

# Стек

Трифон Трифонов

Структури от данни и програмиране,  
спец. Компютърни науки, 1 поток, 2017/18 г.

24–25 октомври 2017 г.



# АТД: стек

Хомогенна линейна структура с организация “последен влязъл — пръв излязъл” (LIFO)

## Операции

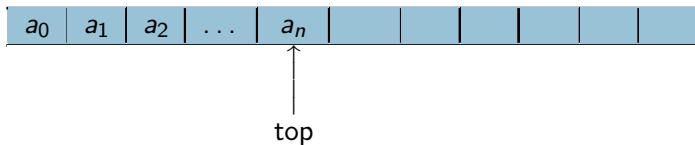
- `create()` — създаване на празен стек
- `empty()` — проверка за празнота на стек
- `push(x)` — включване на елемент на стек
- `pop()` — изключване на елемент от стек
- `peek()` — последен елемент на стека

## АТД: стек

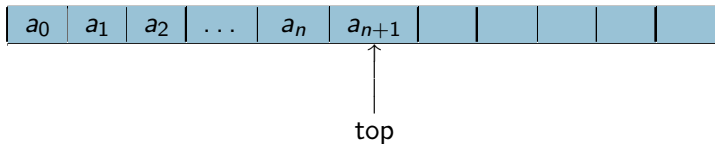
## Свойства на операциите

- ✓ • `create().empty() = true`
- ✓ • `s.push(x).empty() = false`
- ✗ • `create().peek(), create().pop() — грешка`
- ✓ • `s.push(x).peek() = x`
- ✓ • `s.push(x).pop() = s`

## Последователно представяне

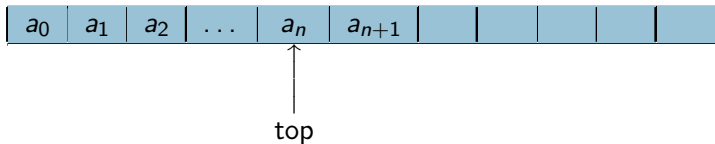


## Последователно представяне



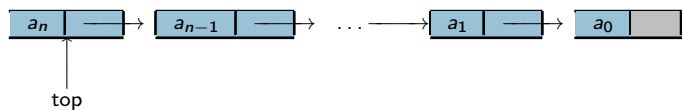
- включване на елемент (push)

# Последователно представяне

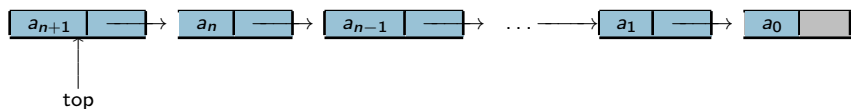


- включване на елемент (push)
- изключване на елемент (pop)

## Свързано представяне



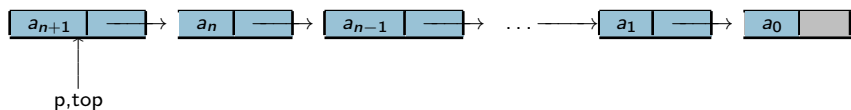
## Свързано представяне



- включване на елемент (push)

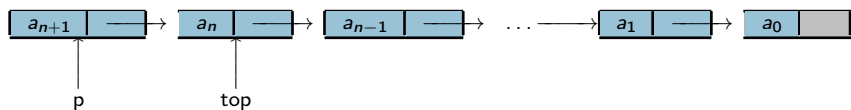


## Свързано представяне



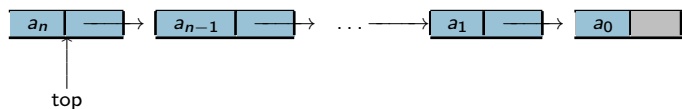
- включване на елемент (push)
- изключване на елемент (pop)

## Свързано представяне



- включване на елемент (push)
- изключване на елемент (pop)

## Свързано представяне

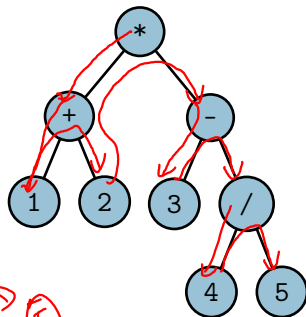


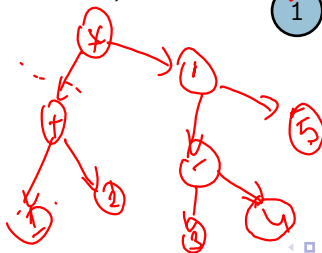
- включване на елемент (push)
- изключване на елемент (pop)

## Обратен полски запис

$$\langle \text{израз} \rangle ::= \langle \text{число} \rangle \mid (\langle \text{израз} \rangle \langle \text{оператор} \rangle \langle \text{израз} \rangle)$$

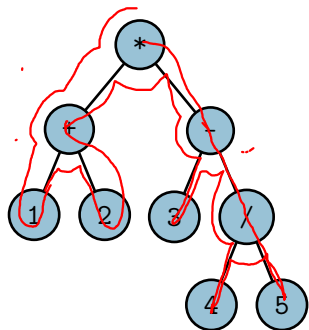
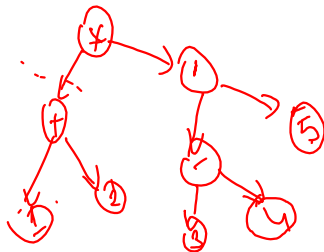
- инфиксен запис:  
 $(1+2) * (3-4) / 5$
- префиксен (полски) запис:  
 $*+12-3/45$
- постфиксен (обратен полски) запис  
 $12+345/-*$



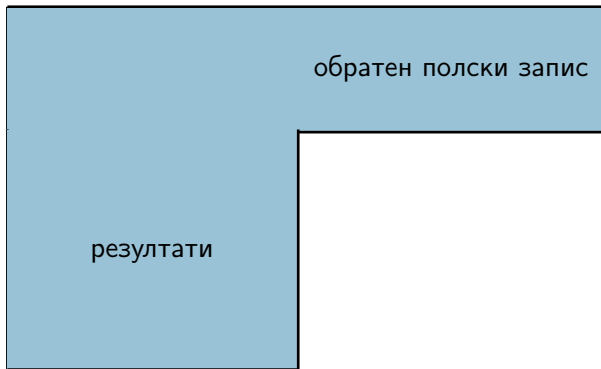
$$*+12/-345$$


# Обратен полски запис

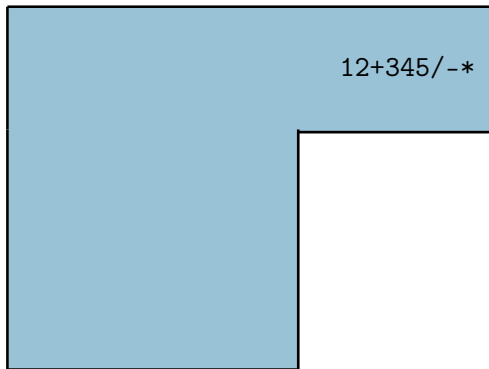
- инфиксен запис:  
 $(1+2)*(3-4/5)$
- префиксен (полски) запис:  
 $*+12-3/45$
- постфиксен (обратен полски) запис  
 $12+345/-*$



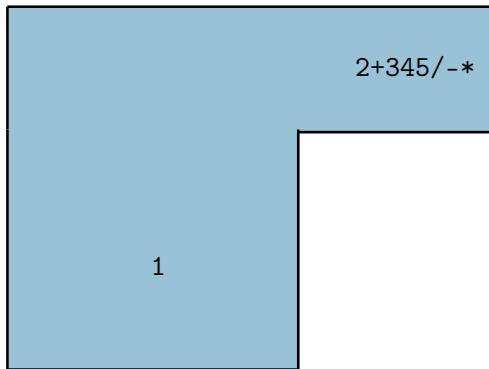
# Пресмятане на израз в обратен полски запис



# Пример: Пресмятане на израз в обратен полски запис

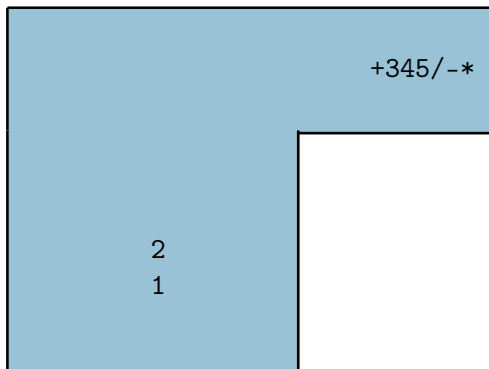


# Пример: Пресмятане на израз в обратен полски запис

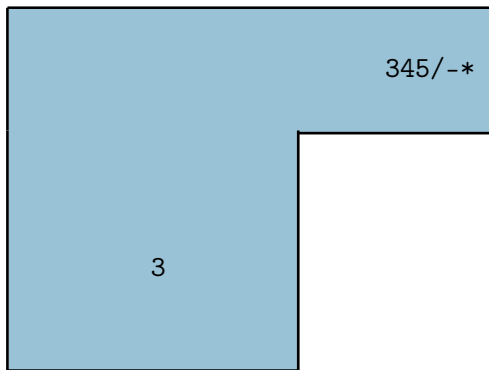




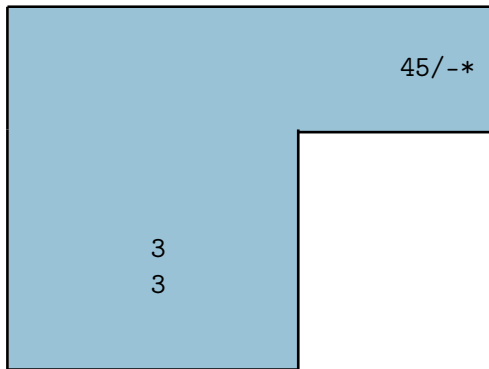
# Пример: Пресмятане на израз в обратен полски запис



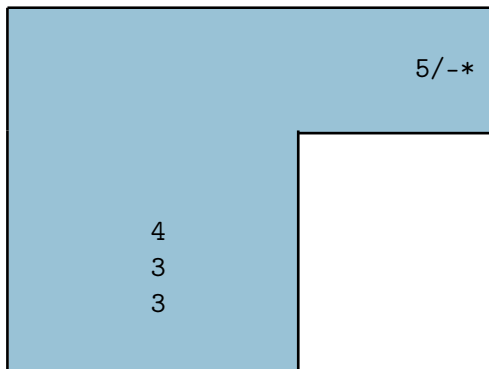
# Пример: Пресмятане на израз в обратен полски запис



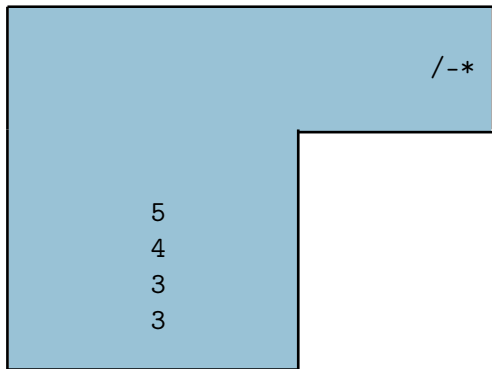
# Пример: Пресмятане на израз в обратен полски запис



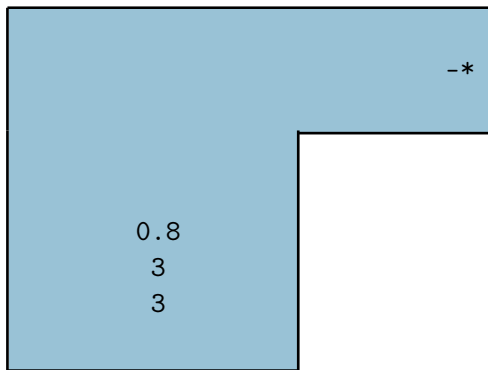
# Пример: Пресмятане на израз в обратен полски запис



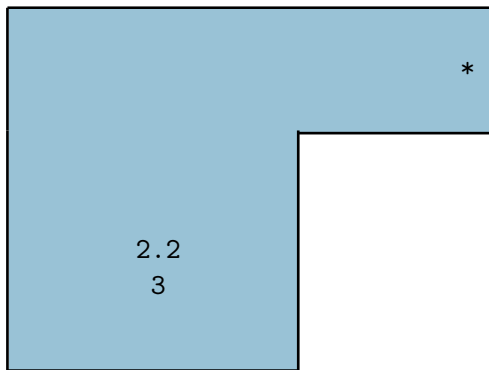
## Пример: Пресмятане на израз в обратен полски запис



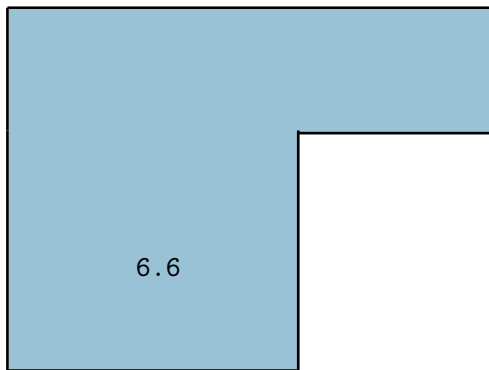
# Пример: Пресмятане на израз в обратен полски запис



# Пример: Пресмятане на израз в обратен полски запис

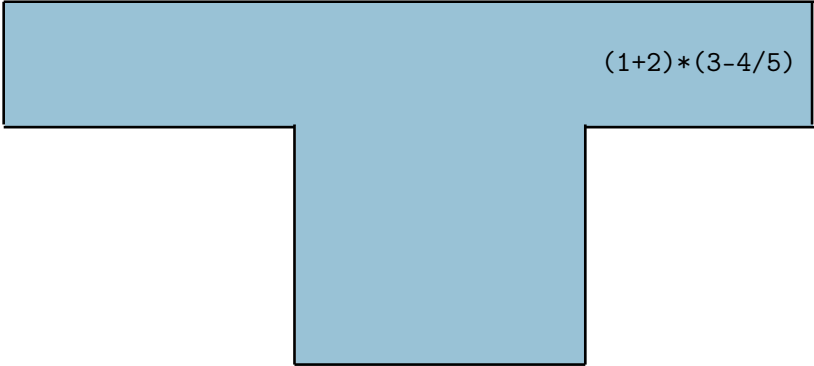


# Пример: Пресмятане на израз в обратен полски запис



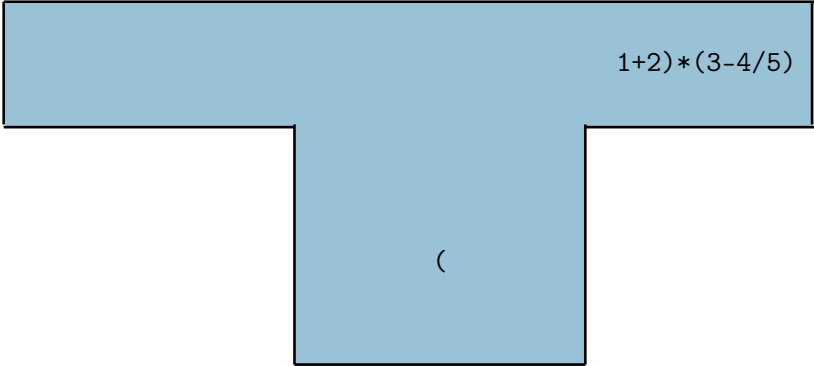


# Пример: Преобразуване в обратен полски запис



$(1+2)*(3-4/5)$

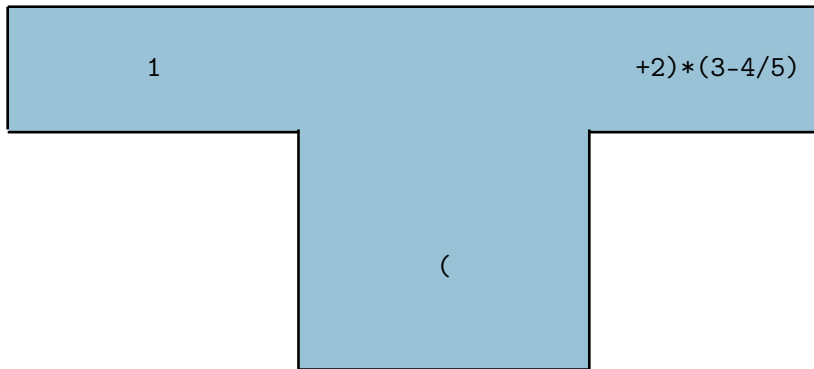
# Пример: Преобразуване в обратен полски запис



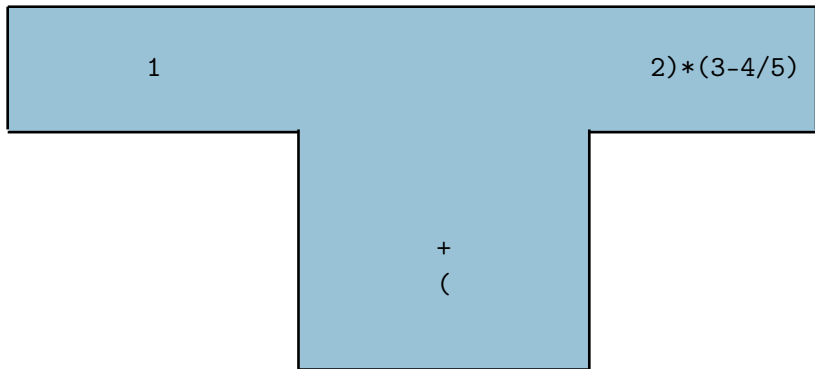
$1+2)*(3-4/5)$

(

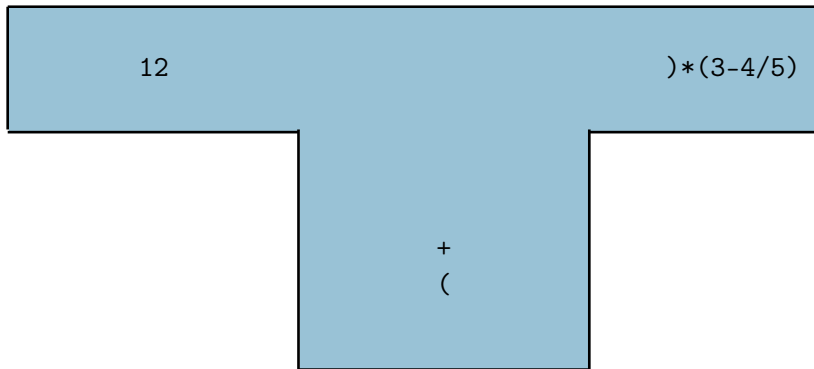
# Пример: Преобразуване в обратен полски запис



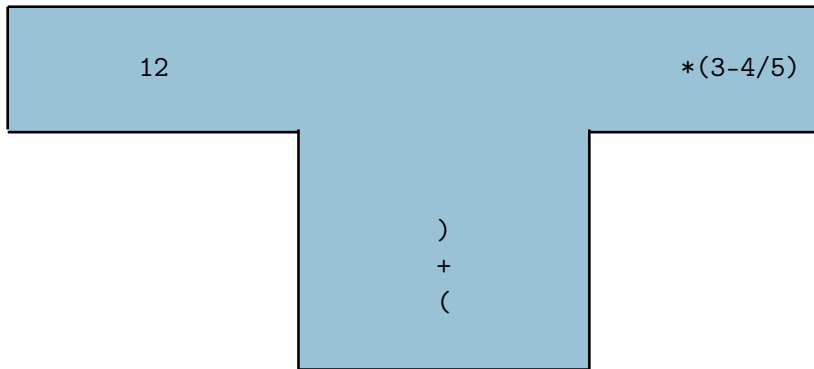
# Пример: Преобразуване в обратен полски запис



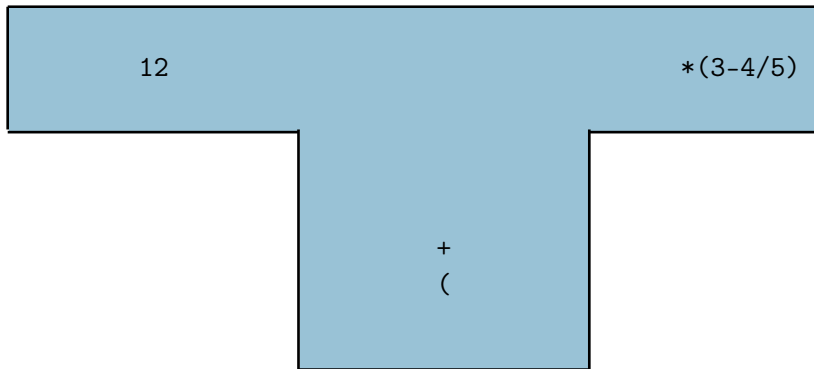
# Пример: Преобразуване в обратен полски запис



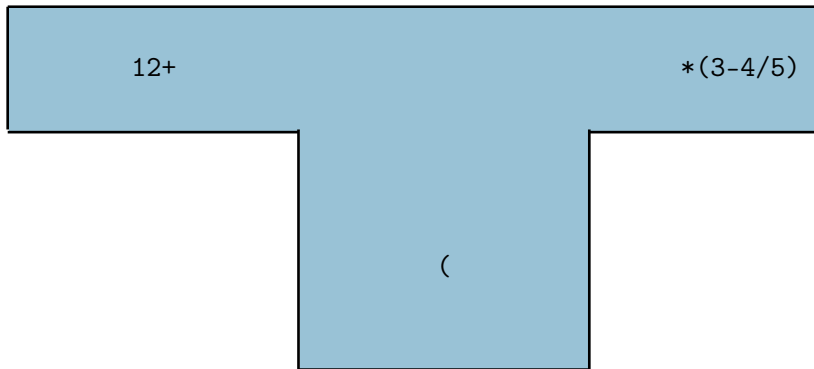
# Пример: Преобразуване в обратен полски запис



# Пример: Преобразуване в обратен полски запис

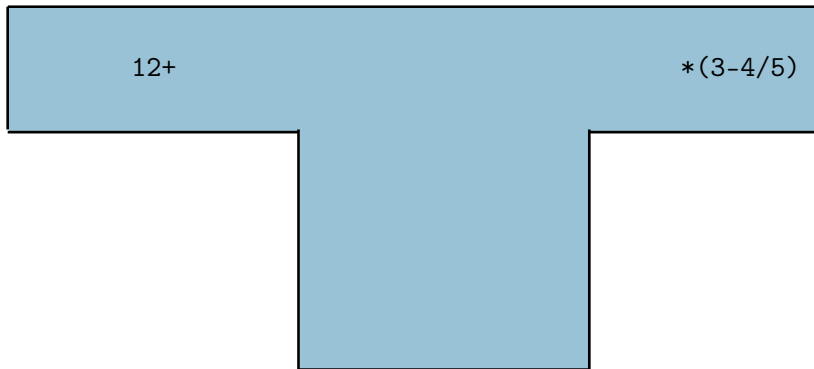


# Пример: Преобразуване в обратен полски запис

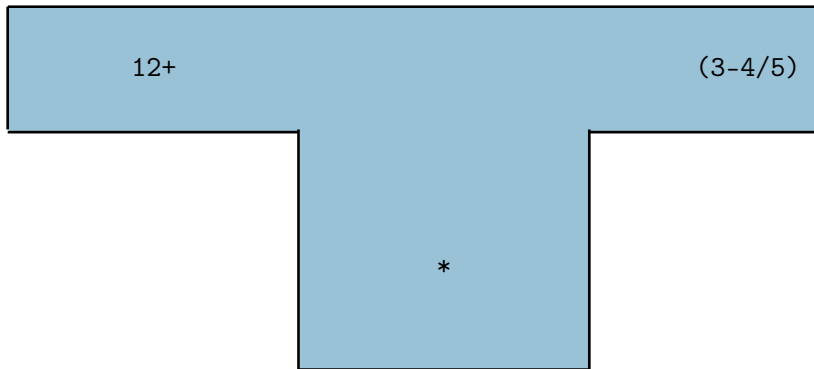




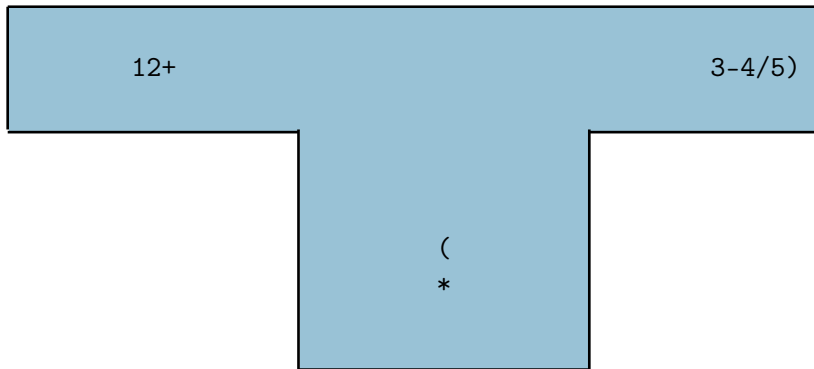
# Пример: Преобразуване в обратен полски запис



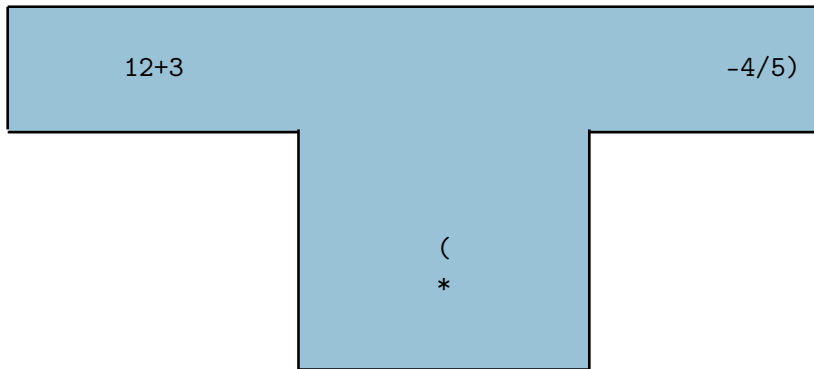
# Пример: Преобразуване в обратен полски запис



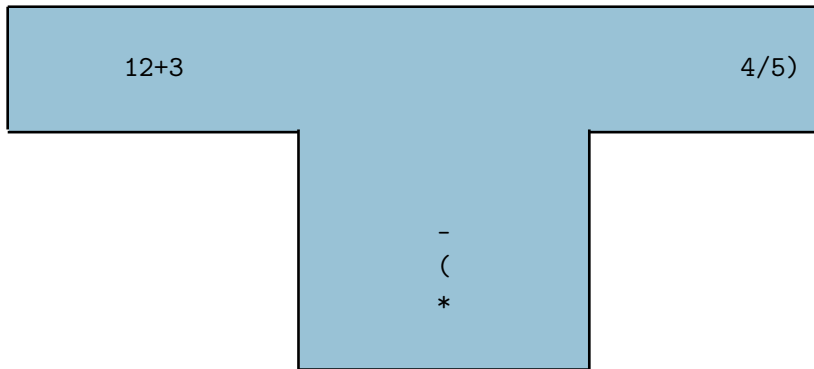
# Пример: Преобразуване в обратен полски запис



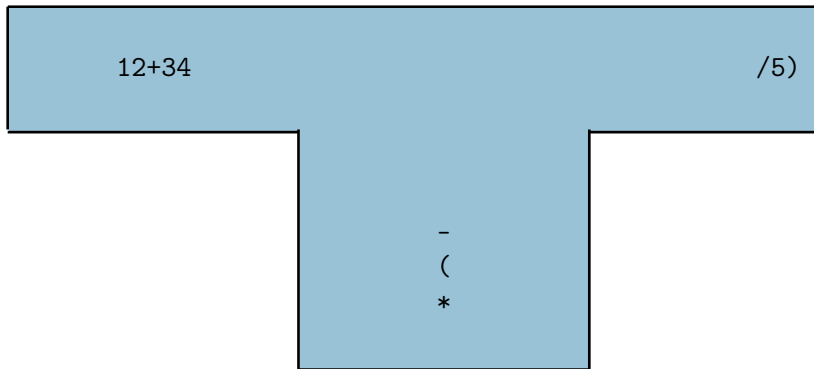
# Пример: Преобразуване в обратен полски запис



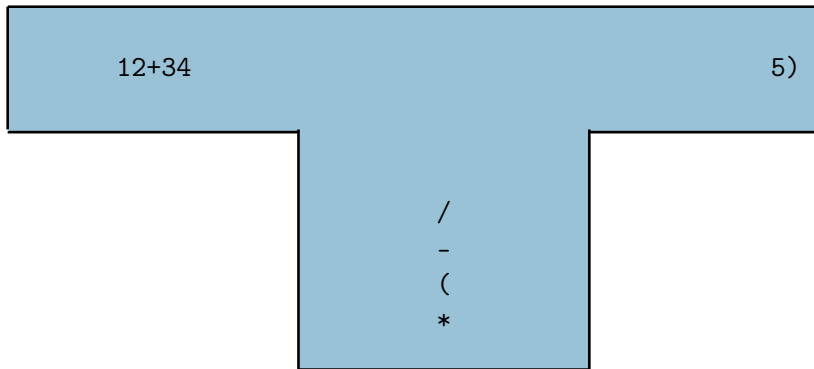
# Пример: Преобразуване в обратен полски запис



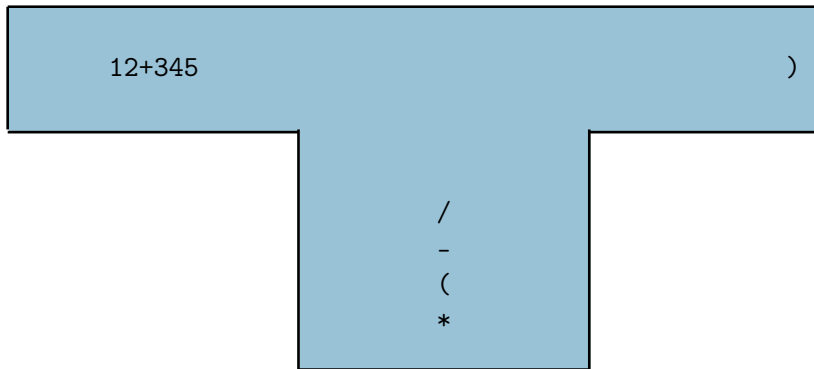
# Пример: Преобразуване в обратен полски запис



# Пример: Преобразуване в обратен полски запис



# Пример: Преобразуване в обратен полски запис





# Пример: Преобразуване в обратен полски запис

12+345

)

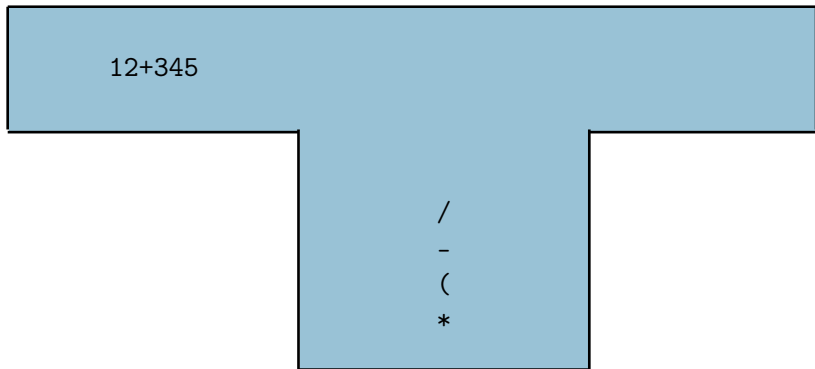
/

-

(

\*

# Пример: Преобразуване в обратен полски запис



# Пример: Преобразуване в обратен полски запис

12+345/

-

(

\*

# Пример: Преобразуване в обратен полски запис

12+345/-

(

\*

# Пример: Преобразуване в обратен полски запис

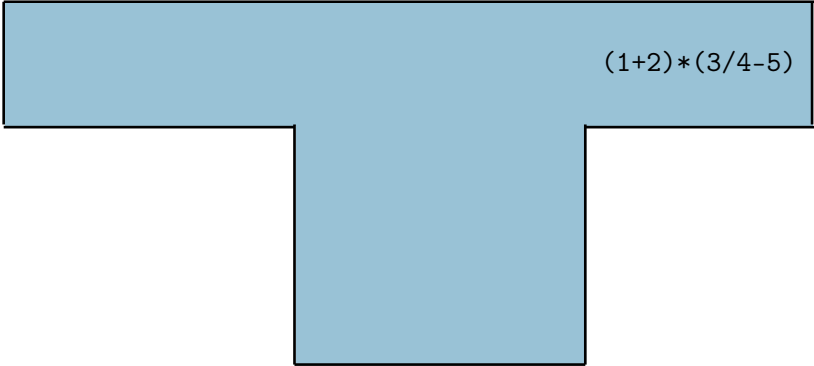
12+345/-

\*

# Пример: Преобразуване в обратен полски запис

12+345/-\*

## Пример 2: Преобразуване в обратен полски запис



$(1+2)*(3/4-5)$

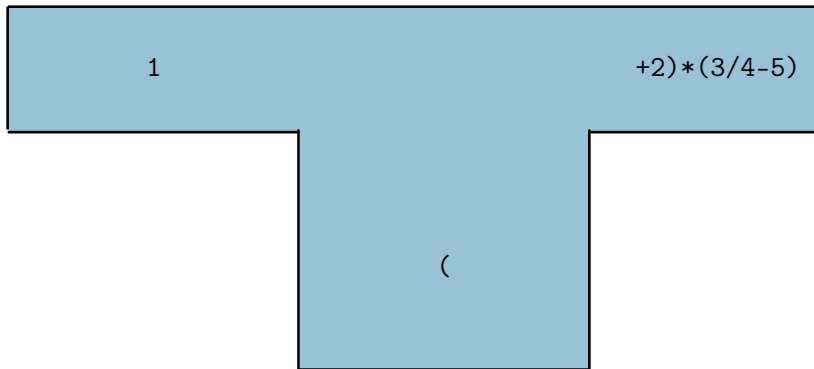
## Пример 2: Преобразуване в обратен полски запис

$1+2)*(3/4-5)$

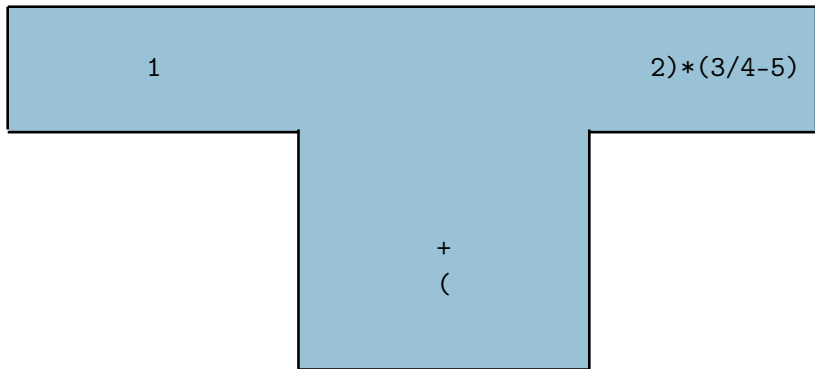
(



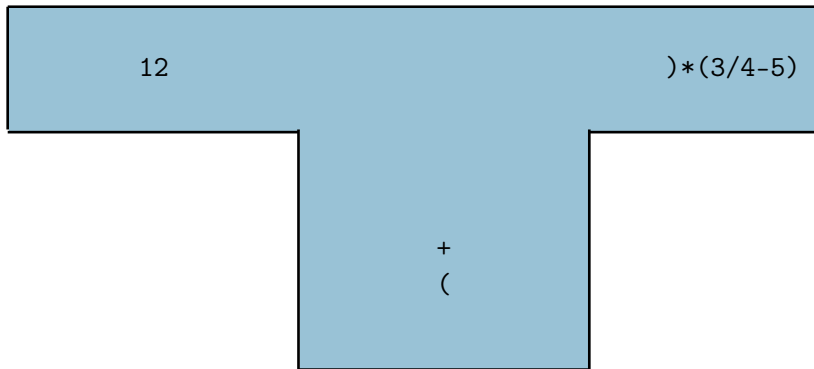
## Пример 2: Преобразуване в обратен полски запис



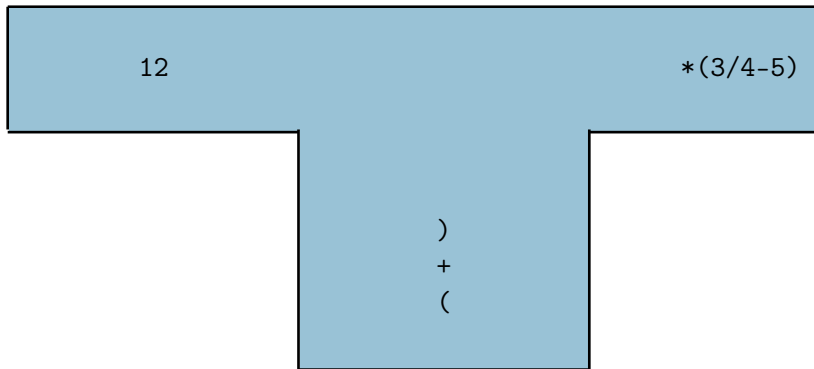
## Пример 2: Преобразуване в обратен полски запис



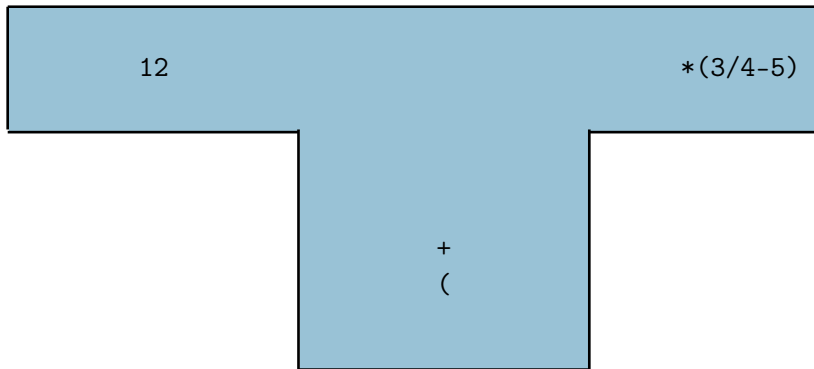
## Пример 2: Преобразуване в обратен полски запис



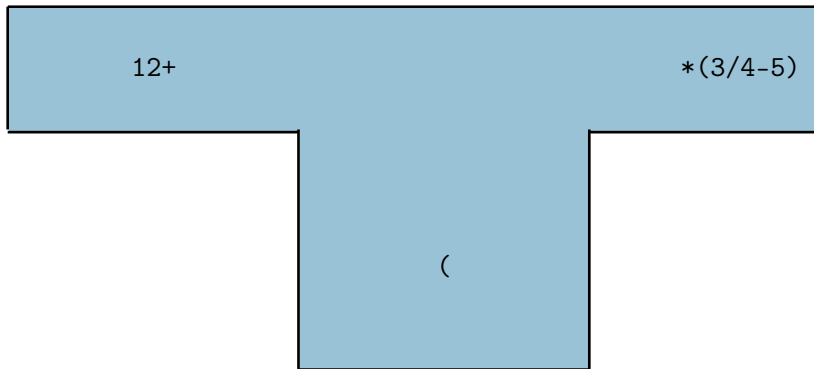
## Пример 2: Преобразуване в обратен полски запис



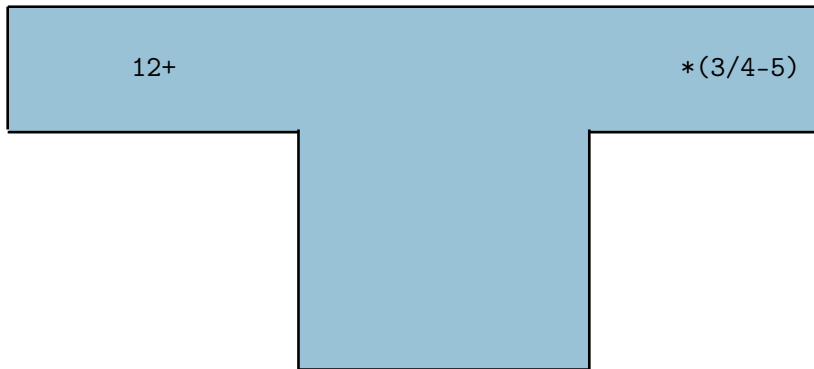
## Пример 2: Преобразуване в обратен полски запис



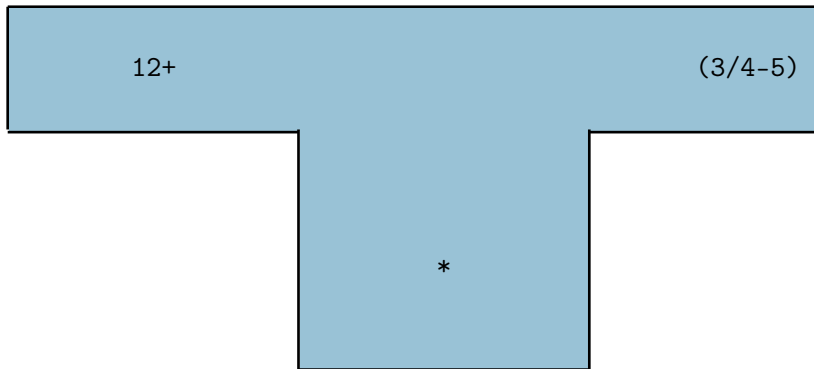
## Пример 2: Преобразуване в обратен полски запис



## Пример 2: Преобразуване в обратен полски запис

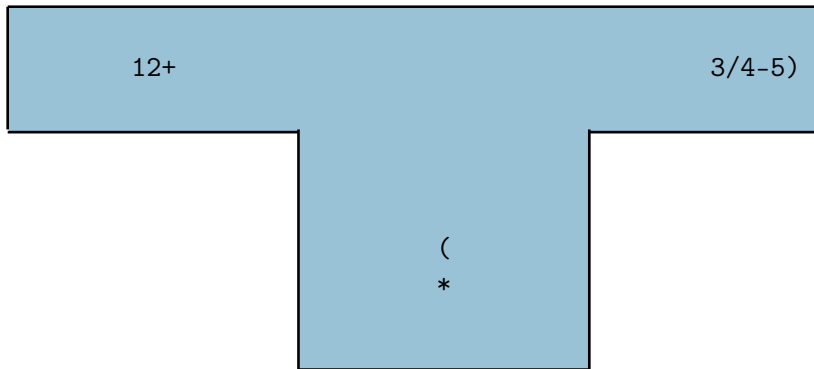


## Пример 2: Преобразуване в обратен полски запис

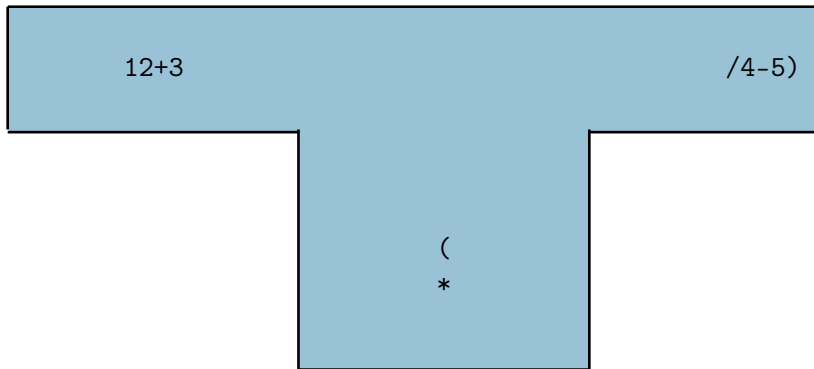




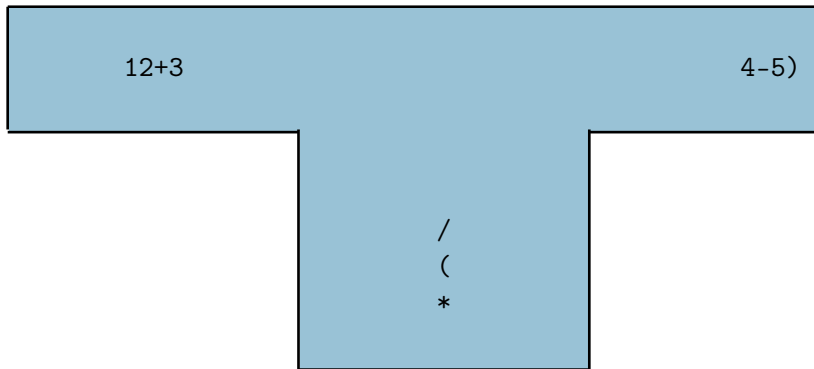
## Пример 2: Преобразуване в обратен полски запис



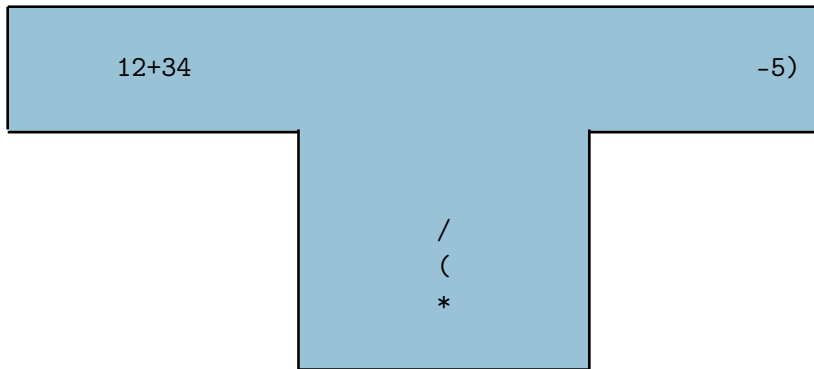
## Пример 2: Преобразуване в обратен полски запис



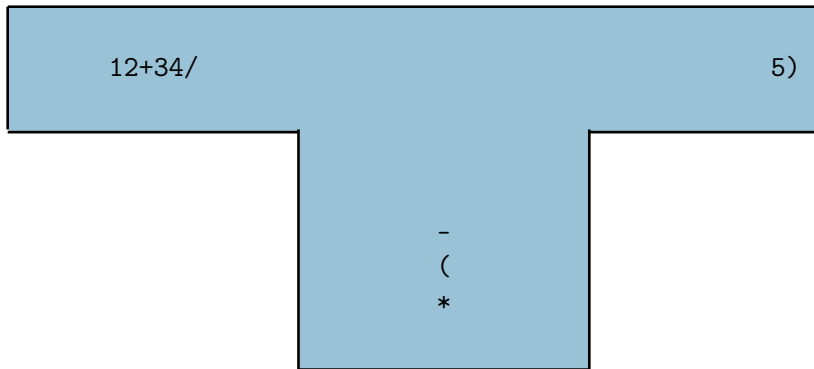
## Пример 2: Преобразуване в обратен полски запис



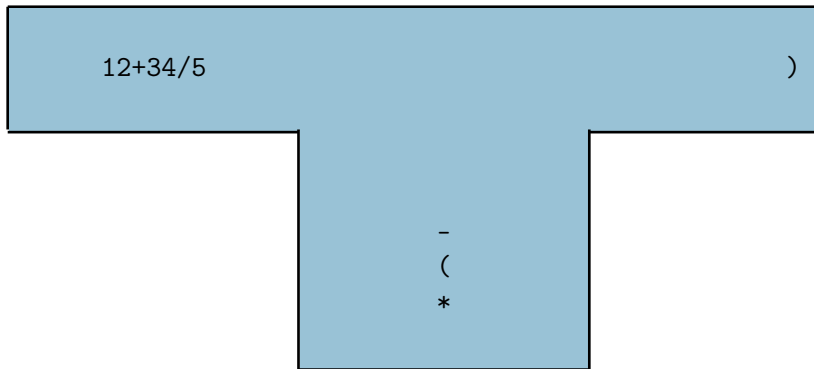
## Пример 2: Преобразуване в обратен полски запис



## Пример 2: Преобразуване в обратен полски запис



## Пример 2: Преобразуване в обратен полски запис



## Пример 2: Преобразуване в обратен полски запис

12+34/5

)

-

(

\*

## Пример 2: Преобразуване в обратен полски запис

12+34/5

-

(

\*



## Пример 2: Преобразуване в обратен полски запис

12+34/5-

(

\*

## Пример 2: Преобразуване в обратен полски запис

12+34/5-

\*

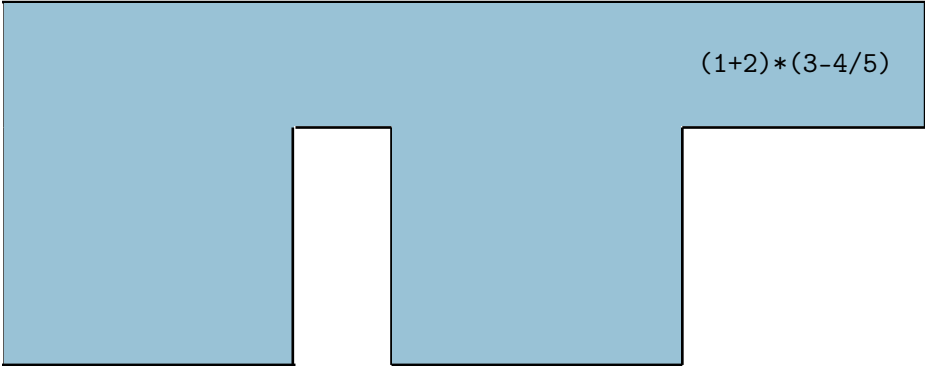
## Пример 2: Преобразуване в обратен полски запис

12+34/5-\*

# Директно пресмятане на израз



# Пример: Директно пресмятане на израз

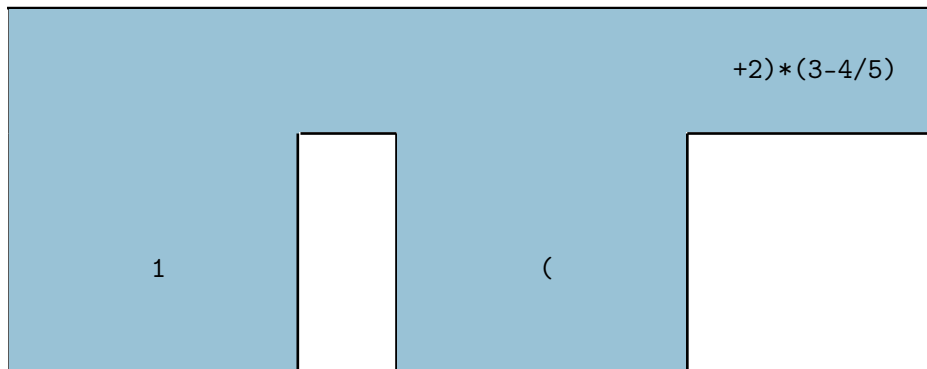

$$(1+2)*(3-4/5)$$

# Пример: Директно пресмятане на израз

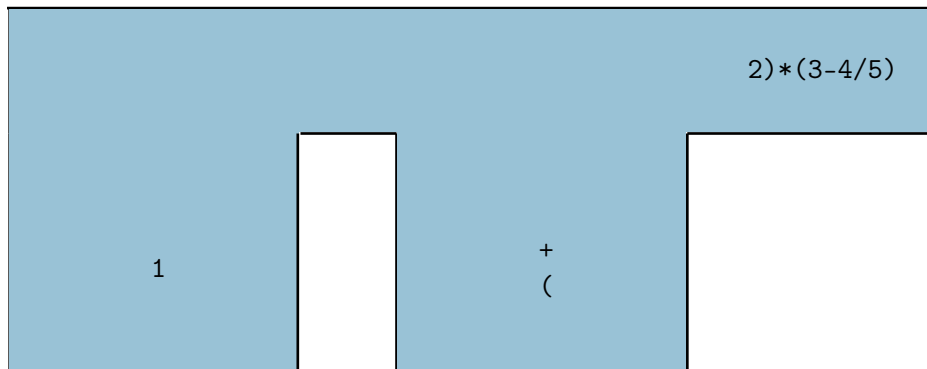
$1+2)*(3-4/5)$

(

# Пример: Директно пресмятане на израз

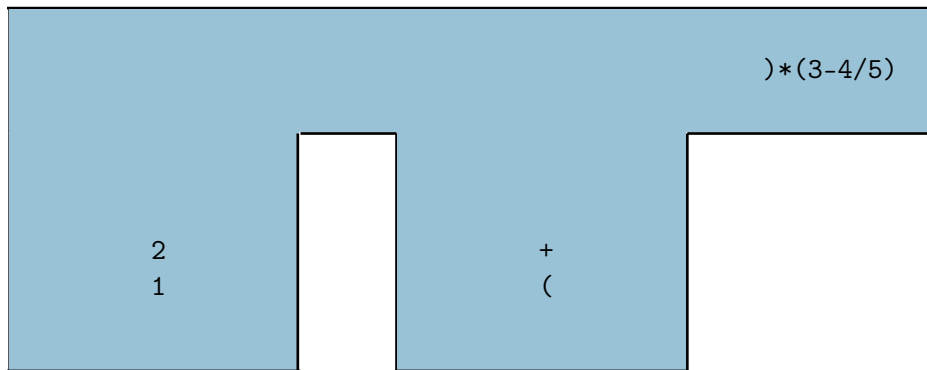


## Пример: Директно пресмятане на израз

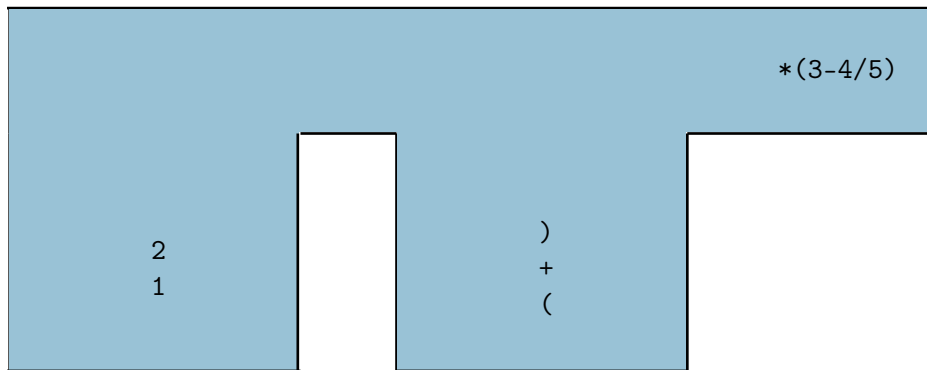




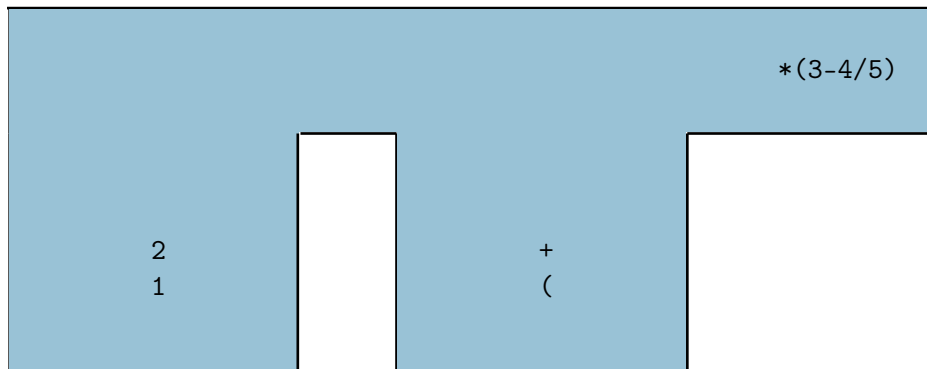
## Пример: Директно пресмятане на израз



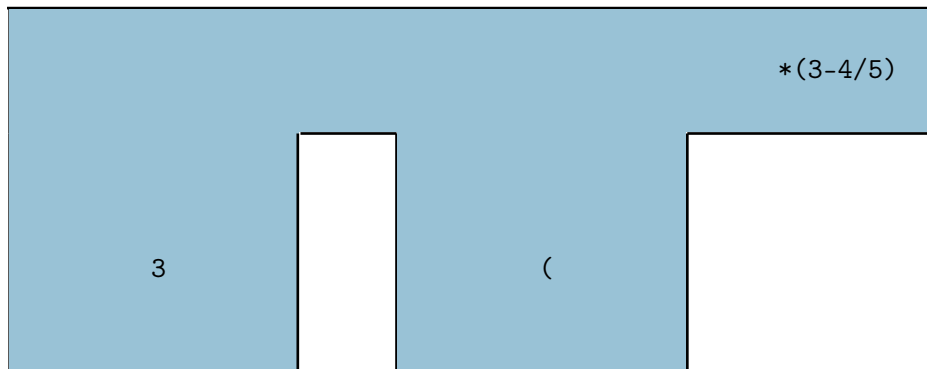
## Пример: Директно пресмятане на израз



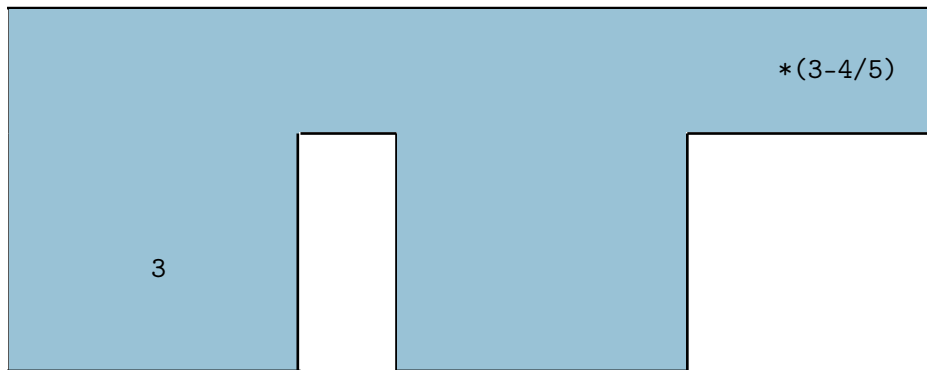
## Пример: Директно пресмятане на израз



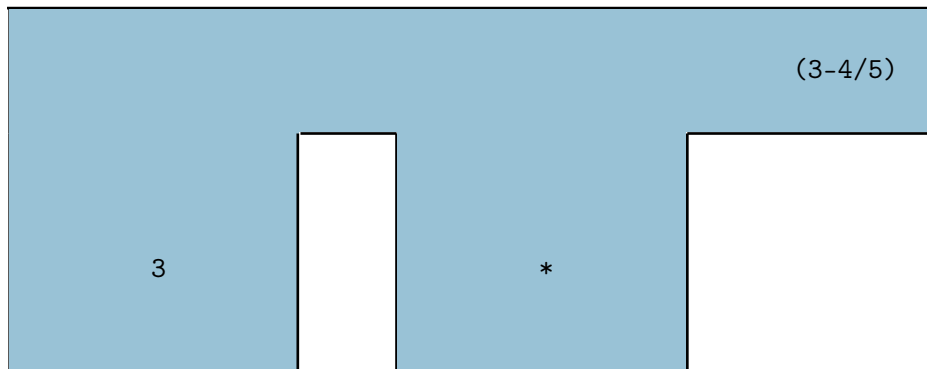
## Пример: Директно пресмятане на израз



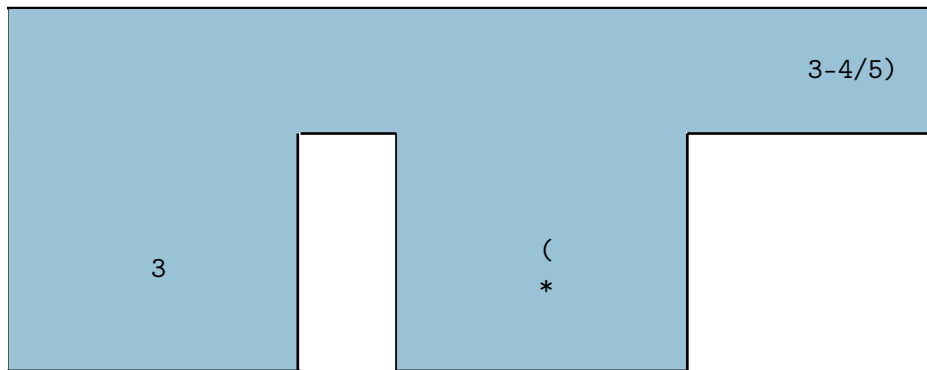
# Пример: Директно пресмятане на израз



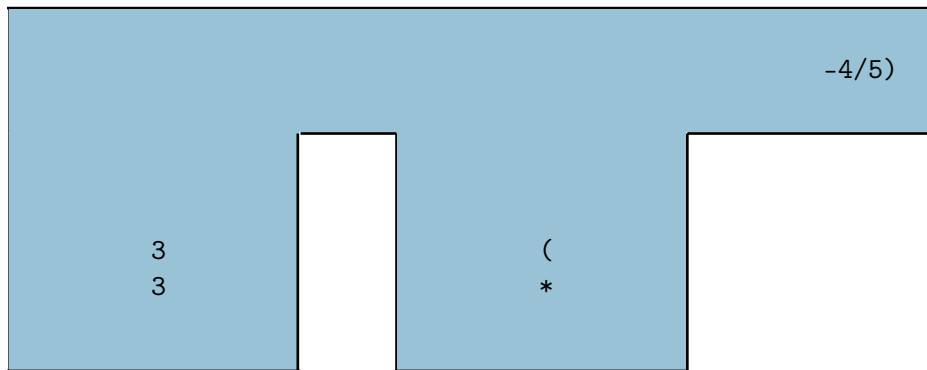
# Пример: Директно пресмятане на израз



# Пример: Директно пресмятане на израз

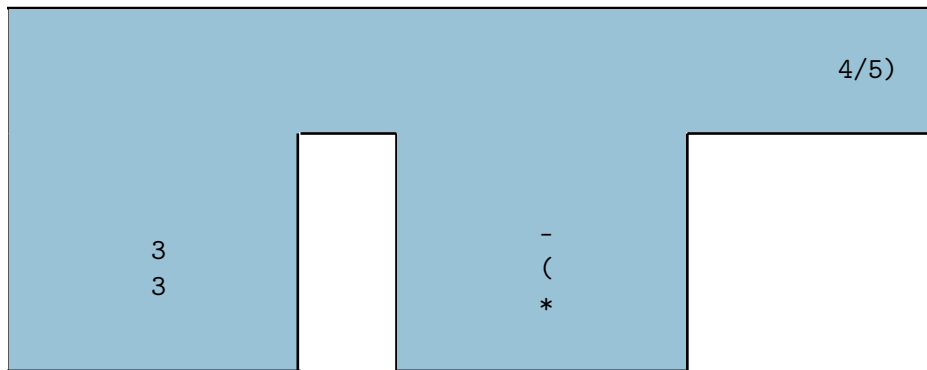


# Пример: Директно пресмятане на израз

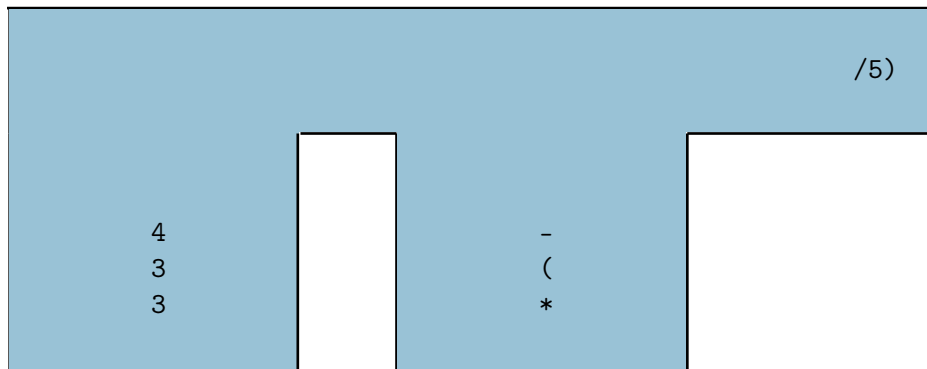




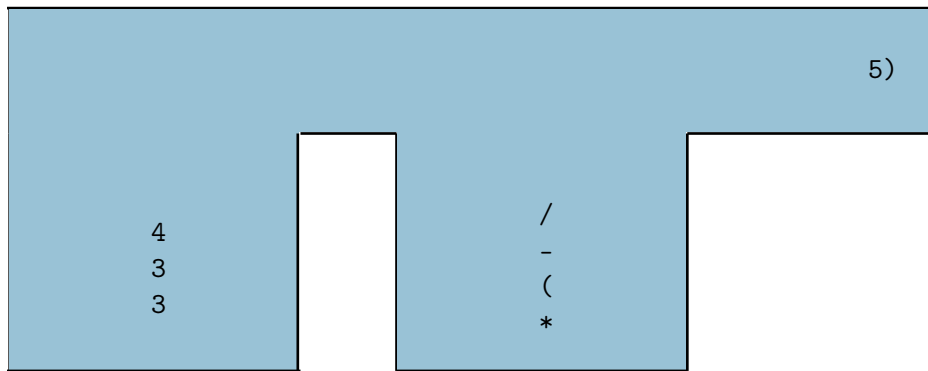
## Пример: Директно пресмятане на израз



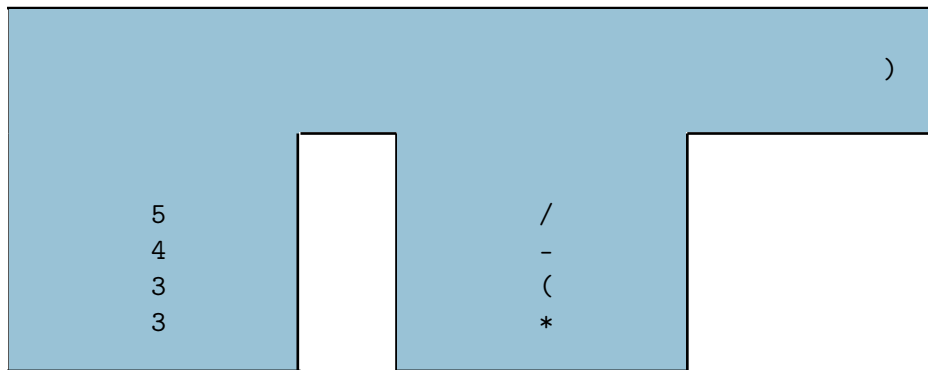
## Пример: Директно пресмятане на израз



## Пример: Директно пресмятане на израз



## Пример: Директно пресмятане на израз



## Пример: Директно пресмятане на израз

5  
4  
3  
3)  
/  
-  
(  
\*

## Пример: Директно пресмятане на израз

5  
4  
3  
3/  
-  
(  
\*

# Пример: Директно пресмятане на израз

0.8  
3  
3

-  
(  
\*

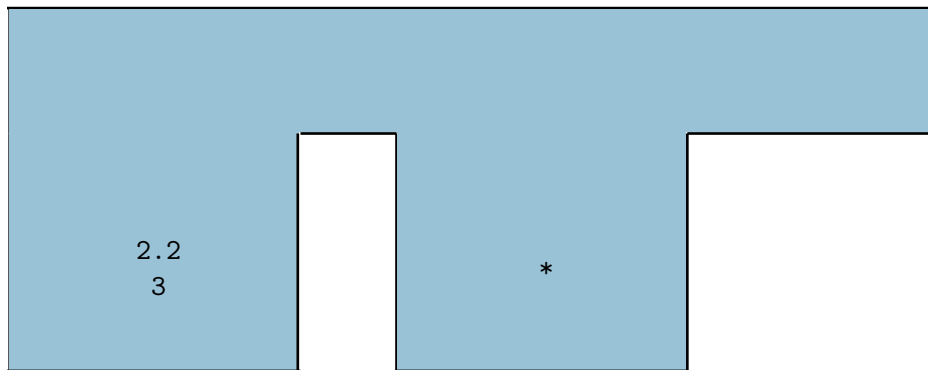
# Пример: Директно пресмятане на израз

2.2  
3

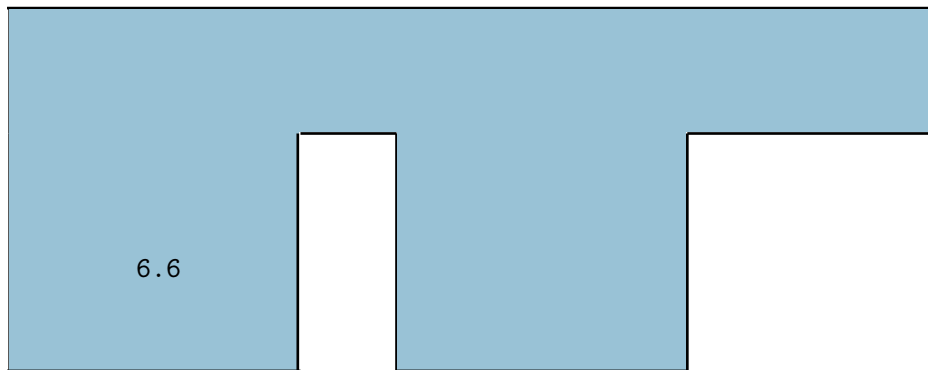
(  
\*



# Пример: Директно пресмятане на израз



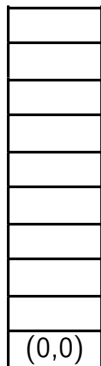
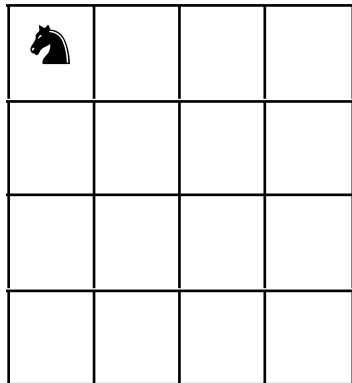
## Пример: Директно пресмятане на израз



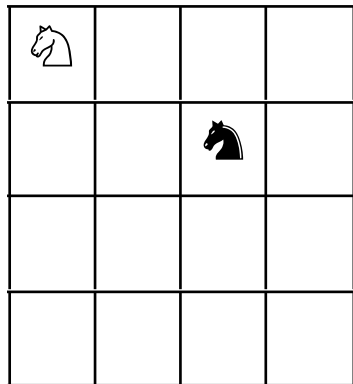
# Симулиране на рекурсия

- Стекова рамка
  - при извикване на функция
  - при рекурсия
- Стек вместо стекова рамка
- Пример: ход на коня

## Пример: ход на коня

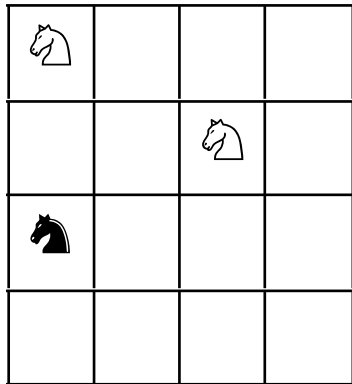

 $(x, y)$ 
 $(x+1, y+2)$ 
 $(x+1, y-2)$ 
 $(x-1, y+2)$ 
 $(x-1, y-2)$ 
 $(x+2, y+1)$ 
 $(x+2, y-1)$ 
 $(x-2, y+1)$ 
 $(x-2, y-1)$

## Пример: ход на коня



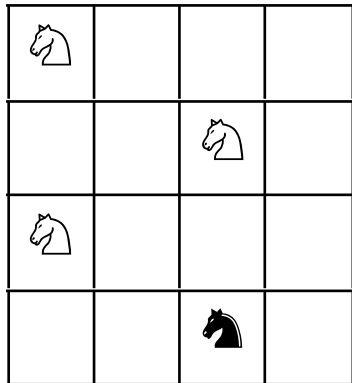
(1,2)
(0,0)

## Пример: ход на коня



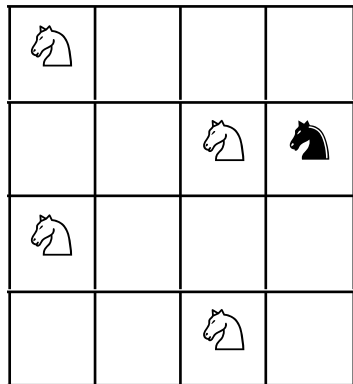
(2,0)
(1,2)
(0,0)

## Пример: ход на коня



(3,2)
(2,0)
(1,2)
(0,0)

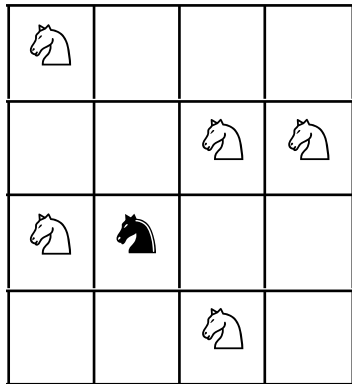
## Пример: ход на коня



(1,3)
(3,2)
(2,0)
(1,2)
(0,0)

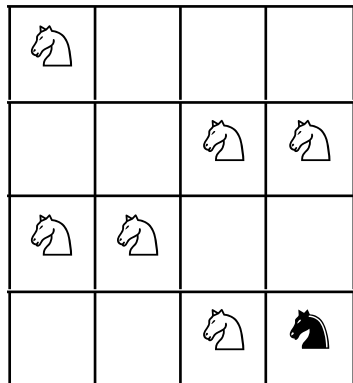


## Пример: ход на коня



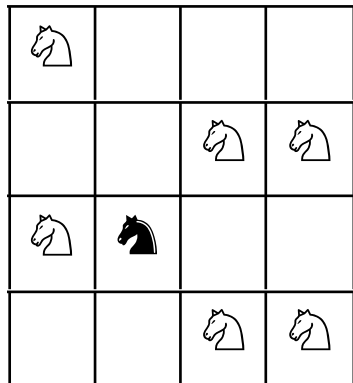
(2,1)
(1,3)
(3,2)
(2,0)
(1,2)
(0,0)

# Пример: ход на коня



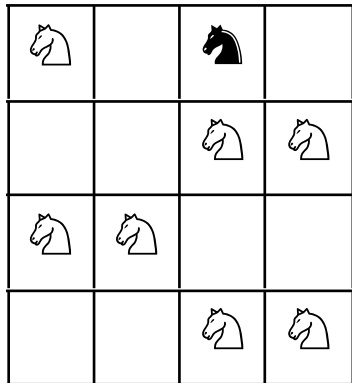
(3,3)
(2,1)
(1,3)
(3,2)
(2,0)
(1,2)
(0,0)

## Пример: ход на коня



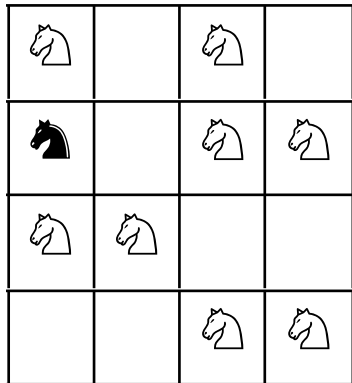
(2,1)
(1,3)
(3,2)
(2,0)
(1,2)
(0,0)

## Пример: ход на коня



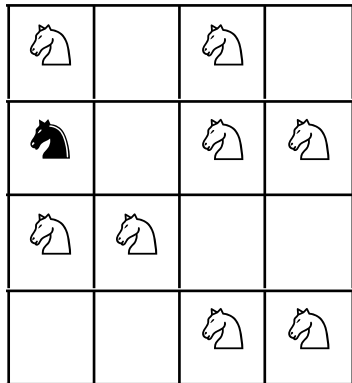
(0,2)
(2,1)
(1,3)
(3,2)
(2,0)
(1,2)
(0,0)

## Пример: ход на коня



(1,0)
(0,2)
(2,1)
(1,3)
(3,2)
(2,0)
(1,2)
(0,0)

# Пример: ход на коня



(1,0)
(0,2)
(2,1)
(1,3)
(3,2)
(2,0)
(1,2)
(0,0)

(0,1)

(1,1)

(0,1)

(3,1)

(2,1)

(3,3)

## `std::stack<T>`

- `stack()` — създаване на празен стек
- `empty()` — проверка за празнота на стек
- `push(x)` — включване на елемент на стек
- `pop()` — изключване на елемент от стек
- `top()` — последен елемент на стека
- `size()` — дължина на стека
- `==, !=, <, >, <=, >=` — лексикорафско сравнение на два стека