENSITY);
49     List_Sort(c,cmp);
50     List_print(c);
51     c = List_Delete(c);
52
53     return 0;
54 }
```

t_int.c (3)

D:\A1\Lect2015\Exmp\9\strc_c\t_int.exe

1 2 3 4 5 6
^Z
6 5 4 3 2 1
1 2 3 4 5 6
2 3 5 -1
-1 5 3 2 7
-1 2 3 5 7

Process exited after 7.257 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

Line: 47 Col: 5 Sel: 0 Lines: 54 Length: 1539 Вставка Done parsing in 0,031 seconds

D:\A1\Lect2015\Exmp\9\strc_c\t_char.c - Dev-C++ 5.10

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

TD-M-GCC 4.8.1 32-bit Debug

(globals)

t_char.c SQL.h

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <windows.h>
4
5 #define T char*
6
7 #include "SQL.h"
8
9 void List_print(Link_List a) { Link_Item p = a->Front; if(!p) return;
10    do printf("%s\n",Item_get_node(p)); while(p=Item_get_next(p)); }
11
12 char* fget_str(char *s,FILE* a) { const int N=10;
13    char c[N]; int b=1, int ls=0,lc;
14    if(fgets(c,N,a)==NULL) return NULL;
15    s = (char*)calloc(1,1);
16    do { lc=strlen(c); if(c[lc-1]=='\n') { b=0; c[--lc]='\0'; }
17        s = (char*)realloc(s,(ls+lc)+1); strcat(s,c);
18    } while(b && fgets(c,N,a));
19    return s;
20 }
21
22 int main() {
23    Link_Stack a;
24    Link_Queue b;
25    Link_List c;
26
27    T k;
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

Line: 51 Col: 5 Sel: 0 Lines: 57 Length: 1622 Вставка Done parsing in 0,047 seconds

t_str.c (1)

D:\A1\Lect2015\Exmp\9\strc_c\t_char.c - Dev-C++ 5.10

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

(globals)

t_char.c SQL.h

```
29 HANDLE hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);  
30  
31     a = Stack_ini(a);  
32     b = Queue_ini(b);  
33     c = List_ini(c);  
34     while(k=fget_str(k,stdin)) { List_Insert_back(c,k); Stack_push(a,k); Queue_put(b,k); }  
35  
36 SetConsoleTextAttribute(hStdOut,FOREGROUND_BLUE | FOREGROUND_INTENSITY);  
37     while(!Stack_Is_Empty(a)) { k=Stack_pop(a); printf("%s\n",k); }  
38     a = Stack_Delete(a);  
39  
40 SetConsoleTextAttribute(hStdOut,FOREGROUND_GREEN | FOREGROUND_INTENSITY);  
41     while(!Queue_Is_Empty(b)) { k=Queue_get(b); printf("%s\n",k); }  
42     b = Queue_Delete(b);  
43  
44 SetConsoleTextAttribute(hStdOut,FOREGROUND_RED | FOREGROUND_INTENSITY);  
45     List_print(c);  
46     List_Revers(c);  
47  
48 SetConsoleTextAttribute(hStdOut,FOREGROUND_GREEN | FOREGROUND_RED | FOREGROUND_INTENSITY);  
49     List_print(c);  
50     List_Sort(c,strcmp);  
51     |  
52 SetConsoleTextAttribute(hStdOut,FOREGROUND_BLUE | FOREGROUND_RED | FOREGROUND_INTENSITY);  
53     List_print(c);  
54     c = List_Delete(c);  
55
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

Line: 51 Col: 5 Sel: 0 Lines: 57 Length: 1622 Вставка Done parsing in 0,047 seconds

t_str.c (2)

C:\Windows\System32\cmd.exe

```
D:\A1\Lect2015\Exmp\9\strc_c>t_char < tst.cpp
>
    return 0;
    while( cin >> k ) cout << "k= " << k << ":" << 2.+f<2> << endl;
    cout.precision(12); cout << exp(1.) << endl;
int main() {
double f(int n) { return n > k ? 1. : n/(n+f(n+1)); }
int k;
using namespace std;
#include <cmath>
#include <iostream>
#include <iostream>
#include <cmath>
using namespace std;
int k;
double f(int n) { return n > k ? 1. : n/(n+f(n+1)); }

int main() {
    cout.precision(12); cout << exp(1.) << endl;
    while( cin >> k ) cout << "k= " << k << ":" << 2.+f<2> << endl;
    return 0;
}
#include <iostream>
#include <cmath>
using namespace std;
int k;
double f(int n) { return n > k ? 1. : n/(n+f(n+1)); }

int main() {
    cout.precision(12); cout << exp(1.) << endl;
    while( cin >> k ) cout << "k= " << k << ":" << 2.+f<2> << endl;
    return 0;
}
```

t_str.c (3)

C:\Windows\System32\cmd.exe

```
> 
    return 0;
    while( cin >> k ) cout << "k= " << k << ":" << 2.+f<2> << endl;
    cout.precision(12); cout << exp(1.) << endl;
int main() {

double f(int n) { return n > k ? 1. : n/(n+f(n+1)); }
int k;
using namespace std;
#include <cmath>
#include <iostream>

    cout.precision(12); cout << exp(1.) << endl;
    return 0;
    while( cin >> k ) cout << "k= " << k << ":" << 2.+f<2> << endl;
#include <cmath>
#include <iostream>
double f(int n) { return n > k ? 1. : n/(n+f(n+1)); }
int k;
int main() {
using namespace std;
}

D:\A1\Lect2015\Exmp\9\strc_c>
```

t_str.c (4)

Ещё понятия

- Полиморфизм
- Шаблон функции
- Инкапсуляция
- Класс
- Шаблон класса

Полиморфизм

разные функции с одинаковым именем и различными параметрами

```
int abs(int a) { return a>=0?a:-a; }
double abs(double a) { return a>=0?a:-a; }
```

Шаблоны функций

```
template<typename T> T abs(T a) {  
    return a>=0?a:-a; }
```

```
int x; ... y = abs(x); // int abs(int);
```

```
float u; ... v = abs(u); // float abs(float);
```

Инкапсуляция

```
complex a, b(1.,2.);
```

```
struct complex { float Re,Im;  
    complex() { Re = Im = 0; }  
    complex(float R, float I) { Re=R; Im=I; }  
    float abs() { return sqrt(Re*Re + Im*Im); }  
    complex operator+(complex a) {  
        return complex(Re+a.Re, Im+a.Im); }  
};  
  
ostream& operator<<(  
    ostream& a,const complex& b){  
    return a << '(' << b.Re << ',' << b.Im << ')'  
    << endl; }  
                                a.abs();  
                                a.operator+(b); // a+b;
```

Класс

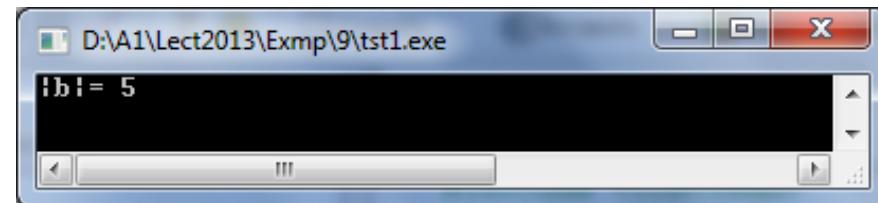
```
class complex { float Re,Im;  
public:  
complex() { Re = Im = 0; }  
complex(float R, float I) { Re=R; Im=I; }  
float abs() { return sqrt(Re*Re + Im*Im); }  
complex operator+(complex a) {  
    return complex(Re+a.Re, Im+a.Im); }  
  
friend ostream& operator<<(br/>    ostream& a, const complex& b) {  
    return a << '(' << b.Re << ',' << b.Im << ')' '  
<< endl; }  
};
```

Шаблон класса

```
template<typename T> class complex {  
    T Re, Im;  
  
public:  
    complex(T R=0, T I=0) { Re=R; Im=I; }  
    T abs() { return sqrt(Re*Re + Im*Im); }  
    complex operator+(complex a) {  
        return complex(Re+a.Re, Im+a.Im); }  
  
    friend ostream& operator<<(ostream& a, const  
        complex& b) {  
        return a << '(' << b.Re << ',' << b.Im <<  
        ')' ; }  
};
```

Использование шаблона класса

```
#include <iostream>
using namespace std;
#include "complex.h"
int main() {
    complex<double> a(1.,2.),b,c(2.,2.);
    b = a + c;
    cout << "|b|= " << b.abs() << endl;
return 0; }
```



Шаблон узла набора данных

```
template <typename T> class Item {  
    T node; Item* next;  
public:  
    Item(const T& elem, Item* n=0) {  
        node=elem; next=n; }  
    T& get_node() { return node; }  
    Item* &get_next() { return next; }  
};
```

Шаблон класса Stack

```
template <typename T> class Stack {  
    Item<T> *top; T rab;  
  
public:  
    Stack() { top = 0; }  
    void push(const T& elem) {  
        top=new Item<T>(elem,top); }  
    T& pop() {  
        if(!top) ERR("Stack::pop: empty stack");  
        rab = top->get_node(); Item<T> *p=top;  
        top = p->get_next(); delete p; return rab; }  
    bool empty() {return top == 0; }  
};
```

Шаблон класса Queue (1)

```
template <typename T> class Queue {  
    Item<T> *head, *tail; T rab;  
public:  
    Queue() { tail = head = 0; }  
  
    bool empty() { return head == 0; }  
  
    void put(const T& elem) {  
        if(tail==0) tail = head = new Item<T>(elem);  
        else tail = (tail->get_next() =  
                     new Item<T>(elem)); }  
};
```

Шаблон класса Queue (2)

```
T& get() {  
    if( !head) ERR("Queue::get: queue is empty");  
    Item<T> *p=head;  
    rab = head->get_node();  
    head=head->get_next();  
    delete p;  
    if(head==0) tail=0;  
    return rab;  
}  
}; // template <typename T> class Queue
```

Шаблон класса List (1)

```
template <typename T> class List {  
  
    Item<T> *front,*back;  T rab;  
  
    Item<T>* find(Item<T>* &F, const T& k) {  
        if(front==NULL) return (F=NULL);  
        Item<T> *ptr=F=front;  
        if(front->get_node( )==k) return 0;  
        while( (F=ptr->get_next( ))!=NULL) {  
            if(F->get_node( )==k) break; ptr=F; }  
        return ptr;  
    }  
};
```

Шаблон класса List (2)

```
public:  
List() { front = back =0; }  
bool empty() { return front==0; }  
void push_back(const T& elem) {  
    if(back==0) front = back = new Item<T>(elem);  
    else back = (back->get_next() = new  
Item<T>(elem)); }
```

Шаблон класса List (3)

```
T& pop_back() {  
    if(back==0)  
        ERR("List::pop_back: list is empty");  
    rab=back->get_node();  
    Item<T> *p=front;  
    if(front==back) front = back = 0;  
    else {  
        while(p->get_next()!=back) p=p->get_next();  
        back=p;  
        p=p->get_next();  
        back->get_next()=0;}  
    delete p;  
    return rab; }
```

Шаблон класса List (4)

```
bool insert_after(const T& k,const T& after){  
    Item<T> *c;  
    find(c,after);  
    if(c==0) return 0;  
    c->get_next() =  
        new Item<T>(k,c->get_next());  
    return 1;  
}
```

Шаблон класса List (5)

```
bool remove(const T& k) {  
    Item<T> *b, *c;  
    b=find(c,k);  
    if(c==NULL) return 0;  
    if(b==NULL) {  
        front=front->get_next(); delete (c); }  
    else {  
        b->get_next()=c->get_next(); delete(c); }  
    return 1; }  
  
void push_front(const T& elem) {  
    front = new Item<T>(elem,front);  
    if(back==0) back = front; }
```

Шаблон класса List (6)

```
T& pop_front() {  
    if(front==0)  
        ERR("List::pop_front: list is empty");  
    Item<T> *p=front; rab = p->get_node();  
    front=p->get_next(); delete p;  
    if(front==0) back=0;  
    return rab; }
```

Шаблон класса List (7)

```
void sort() { Item<T> *D=0;  
    while(front!=0) {  
        Item<T> *p=front; rab = front->get_node();  
        while(p=p->get_next())if(p->get_node()>rab)  
            rab=p->get_node();  
        D = new Item<T>(rab,D); remove(rab); }  
    front = D; back = front;  
    while(back->get_next()!=0)back=back-  
    >get_next();  
}
```

Шаблон класса List (8)

```
void revers() { Item<T> *D=0;  
    while(front!=0) D = new  
    Item<T>(pop_front(),D);  
    front = D; back=front;  
    while(back->get_next()!=0)back=back-  
    >get_next();  
}
```

Шаблон класса List (9)

```
T& operator[](int i) {
    Item<T>* p=front;
    while(i-- && p) p=p->get_next();
    if(p) return p->get_node();
    ERR("LIST::operator[]: end of List appear");}

friend ostream& operator<<
    (ostream& out, const List<T>& a) {
    Item<T>* p=a.front;
    while(p) {
        out << p->get_node() << ' ';
        p=p->get_next(); }
    return out; }

}; // template <typename T> class List
```



```
1 #include <iostream>
2 using namespace std;
3 #include "My_str.h"
4 #include "SQL.h"
5 #include <windows.h>
6
7 HANDLE hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);
8
9 int main() {
10    My_string x;
11    Stack<My_string> a;
12    Queue<My_string> b;
13    List<My_string> c;
14    while(cin >> x) { a.push(x); b.put(x); c.push_back(x); }
15    while(!b.empty()) cout << b.get() << endl;
16    SetConsoleTextAttribute(hStdOut, FOREGROUND_BLUE | FOREGROUND_INTENSITY);
17    while(!a.empty()) cout << a.pop() << endl;
18    SetConsoleTextAttribute(hStdOut, FOREGROUND_GREEN | FOREGROUND_INTENSITY);
19
20    for(int i=3; i>0; i--) { x=c[i]; c.remove(x); }
21    c.revers(); cout << c;
22    SetConsoleTextAttribute(hStdOut, FOREGROUND_RED | FOREGROUND_INTENSITY);
23    c.sort(); cout << c;
24    c.push_front(x); c.insert_after(x,x);
25    SetConsoleTextAttribute(hStdOut, FOREGROUND_BLUE | FOREGROUND_GREEN | FOREGROUND_RED | FOREGROUND_INTENSITY);
26    cout << c;
27    return 0;
28 }
```

Пример 1

D:\A1\Lect2015\Exmp\9\src\SQL.h - Dev-C++ 5.10

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

TDM-GCC 4.8.1 32-bit Debug

(globals)

SQL.h

```
1 #include <stdlib.h>
2
3 inline void ERR(const char* s) { cerr << s << endl; exit(1); }
4
5 template <typename T> class Item {
6     T node; Item* next;
7 public:
8     Item(const T &elem, Item* n=0) { node=elem; next=n; }
9     T& get_node() { return node; }
10    Item* &get_next() { return next; }
11 };
12
13 template <typename T> class Stack {
14     Item<T> *top; T rab;
15 public:
16     Stack() { top = 0; }
17     void push(const T &elem) { top = new Item<T>(elem,top); }
18     T& pop() { if(!top) ERR("Stack::pop: empty stack");
19     rab = top->get_node(); Item<T> *p=top; top = p->get_next();
20     delete p; return rab; }
21     bool empty() const {return top == 0; }
22 };
23
24 template <typename T> class Queue {
25     Item<T> *head,*tail; T rab;
26 public:
27     Queue() { tail = head = 0; }
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

D:\A1\Lect2015\Exmp\9\src\SQL.h - Dev-C++ 5.10

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

SQL.h

```
23
24 template <typename T> class Queue {
25     Item<T> *head,*tail; T rab;
26 public:
27     Queue() { tail = head =0; }
28     void put(const T& elem) {
29         if(tail==0) tail = head = new Item<T>(elem);
30         else tail = (tail->get_next() = new Item<T>(elem));
31     T& get() { if(head==0) ERR("Queue::get: queue is empty");
32     Item<T> *p=head; rab = head->get_node();
33     head=head->get_next(); delete p; if(head==0) tail=0;
34     return rab; }
35     bool empty() { return head==0; }
36 };
37
38 template <typename T> class List {
39     Item<T> *front,*back; T rab;
40     Item<T>* find(Item<T>* &F, const T& k) {
41         if(front==NULL) return (F=NULL);
42         Item<T> *ptr=F=front;
43         if(front->get_node()==k) return 0;
44         while((F=ptr->get_next())!=NULL) {
45             if(F->get_node()==k) break; ptr=F; }
46         return ptr; }
47 public:
48     List() { front = back =0; }
49     bool empty() { return front==0; }
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

Line: 1 Col: 1 Sel: 0 Lines: 94 Length: 3305 Вставка Done parsing in 0,156 seconds

D:\A1\Lect2015\Exmp\9\strc\SQL.h - Dev-C++ 5.10

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

SQL.h

```
47 public:
48 List() { front = back = 0; }
49 bool empty() { return front==0; }
50 void push_back(const T& elem) {
51     if(back==0) front = back = new Item<T>(elem);
52     else back = (back->get_next() = new Item<T>(elem)); }
53 T& pop_back() {
54     if(back==0) ERR("List::pop_back: list is empty");
55     rab=back->get_node(); Item<T> *p=front;
56     if(front==back) front = back = 0;
57     else { while(p->get_next()!=back) p=p->get_next();
58         back=p; p=p->get_next(); back->get_next()=0; }
59     delete p; return rab; }
60 bool insert_after(const T& k, const T& after) {
61     Item<T> *c; find(c,after); if(c==0) return 0;
62     c->get_next()=new Item<T>(k,c->get_next()); return 1; }
63 bool remove(const T& k) { Item<T> *b,*c; b=find(c,k);
64     if(!c) return 0;
65     if(b==NULL) { front=front->get_next(); delete (c); }
66     else { b->get_next()=c->get_next(); delete(c); }
67     return 1; }
68 void push_front(const T& elem) {
69     front = new Item<T>(elem,front);
70     if(back==0) back = front; }
71 T& pop_front() {
72     if(front==0) ERR("List::pop_front: list is empty");
73     Item<T> *p=front; rab = p->get_node();
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

Line: 46 Col: 18 Sel: 0 Lines: 94 Length: 3305 Вставка Done parsing in 0,156 seconds

D:\A1\Lect2015\Exmp\9\strc\SQL.h - Dev-C++ 5.10

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

SQL.h

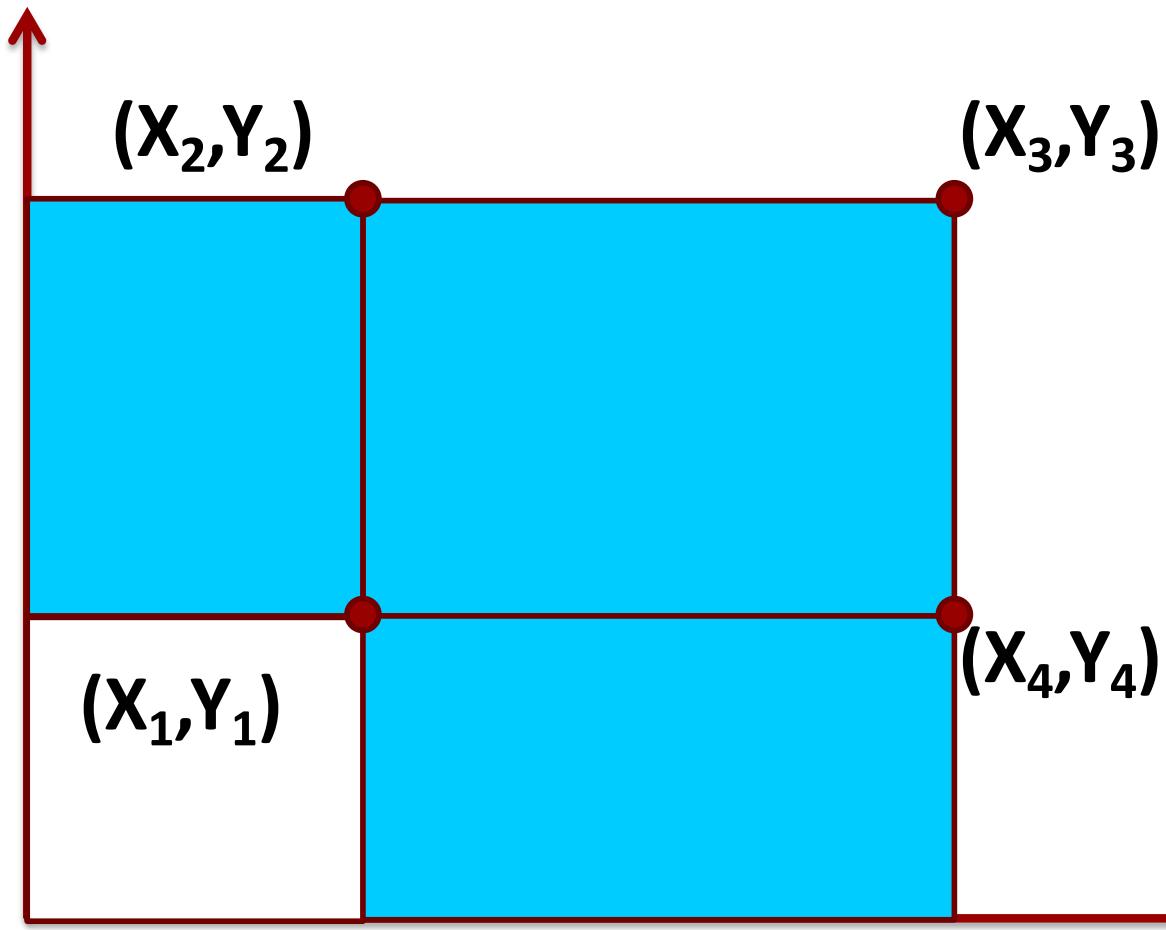
```
69     front = new Item<T>(elem,front);
70     if(back==0) back = front; }
71 T& pop_front() {
72     if(front==0) ERR("List::pop_front: list is empty");
73     Item<T> *p=front; rab = p->get_node();
74     front=p->get_next(); delete p;
75     if(front==0) back=0; return rab; }
76 void sort() { Item<T> *D=0;
77 while(front!=0) { Item<T> *p=front; rab = front->get_node();
78     while(p=p->get_next())if(p->get_node()>rab) rab=p->get_node();
79     D = new Item<T>(rab,D); remove(rab); }
80     front = D; back = front;
81     while(back->get_next()!=0) back=back->get_next(); }
82 void revers() { Item<T> *D=0;
83     while(front!=0) D = new Item<T>(pop_front(),D);
84     front = D; back=front;
85     while(back->get_next()!=0) back=back->get_next(); }
86 T& operator[](int i) { Item<T>* p=front;
87     while(i-- && p) p=p->get_next();
88     if(p) return p->get_node();
89     ERR("LIST::operator[]: end of List appear");}
90 friend ostream& operator<<(ostream& out, const List<T>& a) {
91     Item<T>* p=a.front;
92     while(p) { out << p->get_node() << endl; p=p->get_next(); }
93     return out; }
94 };
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

Line: 46 Col: 18 Sel: 0 Lines: 94 Length: 3305 Вставка Done parsing in 0,156 seconds

Пример 2: вычисление площади прямоугольной фигуры, заданной произвольным перечислением координат вершин

```
#include <iostream>
#include <fstream>
#include "SQL.h" /* Файл с описанием шаблонов
    Стека, Очереди и Списка */
using namespace std;
```



Вычисление площади фигуры

$$S = + x_3 * y_3
- x_4 * y_4
- x_2 * y_2
+ x_1 * y_1$$

```
class point {  
    int x,y;  
public:  
    point(int a=0,int b=0) { x=a; y=b; }  
    friend ostream& operator<< ( ostream& a, const point& b ) {  
        return a << '(' << b.x << ',' << b.y << ')'; }  
    friend istream& operator>> ( istream& a, point& b ) {  
        return a >> b.x >> b.y; }  
    bool operator> (const point& b) {  
        if (x==b.x) return y>b.y; else return x>b.x; }  
    bool operator!= (const point& b) { return x!=b.x || y!=b.y; }  
    bool operator== (const point& b) { return x==b.x && y==b.y; }  
    point operator! () { return point(y,x); }  
    int operator[](int i) { return i==1?x:y; }  
};
```

```
int main() {  
List<point> Sx, Sy;  
point a,b; int i,k,s=0;  
ifstream in("test_sq.dat");  
    while(in>>a) { Sx.push_front(a); Sy.push_front(!a); }  
Sx.sort(); Sy.sort(); a=b=Sx[0];  
do {      for(i=k=0;Sx[i][1]<a[1];i++);  
        while(a[1]==Sx[i][1] && Sx[i][2]<a[2]) i++,k++;  
        if(k%2) i--; else i++; a=!Sx[i]; s-=a[1]*a[2];  
        for(i=k=0;Sy[i][1]<a[1];i++);  
        while(a[1]==Sy[i][1] && Sy[i][2]<a[2]) i++,k++;  
        if(k%2) i--; else i++; a=!Sy[i]; s+=a[1]*a[2];  
} while(a!=b);  
cout << "S= " << s << endl;  
return 0; }
```

D:\A1\Lect2015\Exmp\9\str\ tst1.cpp - Dev-C++ 5.10

Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка

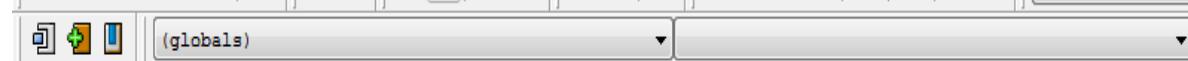
(globals)

SQL.h tst1.cpp

```
1 #include <iostream>
2 #include <fstream>
3 using namespace std;
4 #include "SQL.h" /* Файл с описанием шаблонов Стека, Очереди и Списка */
5
6
7 class point {
8     int x,y;
9 public:
10    point(int a=0,int b=0) { x=a; y=b; }
11    friend ostream& operator<< ( ostream& a,const point& b) {
12        return a << '(' << b.x << ',' << b.y << ')';
13    friend istream& operator>> ( istream& a,point& b) {
14        return a >> b.x >> b.y; }
15    bool operator> (const point& b) {
16        if (x==b.x) return y>b.y; else return x>b.x; }
17    bool operator!= (const point& b) { return x!=b.x || y!=b.y; }
18    bool operator== (const point& b) { return x==b.x && y==b.y; }
19    point operator!() { return point(y,x); }
20    int operator[](int i) { return i==1?x:y; }
21 };
22
23 int main() {
24     List<point> Sx,Sy;
25     point a,b; int i,k,s=0;
26     ifstream in("test_sq.dat");
27     while(in>>a) { Sx.push_front(a); Sy.push_front(!a); }
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска

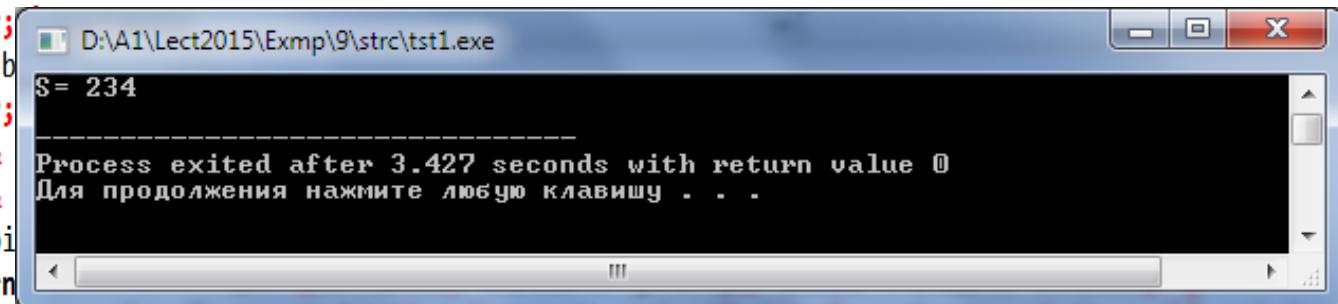
Line: 1 Col: 1 Sel: 0 Lines: 38 Length: 1357 Вставка Done parsing in 1,233 seconds



SQL.h

tst1.cpp

```
13     friend istream& operator>> ( istream& a,point& b) {
14         return a >> b.x >> b.y;
15     bool operator> (const point& b
16         if (x==b.x) return y>b.y;
17     bool operator!= (const point&
18     bool operator== (const point&
19     point operator!() { return poi
20     int operator[](int i) { return
21 };
22
23 int main() {
24 List<point> Sx,Sy;
25 point a,b; int i,k,s=0;
26 ifstream in("test_sq.dat");
27     while(in>>a) { Sx.push_front(a); Sy.push_front(!a); }
28     Sx.sort(); Sy.sort(); a=b=Sx[0];
29 do { for(i=k=0;Sx[i][1]<a[1];i++);
30         while(a[1]==Sx[i][1] && Sx[i][2]<a[2]) i++,k++;
31         if(k%2) i--; else i++; a=!Sx[i]; s-=a[1]*a[2];
32         for(i=k=0;Sy[i][1]<a[1];i++);
33         while(a[1]==Sy[i][1] && Sy[i][2]<a[2]) i++,k++;
34         if(k%2) i--; else i++; a=!Sy[i]; s+=a[1]*a[2];
35     } while(a!=b);
36     cout << "S= " << s << endl;
37     return 0;
38 }
```



Типы данных

- Базовые
- Указатели
- Составные
 - Массивы, строки,
 - Структуры, объединения, перечисления
- Абстрактные
 - Списки
 - Очереди
 - Стеки...