

Материалы презентации предназначены для размещения только для использования студентами кафедры «Компьютерные системы и технологии» НИЯУ МИФИ дневного и вечернего отделений, изучающими курс «Программирование (Алгоритмы и структуры данных)».

Публикация (размещение) данных материалов полностью или частично в электронном или печатном виде в любых других открытых или закрытых изданиях (ресурсах), а также использование их для целей, не связанных с учебным процессом в рамках курса «Программирование (Алгоритмы и структуры данных)» кафедры «КСиТ» НИЯУ МИФИ, без письменного разрешения автора запрещена.

# **Задача по теме 3 «Обработка таблиц»**

# Задача

Написать программу для работы с таблицей по запросам оператора.

Предусмотреть следующие операции:

- вставка нового элемента
- удаление элемента
- поиск элемента
- вывод содержимого таблицы

# Задача

Написать программу для работы с таблицей по запросам оператора.

Вариант а)

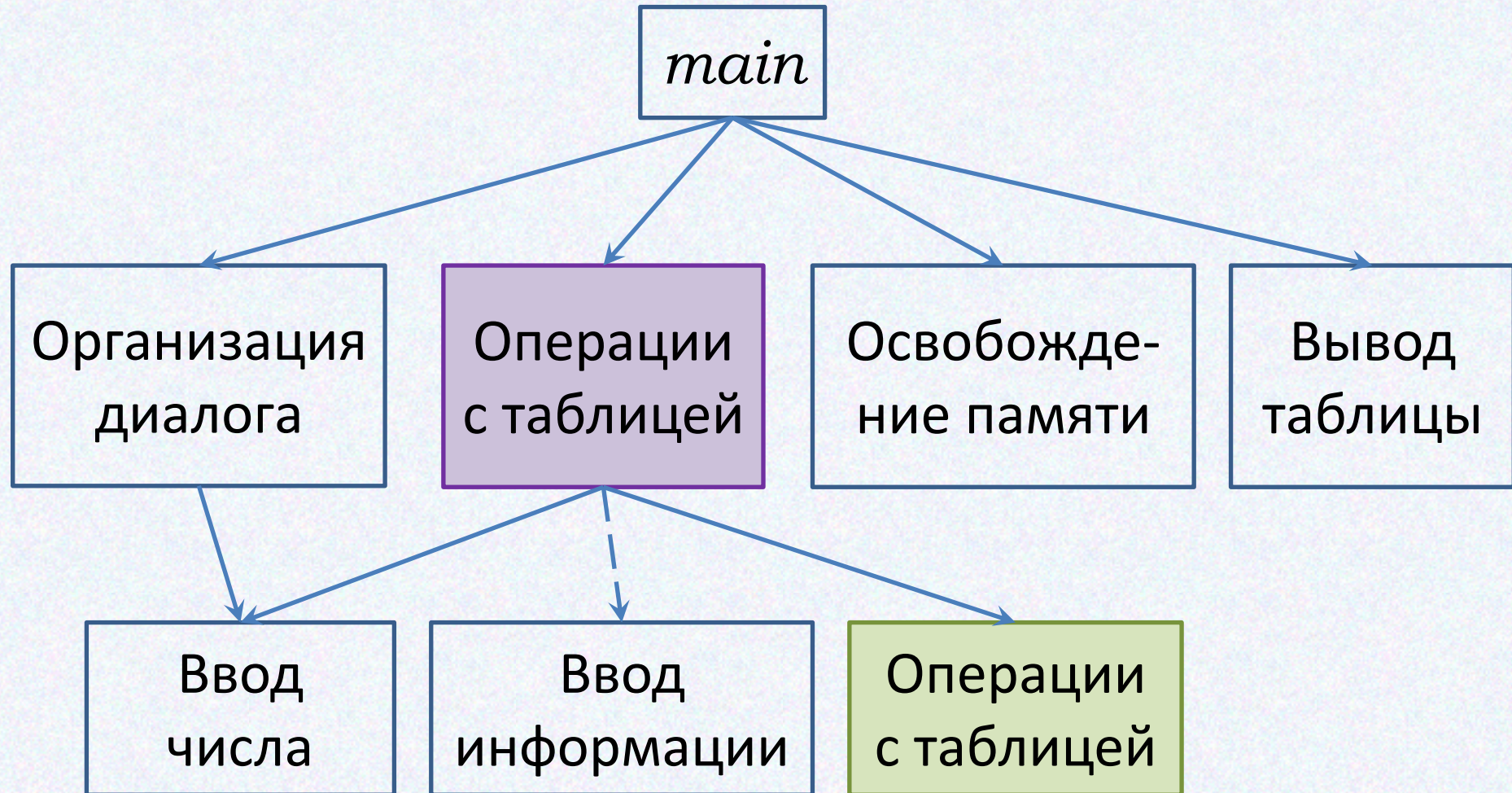
– и сама таблица, и информация, относящаяся к элементу таблицы, хранятся в основной памяти



# Структура программы

- Функция *main()*
- Функции ввода данных с необходимыми проверками
- Функции для организации диалога
- Функции для работы с таблицами

# Структура программы



# Задание таблицы

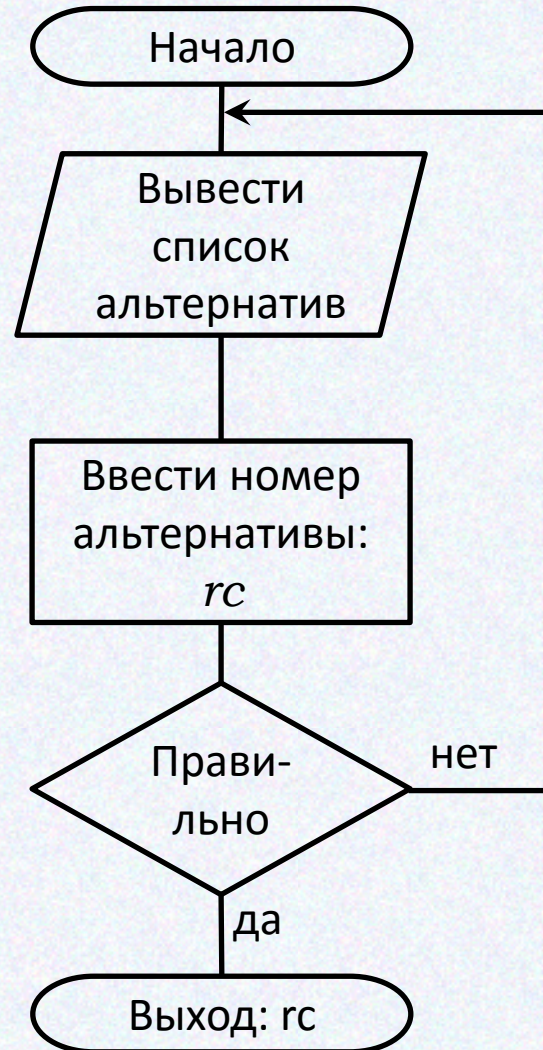
## Элемент таблицы

```
typedef struct Item {  
    мип1 key;  
    ...  
    мип2 info;  
} Item;
```

## Таблица

```
typedef struct Table {  
    ...  
} Table;
```

# Организация диалога





# Организация диалога

// Альтернативы меню для организации диалога

// Список альтернатив

```
const char *msgs[ ] = {"0. Quit", "1. Add",  
"2. Find", "3. Delete", "4. Show"};
```

// Количество альтернатив

```
const int NMsgs = sizeof(msgs) / sizeof(msgs[0]);
```

# Организация диалога

// Функция для организации диалога

// Выбирает и возвращает номер альтернативы

*int dialog(const char \*msgs[ ], int);*

// Функции работы с таблицей

// При обнаружении конца файла возвращают 0

*int D\_Add(Table \*),* // вставка элемента

*D\_Find(Table \*),* // поиск элемента

*D\_Delete(Table \*),* // удаление элемента

*D\_Show(Table \*);* // вывод таблицы

# Организация диалога

// Массив указателей на функции –  
// для реализации выбора функции.  
// Порядок перечисления функций должен совпадать  
// с порядком указания альтернатив  
// в списке альтернатив

```
int (*fptr[ ])(Table *) = {NULL, D_Add, D_Find,  
D_Delete, D_Show };
```

# Организация диалога

```
int main()
{
    Table table = {. . .};
    int rc;
    while(rc = dialog(msgs, NMsgs))
        if(!fptr[rc](&table))
            break; // обнаружен конец файла
    printf("That's all. Bye!\n");
    delTable(&table);
    return 0;
}
```



# Организация диалога

```
// Функция для выбора номера альтернативы.  
// Возвращает правильный номер альтернативы.  
// В строке может быть указан только номер  
// альтернативы; если в строке после номера  
// альтернативы есть что-то еще, весь остаток  
// строки удаляется из буфера
```

```
int dialog(const char *msgs[ ], int N)  
{  
    char *errmsg = "";  
    int rc;  
    int i, n;
```

# Организация диалога

```
do{  
    puts(errmsg);  
    errmsg = "You are wrong. Repeat, please!";  
  
    // вывод списка альтернатив  
    for(i = 0; i < N; ++i)  
        puts(msgs[i]);  
    puts("Make your choice: --> ");
```

# Организация диалога

```
n = getInt(&rc); // ввод № альтернативы  
if(n == 0)        // конец файла – конец работы  
    rc = 0;  
} while(rc < 0 || rc >= N);  
  
return rc;  
}
```

# Организация диалога

## *Примечание:*

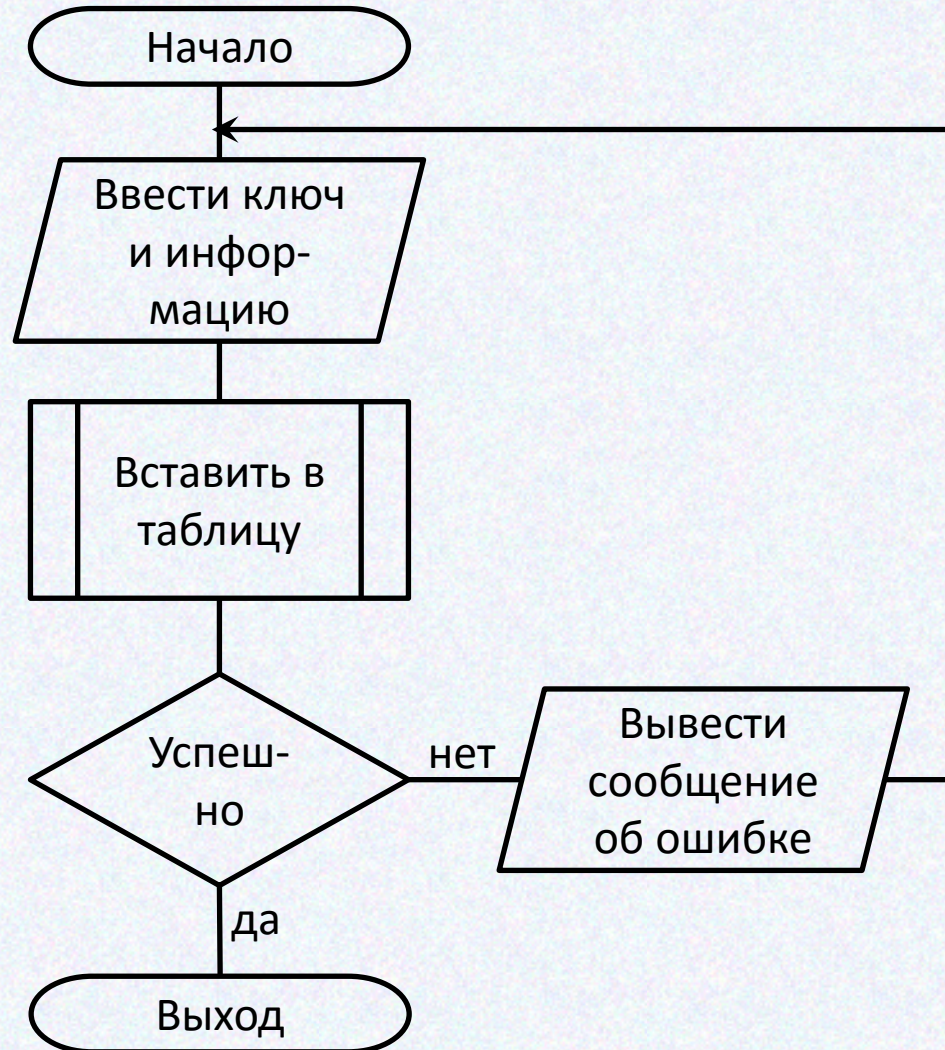
После ввода числа в буфере ввода остается, по крайней мере, символ перевода строки (`\n`)

## *Очистка буфера ввода:*

```
scanf("%*[^\\n]");    // или scanf_s("%*[^\\n]");  
scanf("%*c");         // или scanf_s("%*c");
```



# Вставка в таблицу



# Вставка в таблицу

// функции ввода

*char \*getStr();*

*int getInt(int \*);*

// вставка элемента – диалоговая функция

*int D\_Add(Table \*ptab)*

{

*int k, rc, n;*

*char \*info = NULL;*

*puts("Enter key: -->");*

*n = getInt(&k);*

*if(n == 0)*

*return 0; // обнаружен конец файла*

# Вставка в таблицу

```
puts("Enter info:\n");  
info = getStr(); // вся строка вводится целиком  
if(info == NULL)  
    return 0; // обнаружен конец файла  
rc = insert(ptab, k, info); // вставка элемента  
free(info); // в таблицу вставляется копия  
if(rc)  
    puts("Ok\n");  
else  
    printf("Duplicate key: %d\n", k);  
return 1;  
}
```