

Соловьев С.Ю.
soloviev@glossary.ru

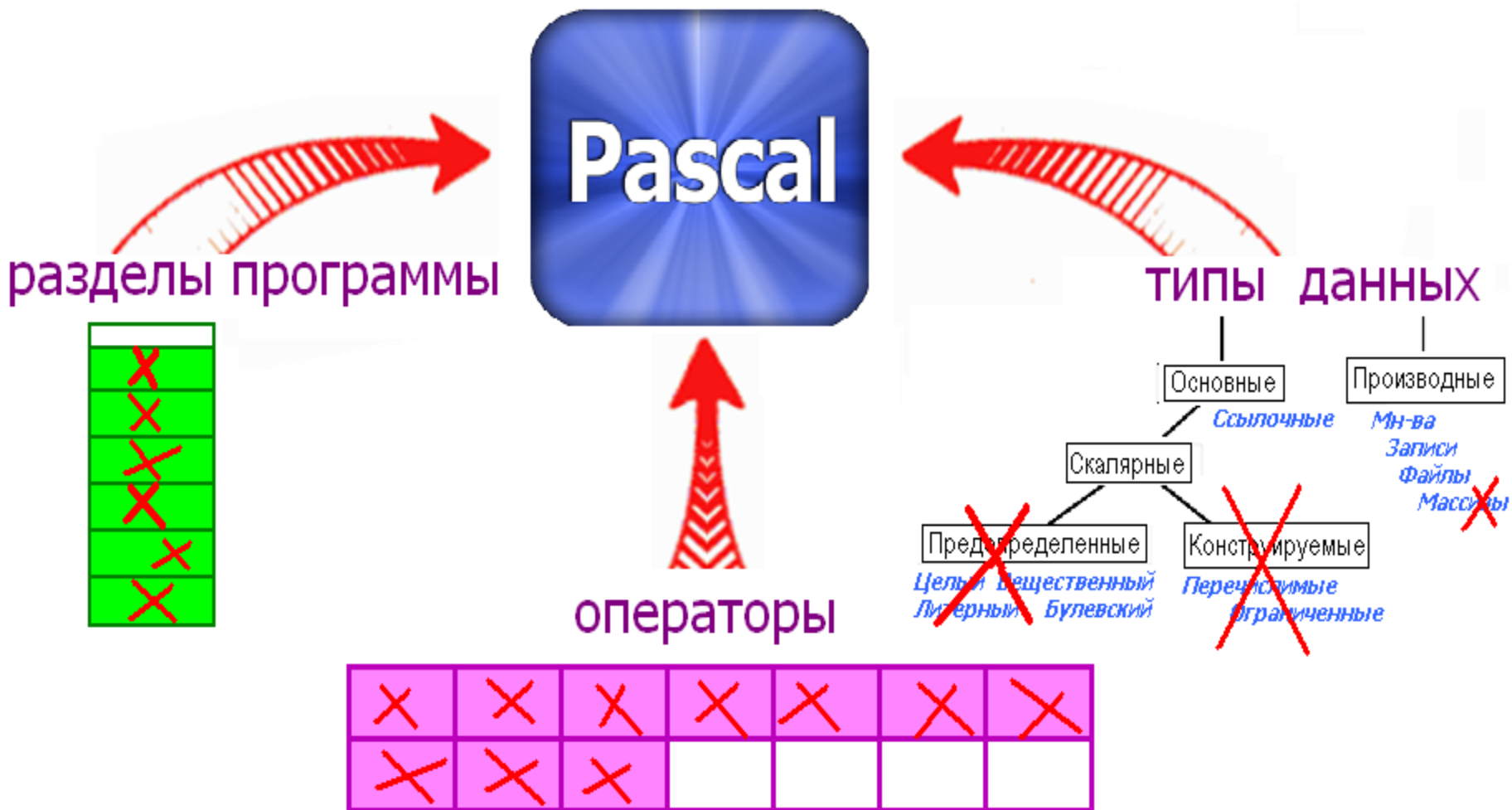
Алгоритмы и
Алгоритмические языки

www.park.glossary.ru/pascal/

Лекция No. 9

2022

Напоминание



Комбинированные типы / Записи

Запись

ИМЕНОВАННЫХ

~~Массив~~ – ~~упорядоченный~~ набор фиксированного количества ~~однотипных~~ компонент.

Запись – набор фиксированного количества именованных компонент.

полей

Поле

Имя

Тип

без ограничений

имена одной записи различны

pred('A')

<задание комбинированного типа> ::= record → <список полей> → ; → end →

<список полей> ::= <секция записи> → ; →

<секция записи> ::= <ИМЯ ПОЛЯ> → : → <ТИП> →

Селекторы

```
program Sep25a;
  type typdate = record
    YE : integer;
    MO : 1..12;
    DA : 1..31;
  end;
  var DT : typdate;      { (Б) перемен. комб. типа }

  (* Пусть <имя полной переменной> – переменная *)
  (* комбинированного типа, тогда *)
  (* <имя полной переменной>.<имя поля> *)
  (* называется селектором (исп-ся для указания полей) *)

begin
  DT.YE:=2020;
  DT.MO:=9;
  DT.DA:=25;
  writeln(DT.DA, '/', DT.MO, '/', DT.YE);      25/9/2020
end.
```

Операции с записями

(В) Операции.

<селектор> – частный случай <переменная>

⇒ *участвует в выражениях и операторах*



```
var A, B : record ...
```

```
type rctype = record ...
```

```
var A : rctype;
```

```
...
```

```
var B : rctype;
```

```
A := B ;
```

Пример

```
program Sep25b;
  type typdate = record
    YE : integer;
    MO : 1..12;
    DA : 1..31
  end;
  typstud = record
    Name : packed array [1..12] of char;
    Born  : typdate;
    Adds  : packed array [1..10] of char;
    Grou  : integer;
  end;

  var ST : typstud;
begin
  ST.Name:='Сидоров_____';

  ST.Born.YE:=2002;
  ST.Born.MO:=5;
  ST.Born.DA:=27;
  ST.Adds:='Москва_____';
  ST.Grou:=109;

end.
```

ST.Name	строка	ST.Name[I]
ST.Born	запись	
ST.Born.YE	целое	
ST.Born.MO	диапазон	
ST.Born.DA	диапазон	
ST.Adds	строка	ST.Adds[I]
ST.Grou	целое	

Пример селекторов – 2

```
program Sep25c;
type
  typstud = record Name : packed array [1..12] of char;
                 Born  : typdate;
                 Adds  : packed array [1..10] of char;
                 Grou  : integer;
               end;
var
  ST : typstud;
  I,L : integer;
begin
  ST.Name:='Сидоров_____';
  ST.Born.YE:=2002;
  ST.Born.MO:=5;
  ST.Born.DA:=27;
  ST.Adds:='Москва_____';
  ST.Grou:=109;
  L:=0;
  for I:=1 to 12 do
    if ST.Name[I] <> '_' then L:=I
  end;
end.
```

typdate = record YE : integer;
MO : 1..12;
DA : 1..31
end;

(* L := длина (ST.Name) *)

Пример – 3

+ 1 час

```
program Sep25d;
  type typtime = record    H : 0..23;
                           M,S : 0..59
                        end;
  var  T1,T2 : typtime;
begin  T1.H:=14;
       T1.M:=30;
       T1.S:=15;
       T2:=T1;                { N.B. }
       if T2.H = 23 then T2.H:=0
          else T2.H:=T2.H+1;
       writeln(T2.H:2, ':', T2.M:2, ':', T2.S:2);
end.
```

15:30:15

Назначение

```
DT.YE:=2020;
```

```
DT.MO:=9;
```

```
DT.DA:=25;
```

```
if T2.H = 23 then T2.H:=0  
                else T2.H:=T2.H+1;
```

```
ST.Born.YE:=2002;
```

```
ST.Born.MO:=5;
```

```
ST.Born.DA:=27;
```

Оператор присоединения. Введение

для упрощения записи селекторов
в простом случае

with R do S

переменная
комб. типа

оператор, в котором
поля из R обозначаются
своими именами

в общем случае

with R_1, R_2, \dots, R_n do S

эквивалентно

with R_1 do
with R_2, \dots, R_n do S

```
DT.YE:=2020;  
DT.MO:=9;  
DT.DA:=25;
```



```
with DT do begin  
  YE:=2020;  
  MO:=9;  
  DA:=25  
end;
```

```
if T2.H = 23 then T2.H:=0  
  else T2.H:=T2.H+1;  
  
with T2 do  
  if H = 23 then H:=0  
    else H:=H+1;
```

Оператор присоединения. Введение

```
with ST do begin
  Born.YE:=2002;
  Born.MO:=5;
  Born.DA:=27
end;
```

```
with ST.Born do begin
  YE:=2002;
  MO:=5;
  DA:=27
end;
```

```
ST.Born.YE:=2002;
ST.Born.MO:=5;
ST.Born.DA:=27;
```

```
with ST do
with Born do begin
  YE:=2002;
  MO:=5;
  DA:=27
end;
```

```
with ST, Born do begin
  YE:=2002;
  MO:=5;
  DA:=27
end;
```

Оператор присоединения. Определение

<оператор присоединения> ::= <заголовок> <оператор>

<заголовок> ::= with <список переменных записей> do

<список переменных записей> ::=

<переменная запись> {, <переменная запись> }



Конфликт имен

две или более переменных / полей имеют одинаковые имена

Правило:

Внутри оператора присоединения имя соответствует той записи, которая указана позже.

Оператор присоединения. Разрешение конфликта

```
program Sep26;  
  var X : record A,B : integer end;  
      Y : record A,C : char   end;  
begin  with X,Y do begin  
{ Y.A }   A:='1';  
           B:= 2;  
           C:='3';  
end;  
with Y,X do begin  
{ X.A }   A:= 1;  
           B:= 2;  
           C:='3';  
end;  
end.
```



Множественные типы

для работы с подмножествами некоторого базового типа

конечные мн-ва ✦ в квадратных скобках ✦ любой порядок эл-тов

pred('A')

<задание множественного типа> ::= set of <базовый тип>

<базовый тип> ::= <имя базового типа> | <задание базового типа>

целый ~?~
символьный
булевский

перечислимый
ограниченный

type TOB = set of boolean;

(A) Переменные множественного типа var **A** : set of 1..3;
B : TOB;

(Б) Значения множественного типа

A: [],
[1], [2], [3],
[1,2], [1,3], [2,3],
[1,2,3]

B : [],
[true], [false],
[true, false]

Множественные типы – 2

(В) Пусть BaseType – некоторый базовый тип и
var A,B,C : set of BaseType;
X : BaseType

Операции

теория множеств > $A = B \cup C \implies A := B + C < \text{Паскаль}$

теория множеств > $A = B \cap C \implies A := B * C < \text{Паскаль}$

теория множеств > $A = B \setminus C \implies A := B - C < \text{Паскаль}$

Отношения

теория множеств > $A = B \implies A = B < \text{Паскаль}$

теория множеств > $A \neq B \implies A <> B < \text{Паскаль}$

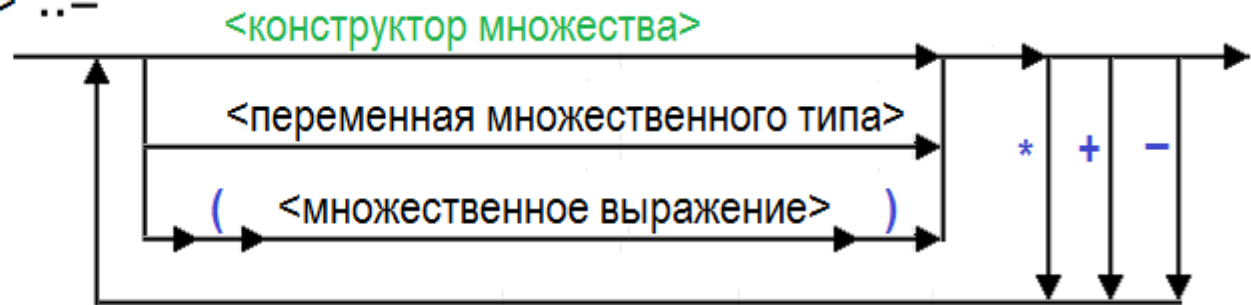
теория множеств > $A \subseteq B \implies A \leq B < \text{Паскаль}$

теория множеств > $A \supseteq B \implies A \geq B < \text{Паскаль}$

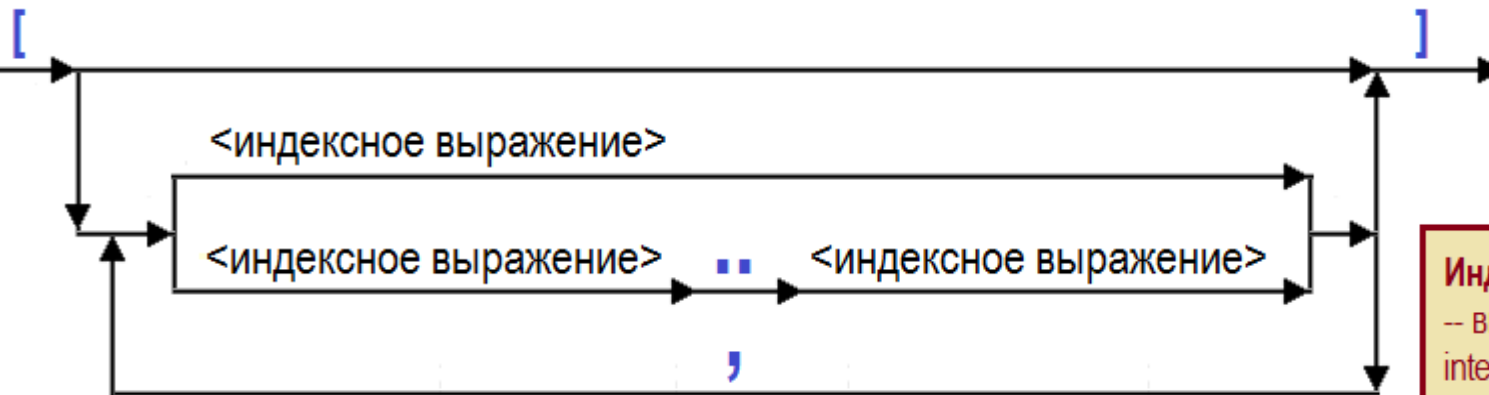
теория множеств > $X \in A \implies X \text{ in } A < \text{Паскаль}$

Выражения множественного типа

<множественное выражение> ::=



<конструктор множества> ::=



Индексное выражение
– выражение типа
integer, char, boolean или
перечислимого типа.

Пусть `var A,S : set of [1..63];
Y : 1..63;`

`A:=[1, 7, 61];
S:=[];
Y:=18;
S:=(S + [Y]) * A;`

Множественные типы. Пример

Дано: A – мн-во целых. Построить $\{n \in A \mid n - \text{нечетное}\}$

```
program Sep26a;
```

```
    type BaseInt = 1..32;
```

```
        BaseSet = set of BaseInt;
```

```
var A,X : BaseSet;
```

```
procedure OddSet(var S,T : BaseSet);
```

```
    var Y : BaseInt;
```

```
begin S:= [ ];
```

```
    for Y:= 1 to 32 do
```

```
        if Y in T then
```

```
            if odd(Y) then S:=S+[Y]
```

```
        end;
```

```
begin A:= [ 1, 4, 6, 8, 29, 30, 32 ];
```

```
    OddSet(X,A);
```

```
(* X = [1, 29] *)
```

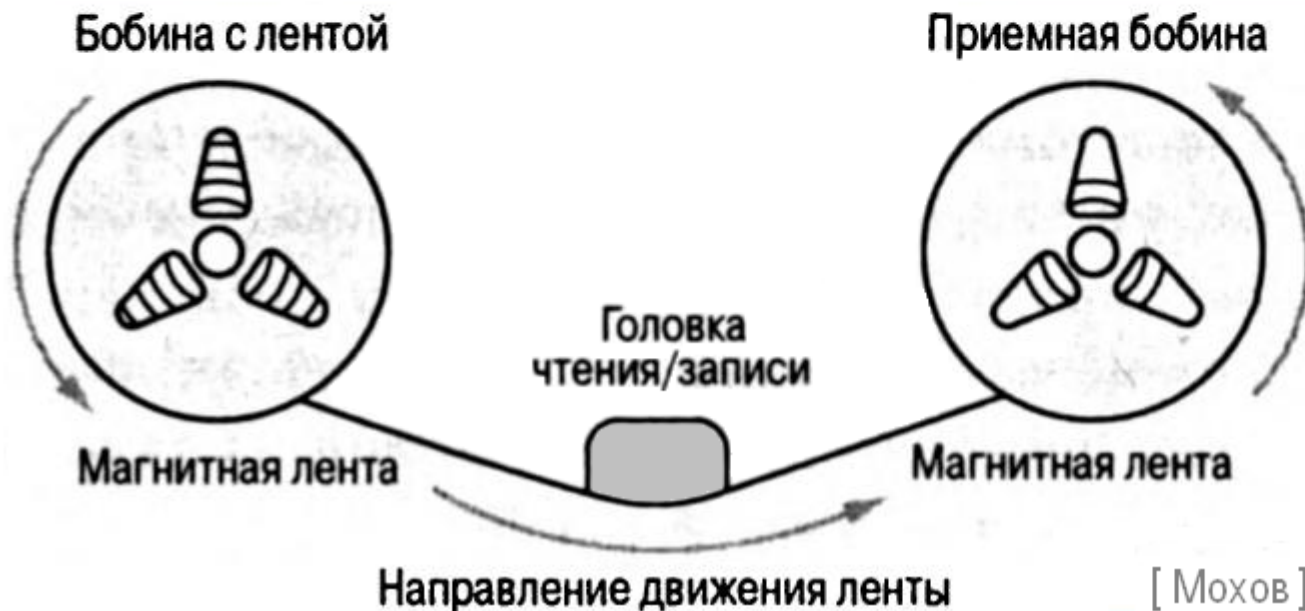
```
end.
```

Файловые типы / Файлы

Запись
~~Массив~~ – ~~упорядоченный~~ набор ~~фиксированного~~ количества ~~однотипных~~ **именованных** компонент.

Файл – **последовательность неопределенной длины**
~~Массив~~ – ~~упорядоченный набор фиксированного количества однотипных компонент.~~

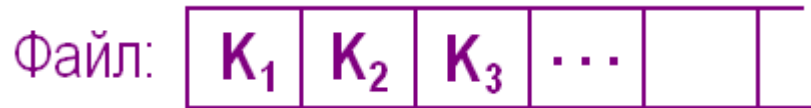
Файл – последовательность (неопределенной длины) компонент одного и того же типа.



[Мохов]

Задание файлового типа

Файл – последовательность однотипных компонент.



Доступ к **N**-му компоненту возможен только после **N-1**-го.

pred('A')

<задание файлового типа> ::= file of <тип компонент> |
packed file of <тип компонент>

<тип компонент> ::= <тип> { ≠ *файловый тип* }

(A) Файловые переменные

```
program phed(A);  
OS>phed.exe_cat.jpg  
www.site.ru/cgi-bin/  
prog.cgi?fir+sec
```

```
type TC = record R,I : real end;  
var A : file of char;  
B : file of TC;
```

внешние файлы
из списка параметров program

внутренние файлы
прочие; временные

Файловые типы / Файлы

(Б) Значение файлового типа – последовательность однотипных компонент.

(В) Операции

Пусть `var F : file of <тип>;`
`X : <тип>;`

Режимы обработки файла

чтение

reset(F)

подготовка чтения

маркер – перед первым компонентом

read(F,X)

очередной компонент
прочитать в X

маркер – на следующий компонент

eof(F)

= **true**, если чтение из F
более невозможно

запись

rewrite(F)

подготовка записи

write(F,X)

содержимое X внести в F
в качестве компонента

close(F)

Буферные переменные

Пусть `var F : file of <тип>;`

`X : <тип>;`



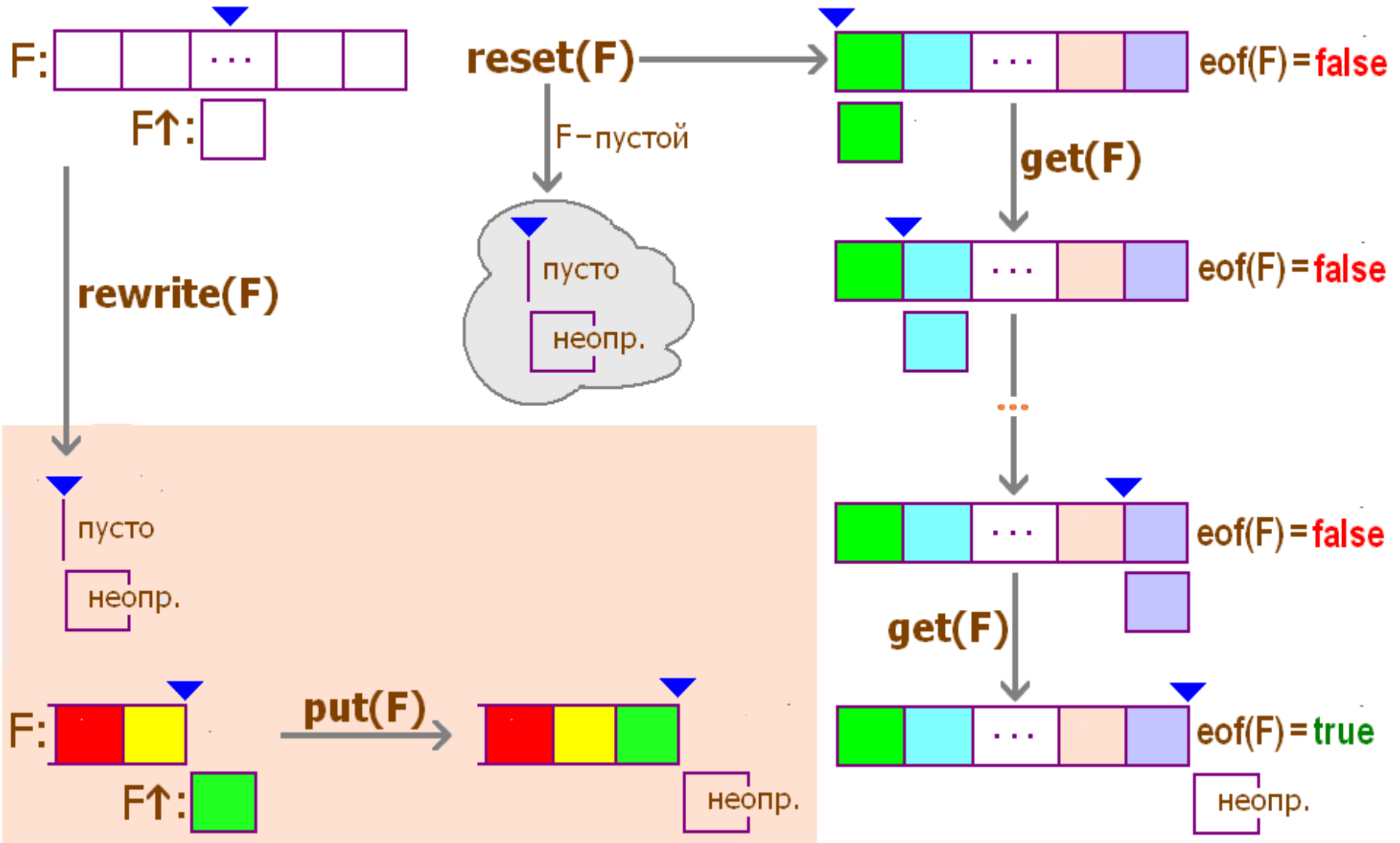
`F↑ : <тип>`

Операции:

get(F) – прочитать из файла очередной компонент;

put(F) – занести в файл очередной компонент.

put / get



Обзор лекции No.9

Комбинированные типы данных

Задание комбинированного типа

Селекторы комбинированного типа

Операции с записями

Оператор присоединения

Разрешение конфликта имен оператора присоединения

Множественные типы данных

Операции множественного типа

Отношения множественного типа

Файловые типы данных

Задание файлового типа данных

Файловые операции

Буферные переменные

Операции put и get

--- Конец лекции No. 9 ---