

Стек

Трифон Трифонов

Структури от данни и програмиране, спец. Компютърни науки, 2 поток, 2024/25 г.

13–20 октомври 2022 г.

Тази презентация е достъпна под лиценза Creative Commons Признание-Некомерсиално-Споделяне на споделеното 4.0 Международен 



"Stack of Rocks, Cattle Point, San Juan Island" от Ryan Harvey.
CC BY SA 2.0

АТД: стек

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

Операции

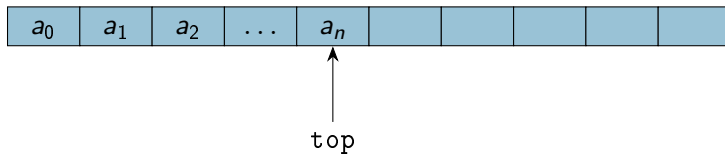
- `create()` — създаване на празен стек
- `empty()` — проверка за празнота на стек
- `push(x)` — включване на елемент на стек
- `pop()` — изключване на елемент от стек
- `peek()` — последен елемент на стека

АТД: стек

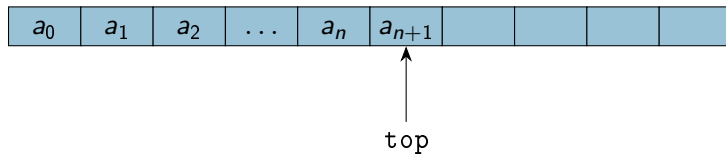
Свойства на операциите

- `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 на стек

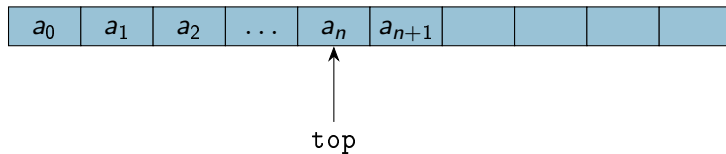


Последователно представяне на стек



- включване на елемент (push)

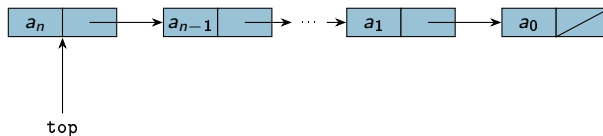
Последователно представяне на стек



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

Свързано представяне на стек

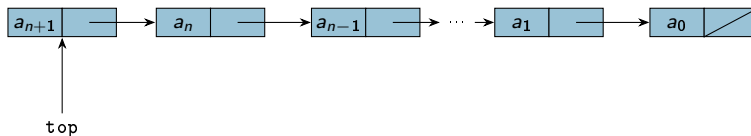
Представяме стека като “верига” от двойни кутии



```
struct StackElement {  
    int data;  
    StackElement* next;  
};
```

Свързано представяне на стек

Представяме стека като “верига” от двойни кутии

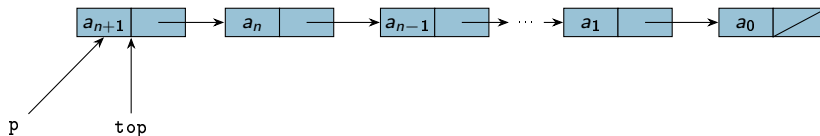


```
struct StackElement {  
    int data;  
    StackElement* next;  
};
```

- включване на елемент (push)

Свързано представяне на стек

Представяме стека като “верига” от двойни кутии



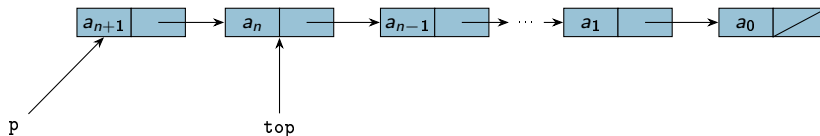
```

struct StackElement {
    int data;
    StackElement* next;
};
  
```

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

Свързано представяне на стек

Представяме стека като “верига” от двойни кутии



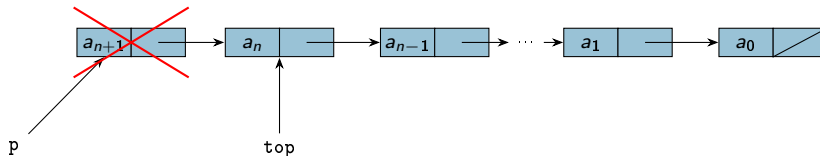
```

struct StackElement {
    int data;
    StackElement* next;
};
  
```

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

Свързано представяне на стек

Представяме стека като “верига” от двойни кутии

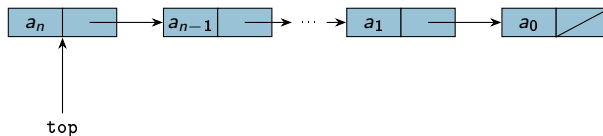


```
struct StackElement {
    int data;
    StackElement* next;
};
```

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

Свързано представяне на стек

Представяме стека като “верига” от двойни кутии

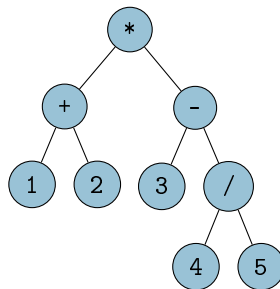


```
struct StackElement {  
    int data;  
    StackElement* next;  
};
```

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

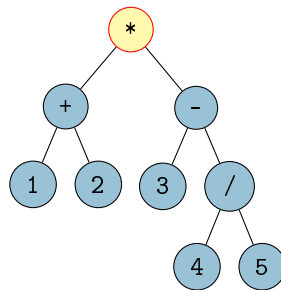
Обратен полски запис

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



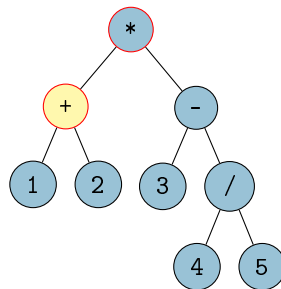
Обратен полски запис

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



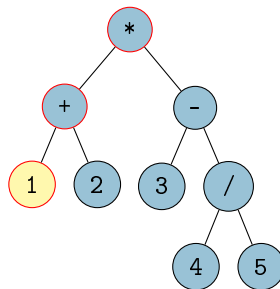
Обратен полски запис

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



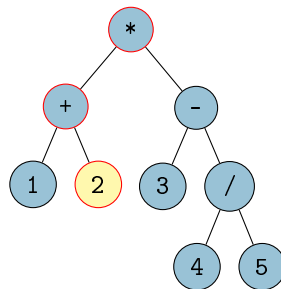
Обратен полски запис

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



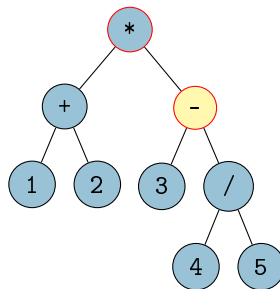
Обратен полски запис

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



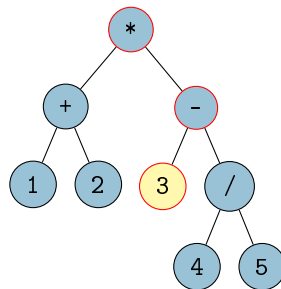
Обратен полски запис

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



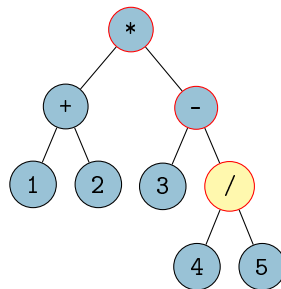
Обратен полски запис

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



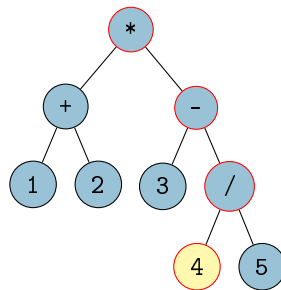
Обратен полски запис

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



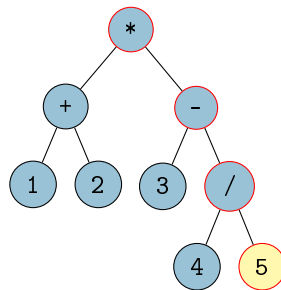
Обратен полски запис

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



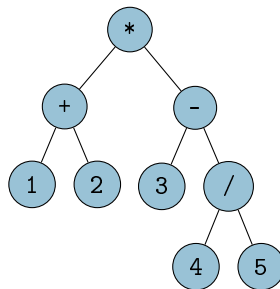
Обратен полски запис

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



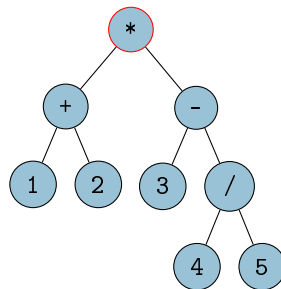
Обратен полски запис

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



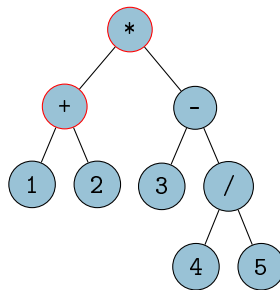
Обратен полски запис

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



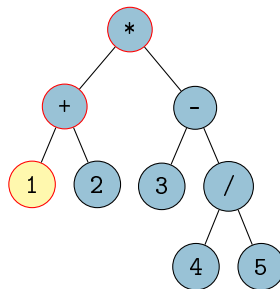
Обратен полски запис

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



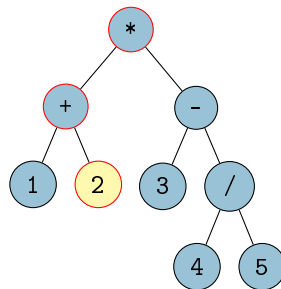
Обратен полски запис

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



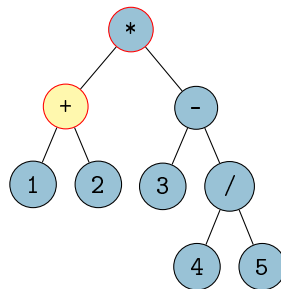
Обратен полски запис

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



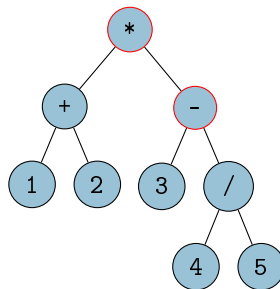
Обратен полски запис

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



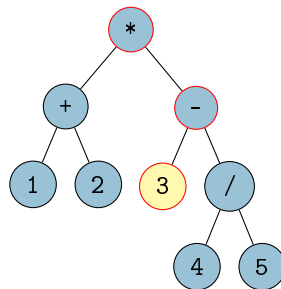
Обратен полски запис

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



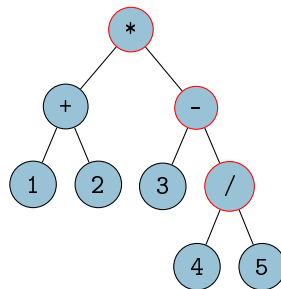
Обратен полски запис

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



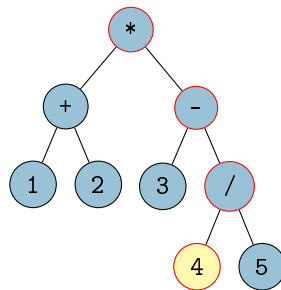
Обратен полски запис

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



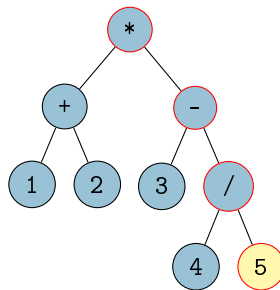
Обратен полски запис

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



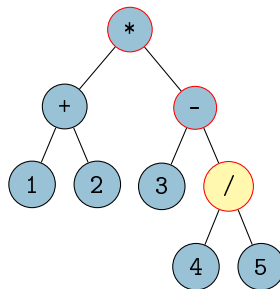
Обратен полски запис

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



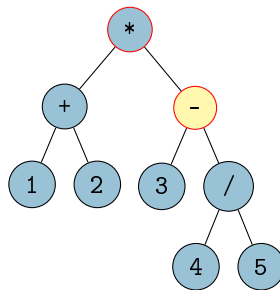
Обратен полски запис

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



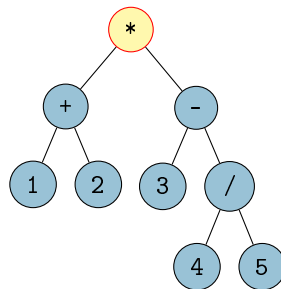
Обратен полски запис

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



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



Алгоритъм за пресмятане на обратен полски запис

За всеки пореден символ s от израза в обратен полски запис:

Алгоритъм за пресмятане на обратен полски запис

За всеки пореден символ s от израза в обратен полски запис:

- Ако s е цифра, добавяме стойността ѝ в стека за резултати

Алгоритъм за пресмятане на обратен полски запис

За всеки пореден символ s от израза в обратен полски запис:

- Ако s е цифра, добавяме стойността ѝ в стека за резултати
- Ако s е (двуместна) операция:

Алгоритъм за пресмятане на обратен полски запис

За всеки пореден символ s от израза в обратен полски запис:

- Ако s е цифра, добавяме стойността ѝ в стека за резултати
- Ако s е (двуместна) операция:
 - изваждаме най-горните два елемента от стека

Алгоритъм за пресмятане на обратен полски запис

За всеки пореден символ s от израза в обратен полски запис:

- Ако s е цифра, добавяме стойността ѝ в стека за резултати
- Ако s е (двуместна) операция:
 - изваждаме най-горните два елемента от стека
 - прилагаме операцията над тях

Алгоритъм за пресмятане на обратен полски запис

За всеки пореден символ s от израза в обратен полски запис:

- Ако s е цифра, добавяме стойността ѝ в стека за резултати
- Ако s е (двуместна) операция:
 - изваждаме най-горните два елемента от стека
 - прилагаме операцията над тях
 - добавяме резултата в стека

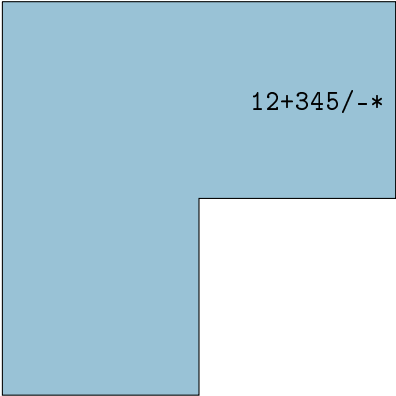
Алгоритъм за пресмятане на обратен полски запис

За всеки пореден символ s от израза в обратен полски запис:

- Ако s е цифра, добавяме стойността ѝ в стека за резултати
- Ако s е (двуместна) операция:
 - изваждаме най-горните два елемента от стека
 - прилагаме операцията над тях
 - добавяме резултата в стека

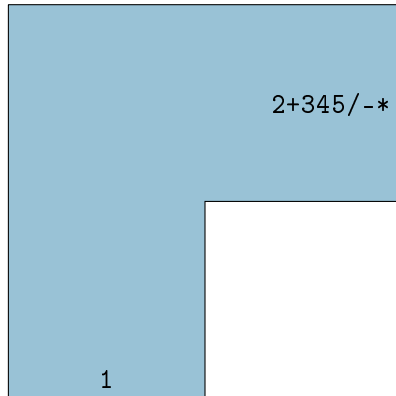
В стека остава единствен елемент: крайният резултат.

Пример: Пресмятане на израз в обратен полски запис

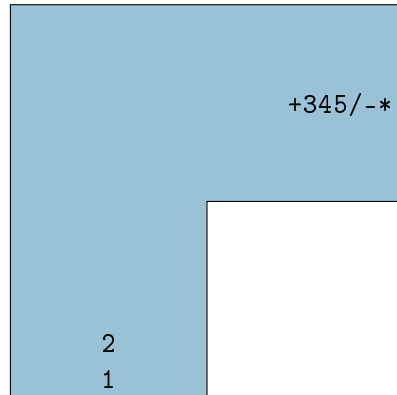


12+345/-*

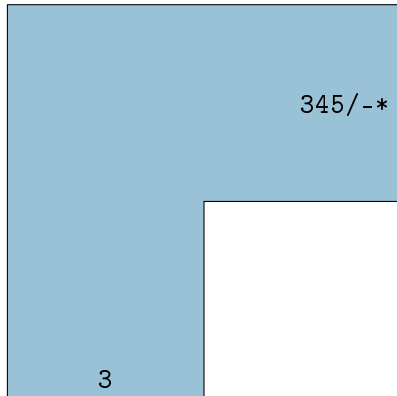
Пример: Пресмятане на израз в обратен полски запис



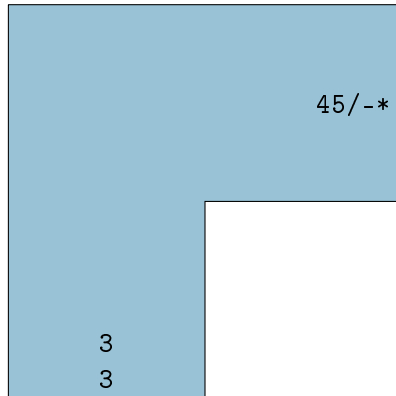
Пример: Пресмятане на израз в обратен полски запис



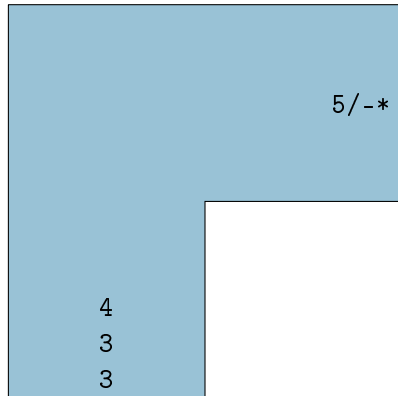
Пример: Пресмятане на израз в обратен полски запис



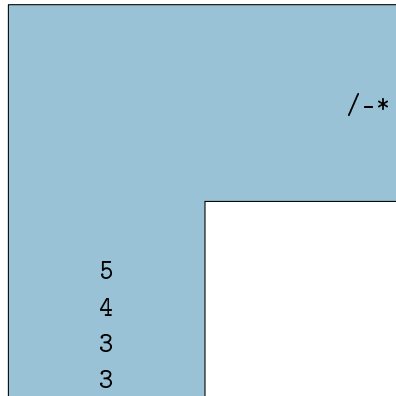
Пример: Пресмятане на израз в обратен полски запис



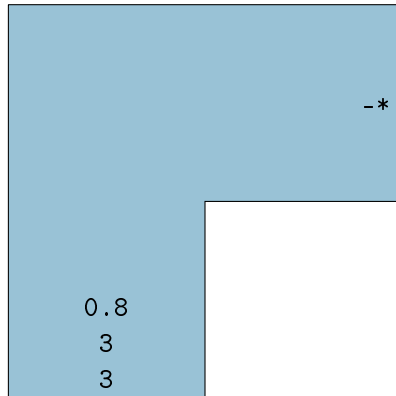
Пример: Пресмятане на израз в обратен полски запис



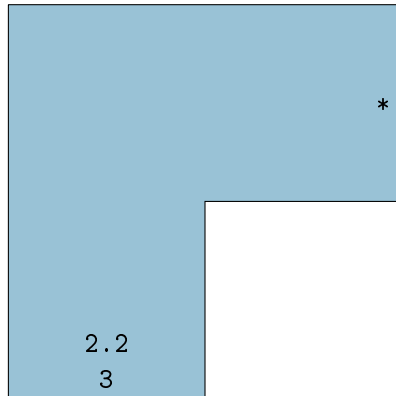
Пример: Пресмятане на израз в обратен полски запис



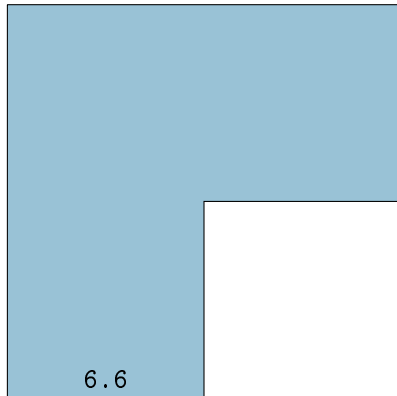
Пример: Пресмятане на израз в обратен полски запис



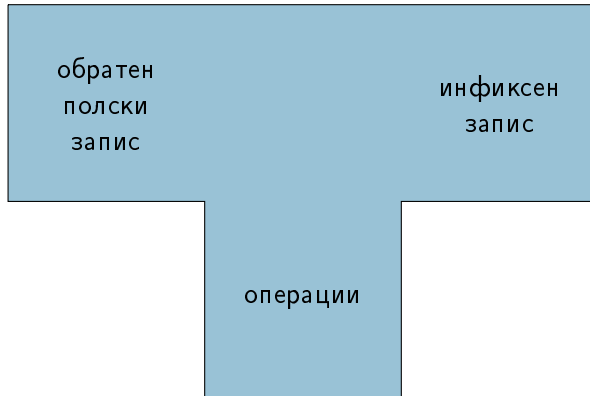
Пример: Пресмятане на израз в обратен полски запис



Пример: Пресмятане на израз в обратен полски запис



Преобразуване в обратен полски запис



Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ с:

Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

- Ако s е цифра, прехвърляме я в крайния резултат

Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

- Ако s е цифра, прехвърляме я в крайния резултат
- Ако s е отваряща скоба, поставяме я в стека с операции

Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

- Ако s е цифра, прехвърляме я в крайния резултат
- Ако s е отваряща скоба, поставяме я в стека с операции
- Ако s е операция, поставяме я в стека с операции

Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

- Ако s е цифра, прехвърляме я в крайния резултат
- Ако s е отваряща скоба, поставяме я в стека с операции
- Ако s е операция, поставяме я в стека с операции
- Ако s е затваряща скоба, последователно изваждаме от стека и записваме в резултата всички операции до достигане на отваряща скоба

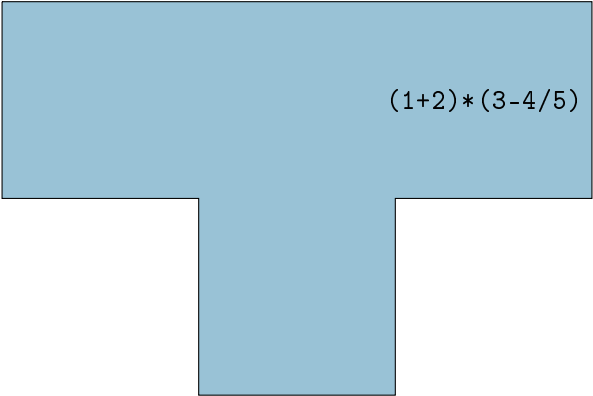
Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

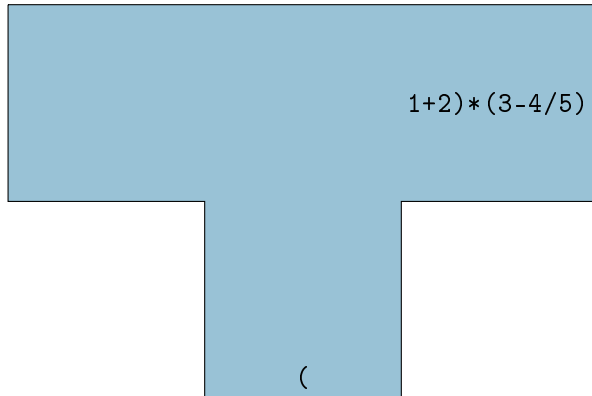
- Ако s е цифра, прехвърляме я в крайния резултат
- Ако s е отваряща скоба, поставяме я в стека с операции
- Ако s е операция, поставяме я в стека с операции
- Ако s е затваряща скоба, последователно изваждаме от стека и записваме в резултата всички операции до достигане на отваряща скоба

След приключване на входния низ, последователно изваждаме от стека и и записваме в резултата всички останали операции.

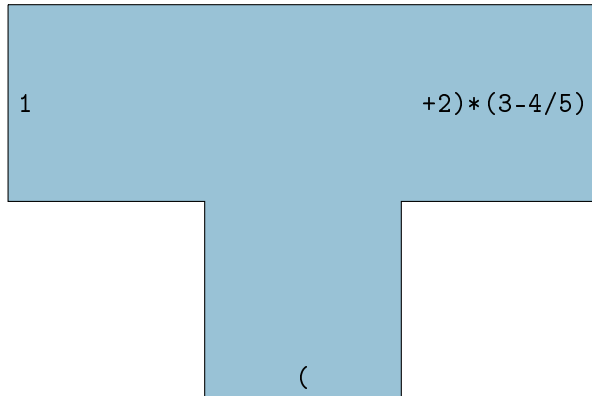
Пример: Преобразуване в обратен полски запис


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

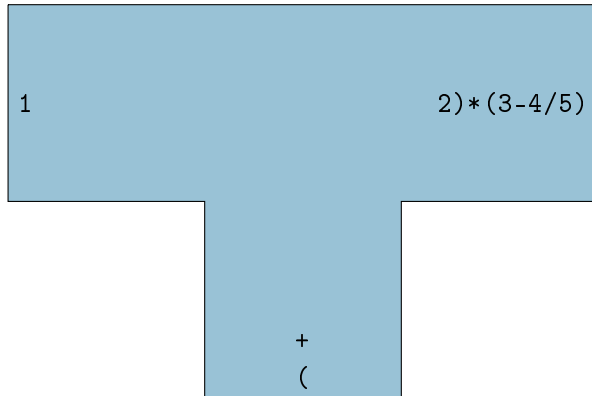
Пример: Преобразуване в обратен полски запис



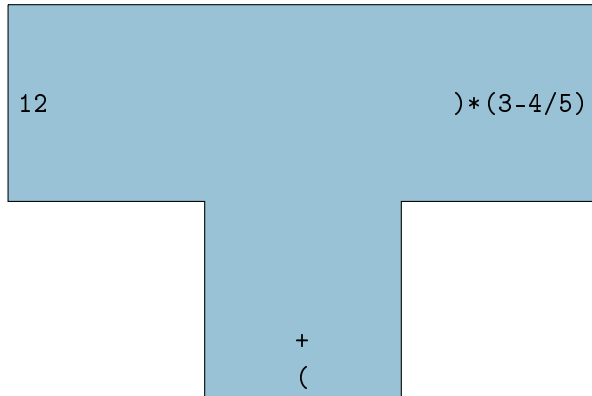
Пример: Преобразуване в обратен полски запис



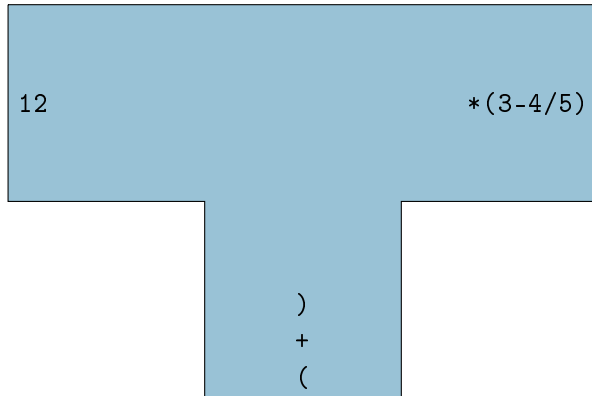
Пример: Преобразуване в обратен полски запис



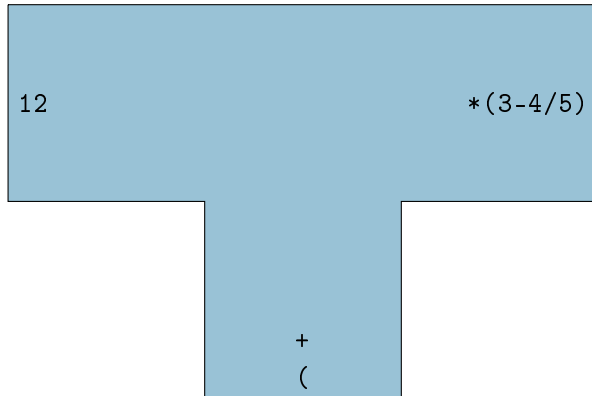
Пример: Преобразуване в обратен полски запис



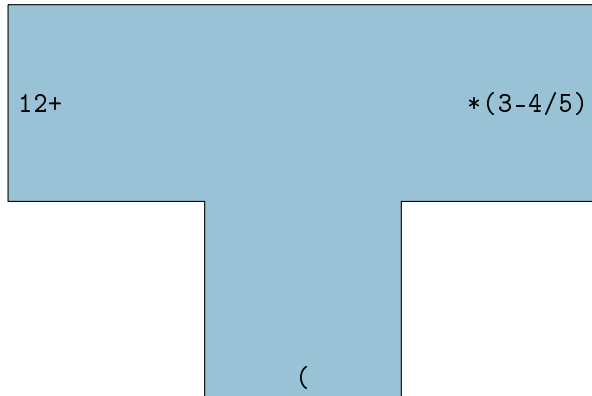
Пример: Преобразуване в обратен полски запис



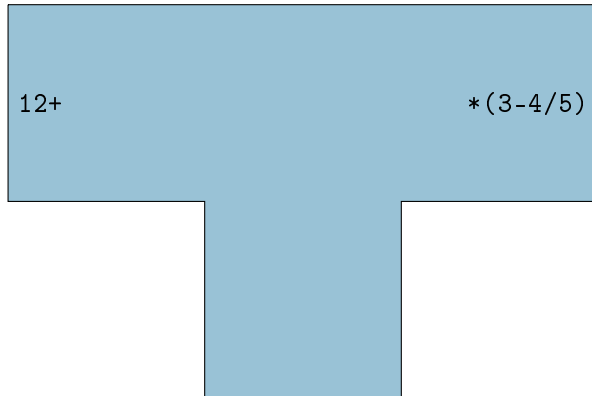
Пример: Преобразуване в обратен полски запис



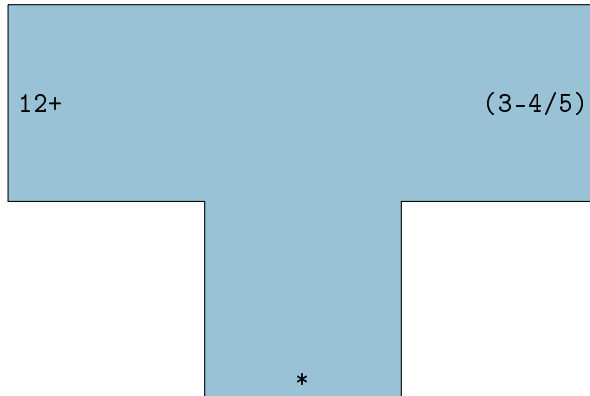
Пример: Преобразуване в обратен полски запис



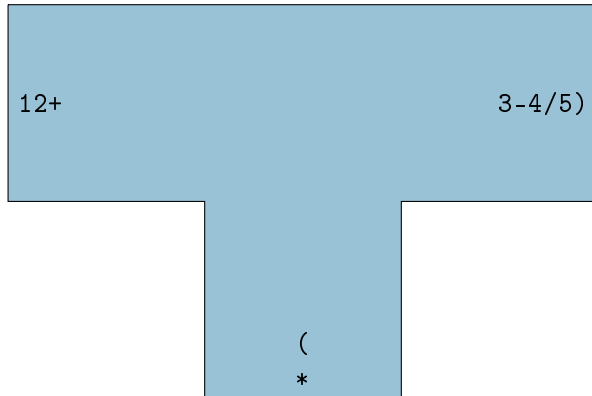
Пример: Преобразуване в обратен полски запис



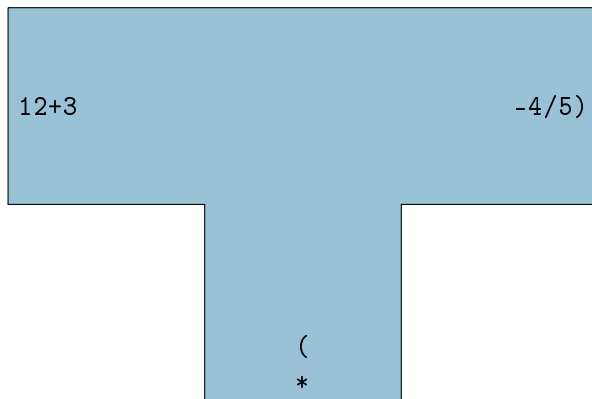
Пример: Преобразуване в обратен полски запис



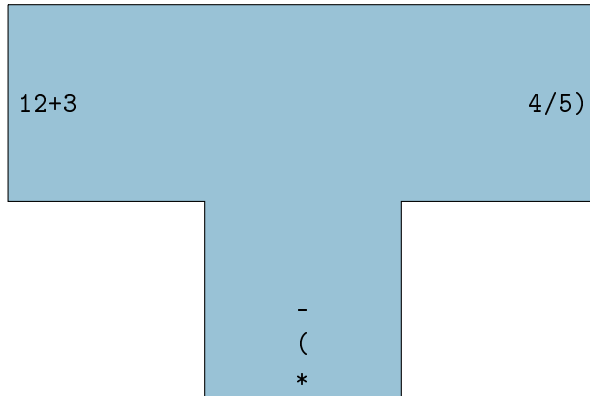
Пример: Преобразуване в обратен полски запис



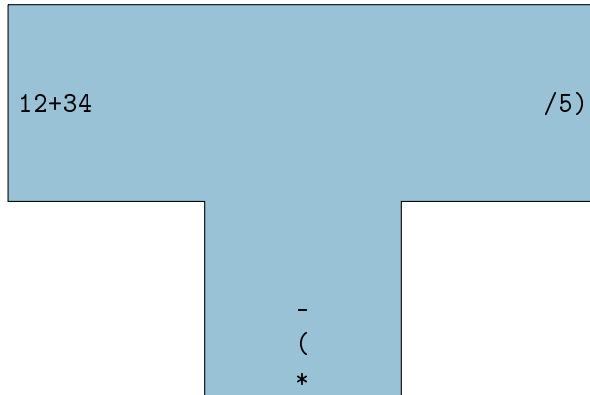
Пример: Преобразуване в обратен полски запис



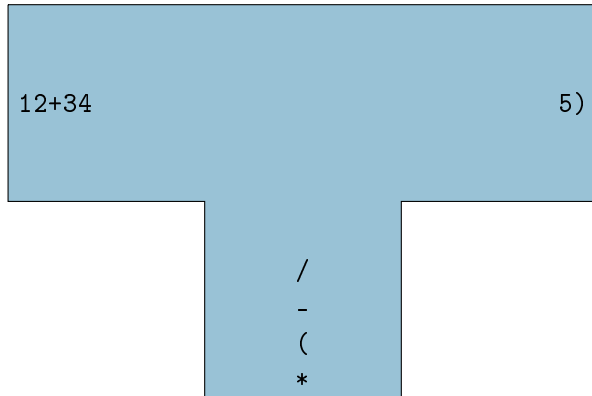
Пример: Преобразуване в обратен полски запис



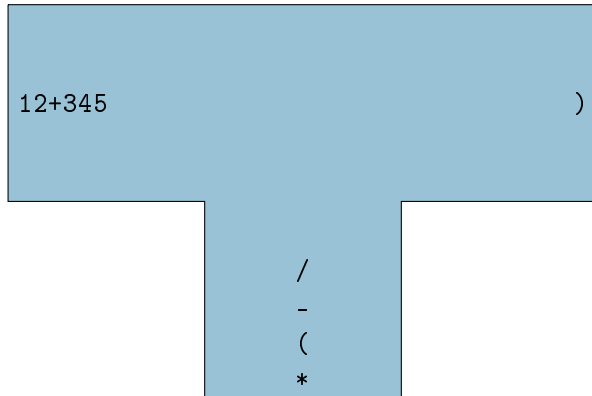
Пример: Преобразуване в обратен полски запис



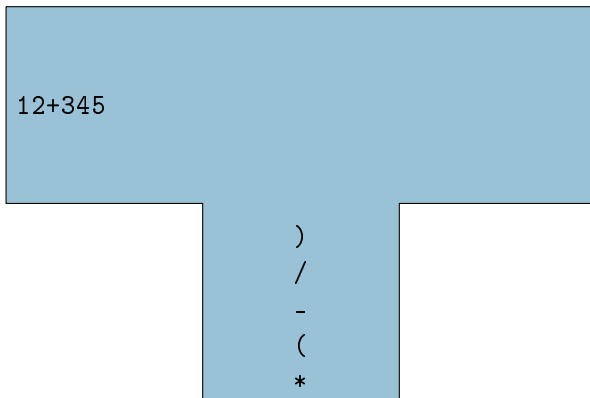
Пример: Преобразуване в обратен полски запис



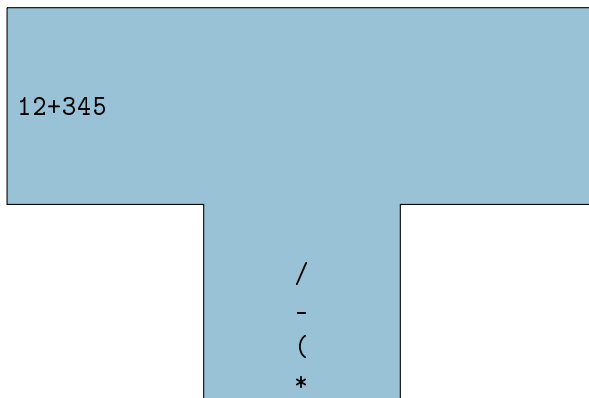
Пример: Преобразуване в обратен полски запис



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

12+345/-

(

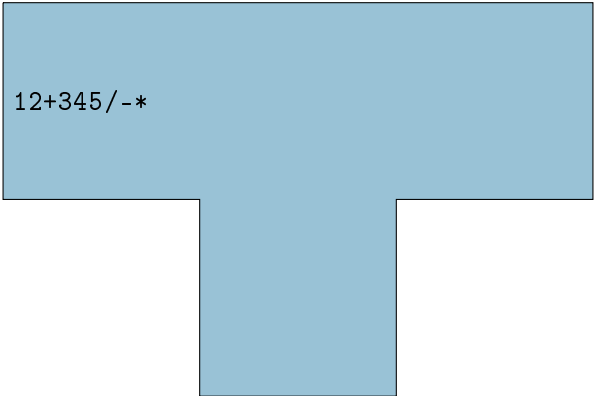
*

Пример: Преобразуване в обратен полски запис

12+345/-

*

Пример: Преобразуване в обратен полски запис



12+345/-*

Приоритети на операциите

Какъв е обратния полски запис на израза $(1+2)*(3/4-5)$?

- Ако следваме алгоритъма, описан по-рано, резултатът би трябвало да е $12+345-/*...$

Приоритети на операциите

Какъв е обратния полски запис на израза $(1+2)*(3/4-5)$?

- Ако следваме алгоритъма, описан по-рано, резултатът би трябвало да е $12+345-/*...$
- ...но резултатът от този израз е -9 , а не -12.75 !

Приоритети на операциите

Какъв е обратния полски запис на израза $(1+2)*(3/4-5)$?

- Ако следваме алгоритъма, описан по-рано, резултатът би трябвало да е $12+345-/*...$
- ...но резултатът от този израз е -9 , а не -12.75 !
- На обратния полски запис $12+345-/*$ съответства инфиксен запис $(1+2)*(3/(4-5))$!

Приоритети на операциите

Какъв е обратния полски запис на израза $(1+2)*(3/4-5)$?

- Ако следваме алгоритъма, описан по-рано, резултатът би трябвало да е $12+345-/*...$
- ...но резултатът от този израз е -9 , а не -12.75 !
- На обратния полски запис $12+345-/*$ съответства инфиксен запис $(1+2)*(3/(4-5))$!
- Не сме взели предвид по-високия приоритет на операцията $/$

Приоритети на операциите

Какъв е обратния полски запис на израза $(1+2)*(3/4-5)$?

- Ако следваме алгоритъма, описан по-рано, резултатът би трябвало да е $12+345-/*...$
- ...но резултатът от този израз е -9 , а не -12.75 !
- На обратния полски запис $12+345-/*$ съответства инфиксен запис $(1+2)*(3/(4-5))$!
- Не сме взели предвид по-високия приоритет на операцията $/$
- Правилният обратен полски запис на $(1+2)*(3/4-5)$ е $12+34/5-*$.

Приоритети на операциите

Какъв е обратния полски запис на израза $(1+2)*(3/4-5)$?

- Ако следваме алгоритъма, описан по-рано, резултатът би трябвало да е $12+345-/*...$
- ...но резултатът от този израз е -9 , а не -12.75 !
- На обратния полски запис $12+345-/*$ съответства инфиксен запис $(1+2)*(3/(4-5))!$
- Не сме взели предвид по-високия приоритет на операцията $/$
- Правилният обратен полски запис на $(1+2)*(3/4-5)$ е $12+34/5-*$.
- Необходимо е да променим алгоритъма, така че да взема предвид приоритета на операциите!

Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

- Ако s е цифра, прехвърляме я в крайния резултат
- Ако s е отваряща скоба, поставяме я в стека с операции
- Ако s е операция, **поставяме я в стека с операции**
- Ако s е затваряща скоба, последователно изваждаме от стека и записваме в резултата всички операции до достигане на отваряща скоба

След приключване на входния низ, последователно изваждаме от стека и и записваме в резултата всички останали операции.

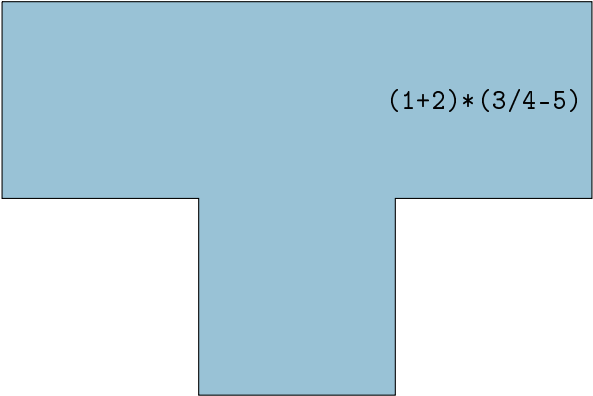
Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

- Ако s е цифра, прехвърляме я в крайния резултат
- Ако s е отваряща скоба, поставяме я в стека с операции
- Ако s е операция:
 - изваждаме от стека и записваме в резултата всички операции с приоритет по-висок или равен на s
 - поставяме s в стека
- Ако s е затваряща скоба, последователно изваждаме от стека и записваме в резултата всички операции до достигане на отваряща скоба

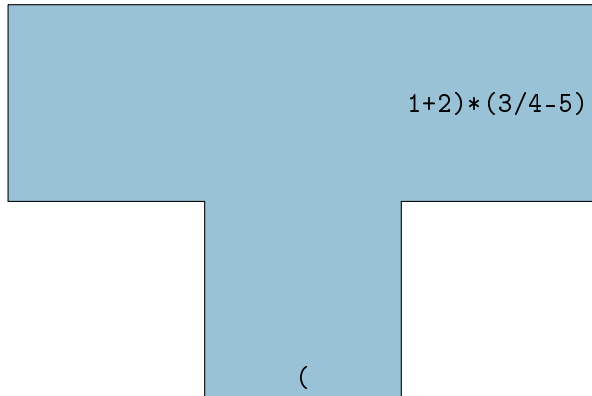
След приключване на входния низ, последователно изваждаме от стека и записваме в резултата всички останали операции.

Пример 2: Преобразуване в обратен полски запис

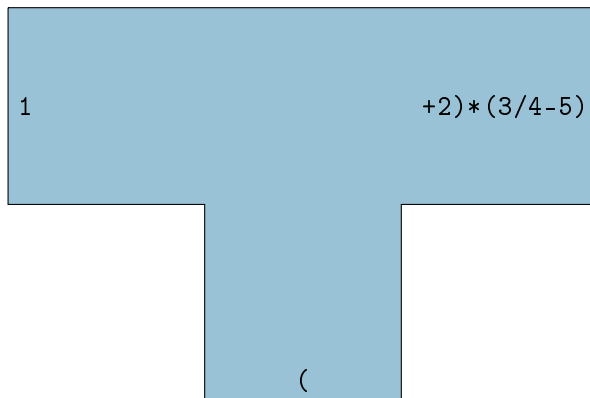


$(1+2)*(3/4-5)$

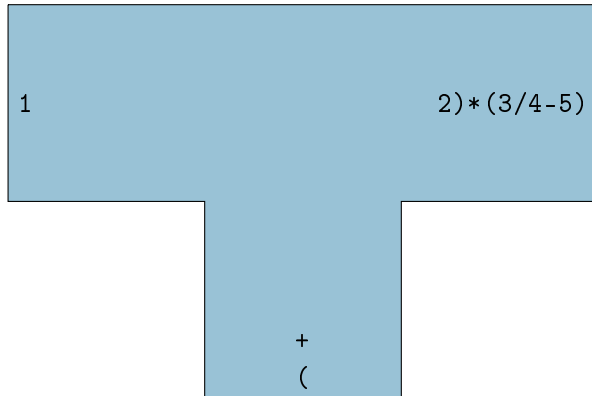
Пример 2: Преобразуване в обратен полски запис



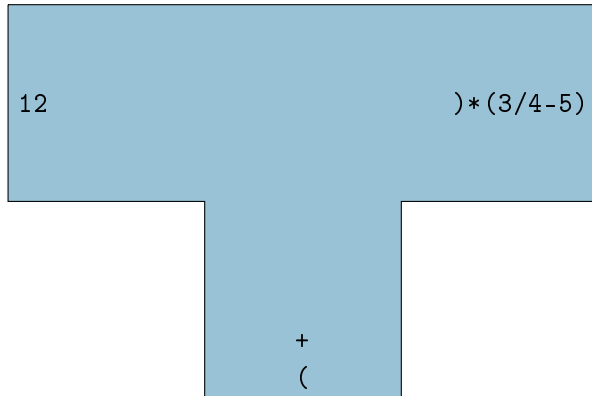
Пример 2: Преобразуване в обратен полски запис



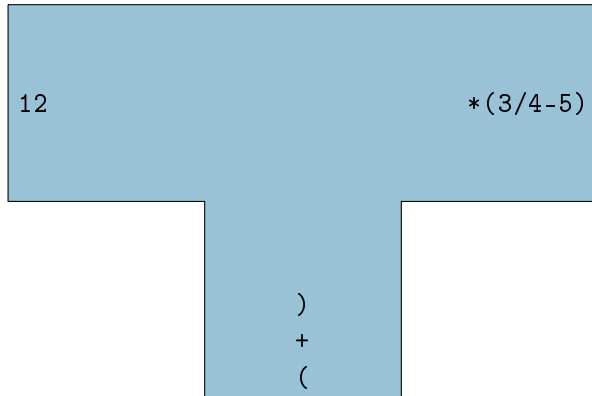
Пример 2: Преобразуване в обратен полски запис



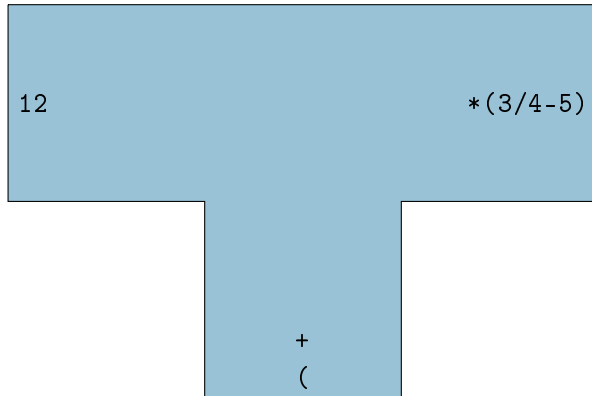
Пример 2: Преобразуване в обратен полски запис



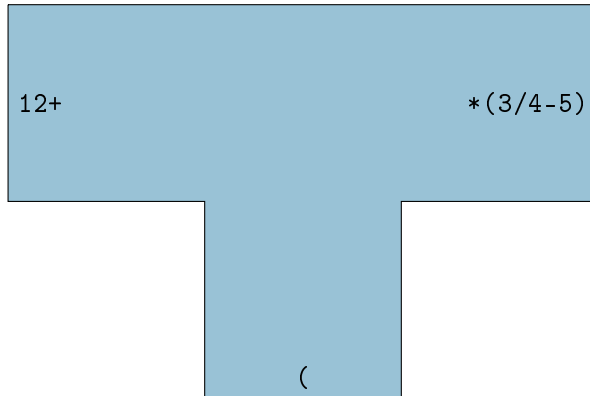
Пример 2: Преобразуване в обратен полски запис



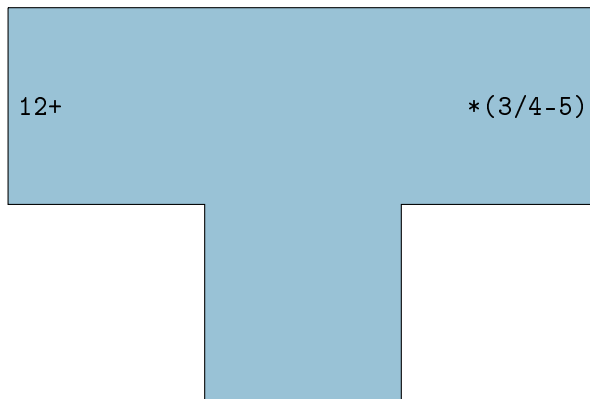
Пример 2: Преобразуване в обратен полски запис



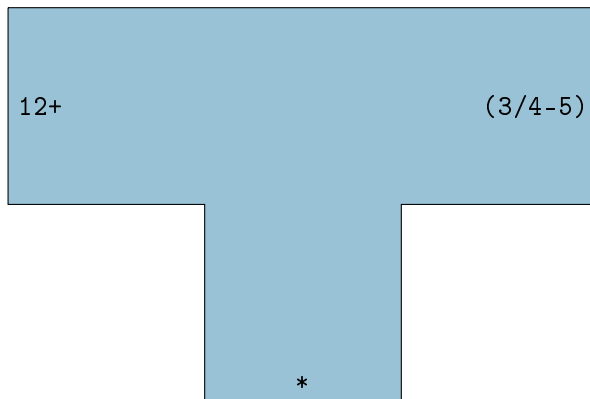
Пример 2: Преобразуване в обратен полски запис



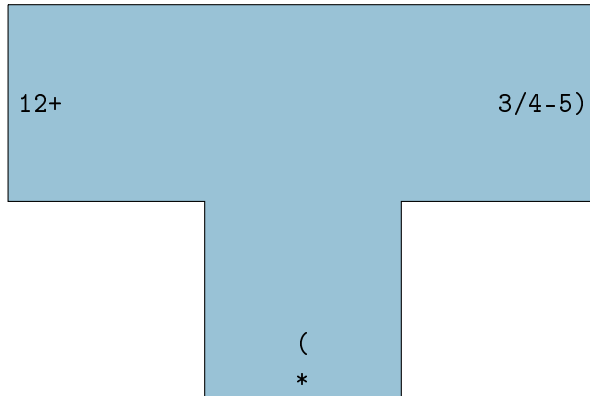
Пример 2: Преобразуване в обратен полски запис



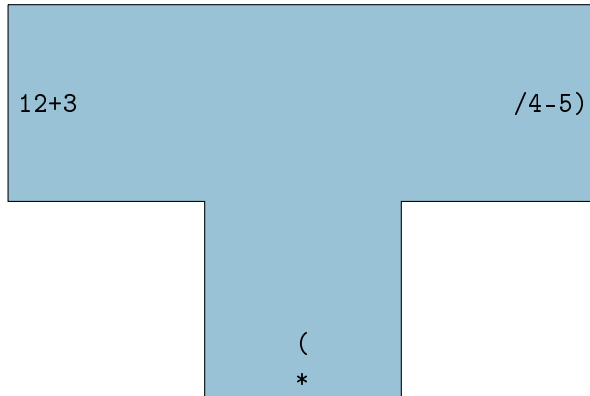
Пример 2: Преобразуване в обратен полски запис



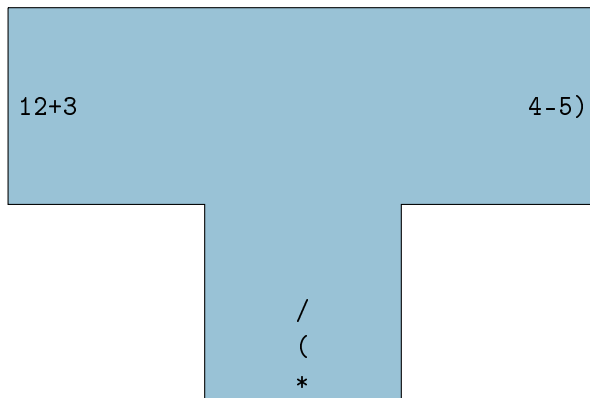
Пример 2: Преобразуване в обратен полски запис



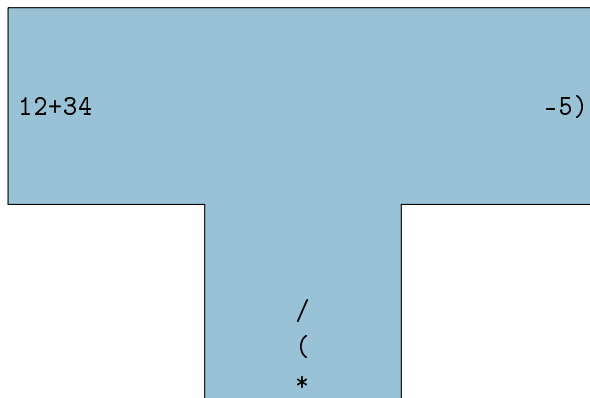
Пример 2: Преобразуване в обратен полски запис



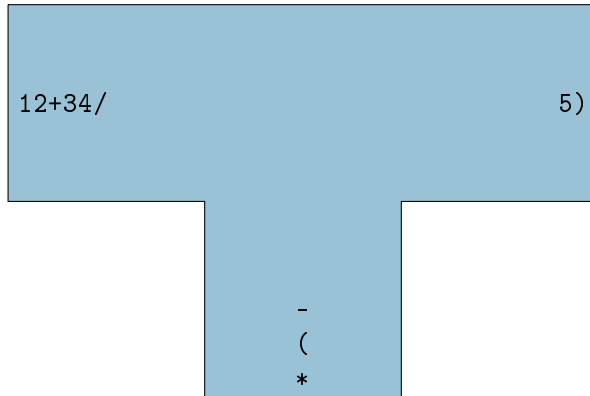
Пример 2: Преобразуване в обратен полски запис



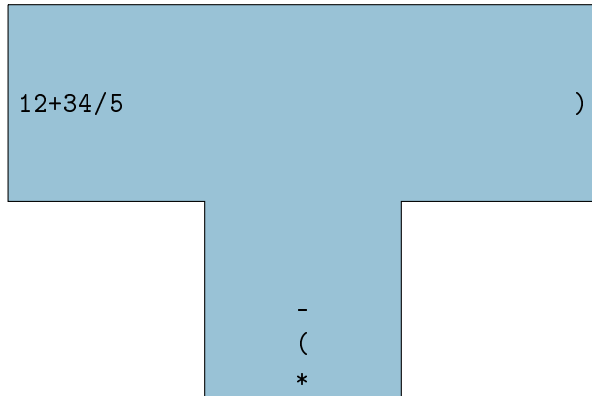
Пример 2: Преобразуване в обратен полски запис



Пример 2: Преобразуване в обратен полски запис



Пример 2: Преобразуване в обратен полски запис



Пример 2: Преобразуване в обратен полски запис

12+34/5

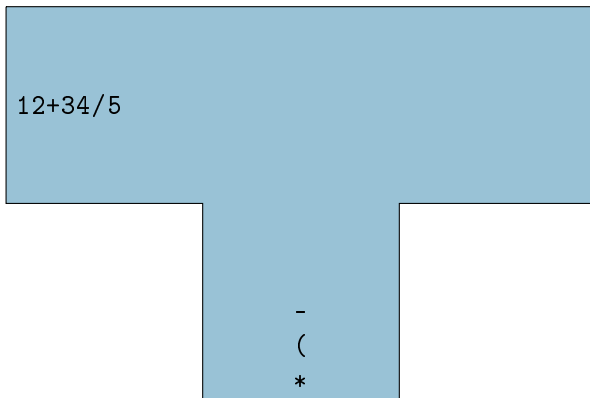
)

-

(

*

Пример 2: Преобразуване в обратен полски запис



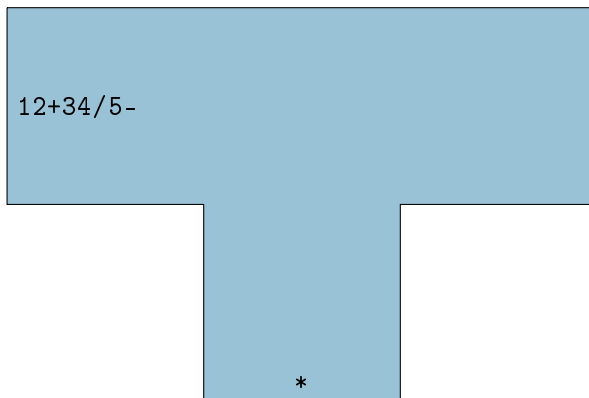
Пример 2: Преобразуване в обратен полски запис

12+34/5-

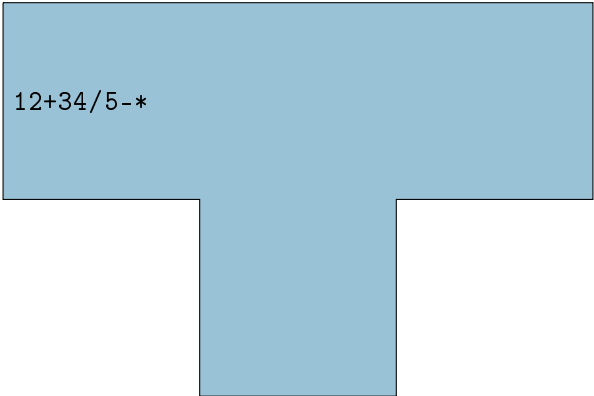
(

*

Пример 2: Преобразуване в обратен полски запис



Пример 2: Преобразуване в обратен полски запис



$12+34/5-*$

Директно пресмятане на израз

Ако комбинираме двата алгоритъма, можем да пресмятаме инфиксен израз директно, без използването на обратен полски запис:



Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

- Ако s е цифра, прехвърляме я в **крайния резултат**
- Ако s е отваряща скоба, поставяме я в стека с операции
- Ако s е операция:
 - изваждаме от стека и записваме в резултата всички операции с приоритет по-висок или равен на s
 - поставяме s в стека
- Ако s е затваряща скоба, последователно изваждаме от стека и записваме в резултата всички операции до достигане на отваряща скоба

След приключване на входния низ, последователно изваждаме от стека и записваме в резултата всички останали операции.

Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

- Ако s е цифра, прехвърляме я в стека с резултати
- Ако s е отваряща скоба, поставяме я в стека с операции
- Ако s е операция:
 - изваждаме от стека и записваме в резултата всички операции с приоритет по-висок или равен на s
 - поставяме s в стека
- Ако s е затваряща скоба, последователно изваждаме от стека и записваме в резултата всички операции до достигане на отваряща скоба

След приключване на входния низ, последователно изваждаме от стека и записваме в резултата всички останали операции.

Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

- Ако s е цифра, прехвърляме я в стека с резултати
- Ако s е отваряща скоба, поставяме я в стека с операции
- Ако s е операция:
 - изваждаме от стека **и записваме в резултата** всички операции с приоритет по-висок или равен на s
 - поставяме s в стека
- Ако s е затваряща скоба, последователно изваждаме от стека **и записваме в резултата** всички операции до достигане на отваряща скоба

След приключване на входния низ, последователно изваждаме от стека **и записваме в резултата** всички останали операции.

Алгоритъм на разпределителната гара (shunting yard)

За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

- Ако s е цифра, прехвърляме я в стека с резултати
- Ако s е отваряща скоба, поставяме я в стека с операции
- Ако s е операция:
 - изваждаме от стека и прилагаме всички операции с приоритет по-висок или равен на s
 - поставяме s в стека
- Ако s е затваряща скоба, последователно изваждаме от стека и прилагаме всички операции до достигане на отваряща скоба

След приключване на входния низ, последователно изваждаме от стека и прилагаме всички останали операции.

Алгоритъм на разпределителната гара (shunting yard)

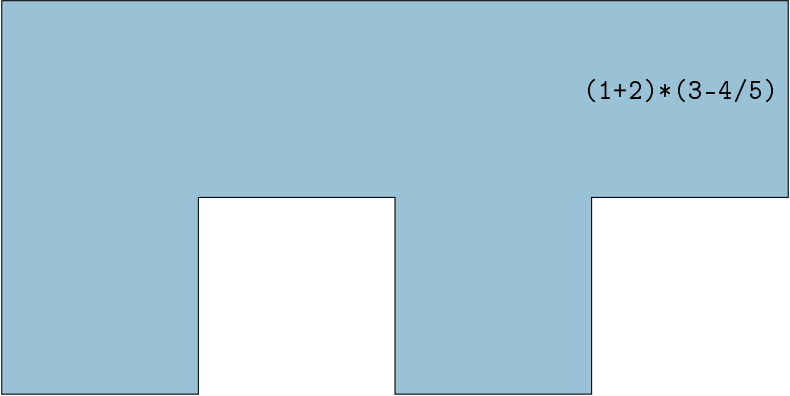
За преобразуване от инфиксен в обратен полски запис, за всеки пореден символ s :

- Ако s е цифра, прехвърляме я в стека с резултати
- Ако s е отваряща скоба, поставяме я в стека с операции
- Ако s е операция:
 - изваждаме от стека и прилагаме всички операции с приоритет по-висок или равен на s
 - поставяме s в стека
- Ако s е затваряща скоба, последователно изваждаме от стека и прилагаме всички операции до достигане на отваряща скоба

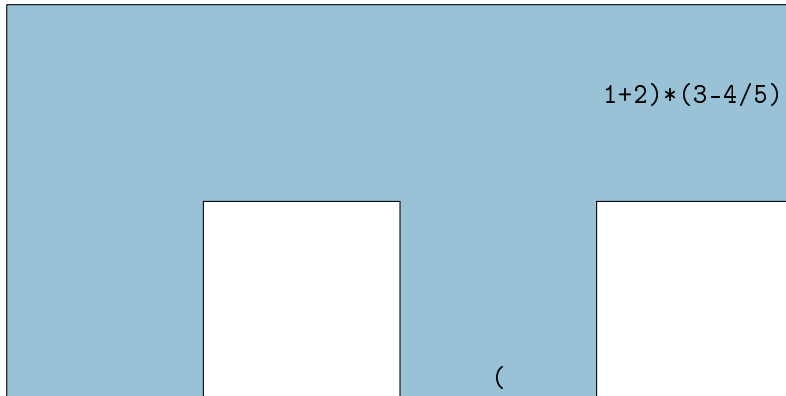
След приключване на входния низ, последователно изваждаме от стека и и прилагаме всички останали операции.

Крайният резултат е единственият елемент в стека с резултати.

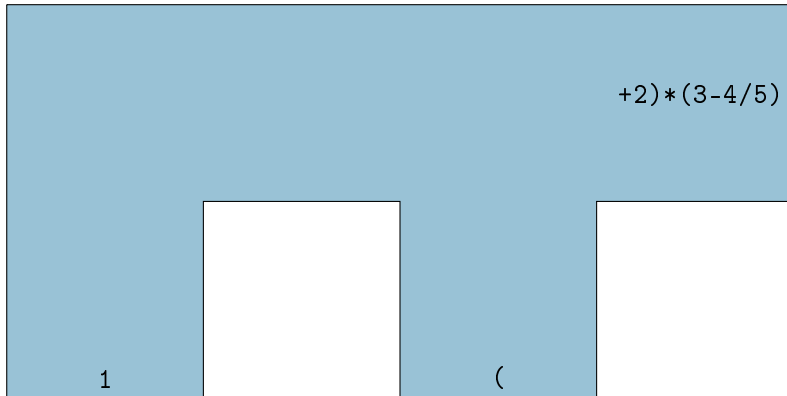
Пример: Директно пресмятане на израз


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

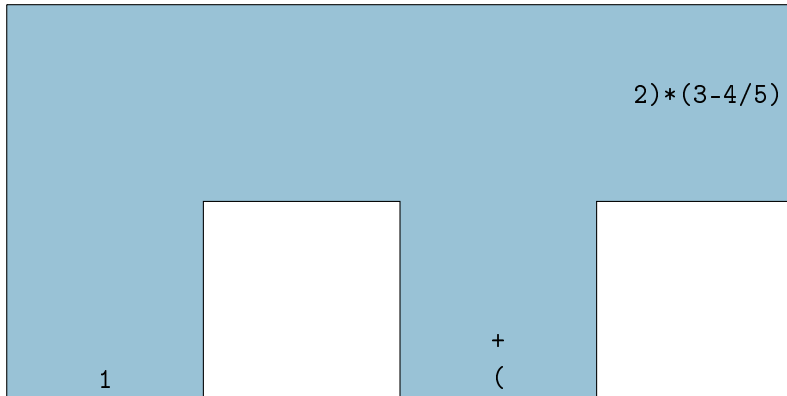
Пример: Директно пресмятане на израз



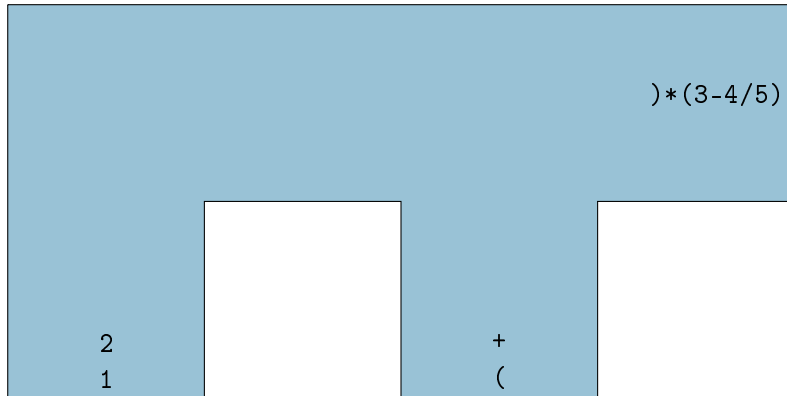
Пример: Директно пресмятане на израз



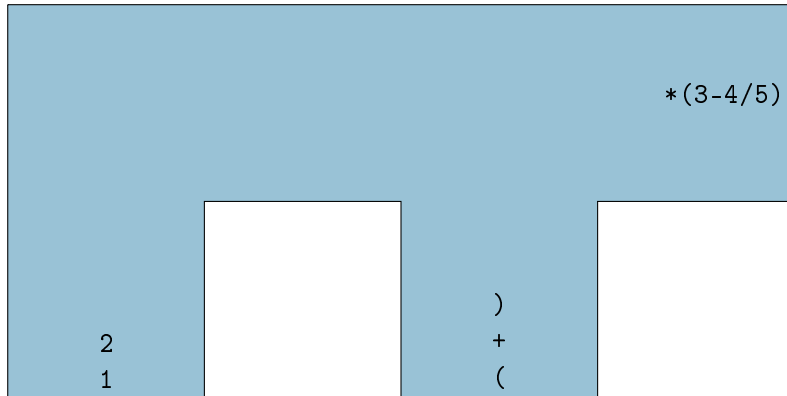
Пример: Директно пресмятане на израз



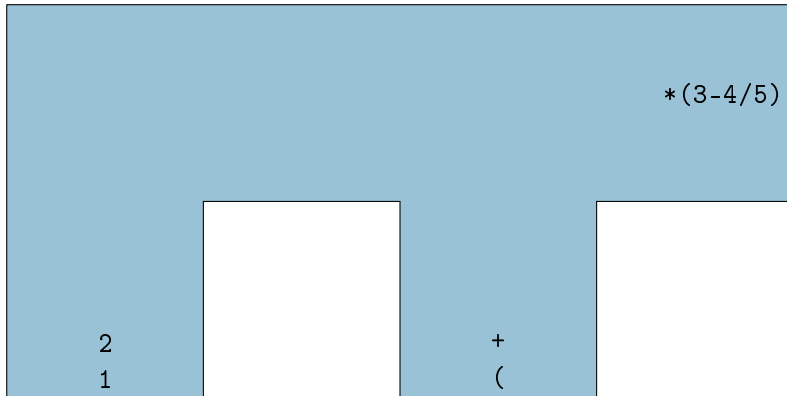
Пример: Директно пресмятане на израз



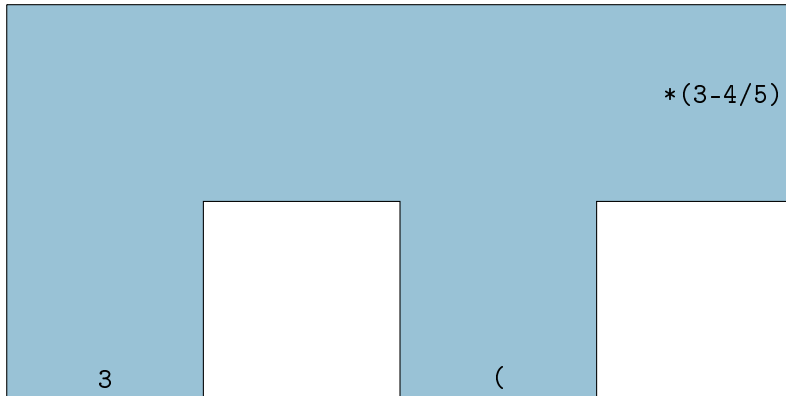
Пример: Директно пресмятане на израз



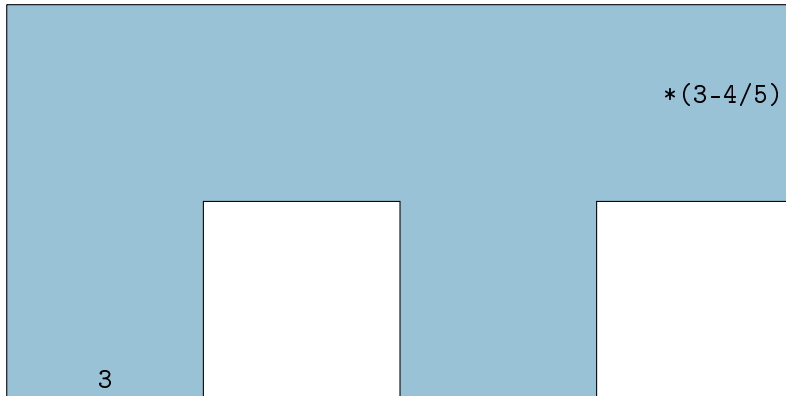
Пример: Директно пресмятане на израз



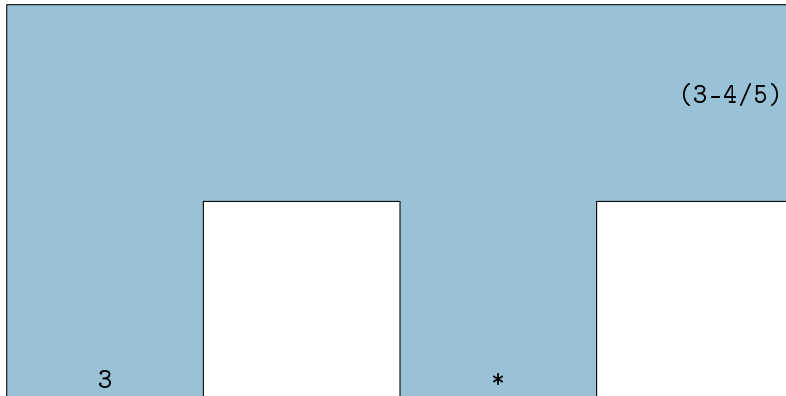
Пример: Директно пресмятане на израз



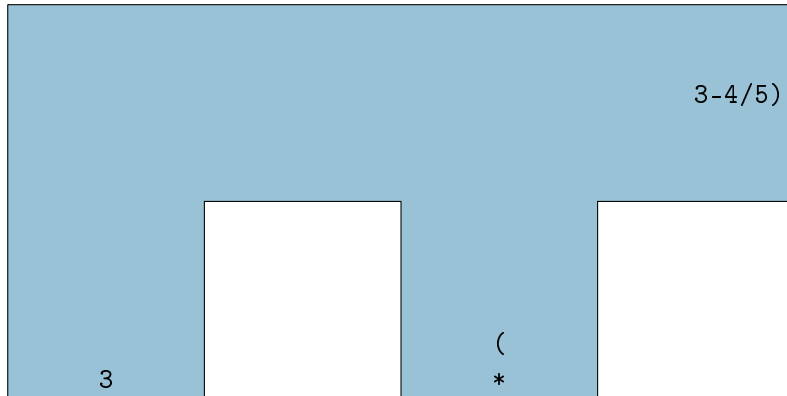
Пример: Директно пресмятане на израз



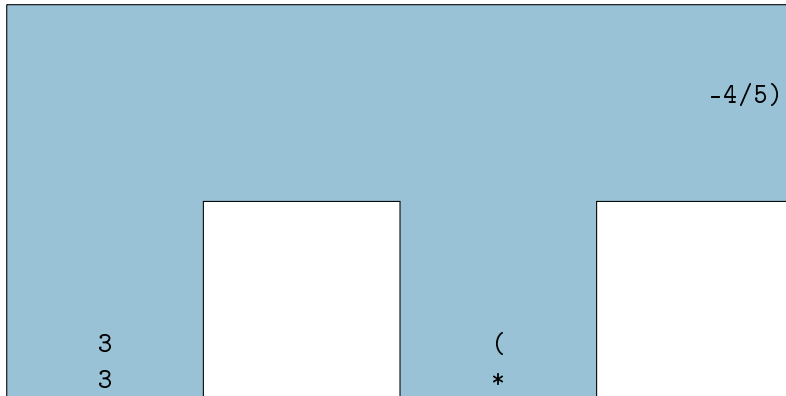
Пример: Директно пресмятане на израз



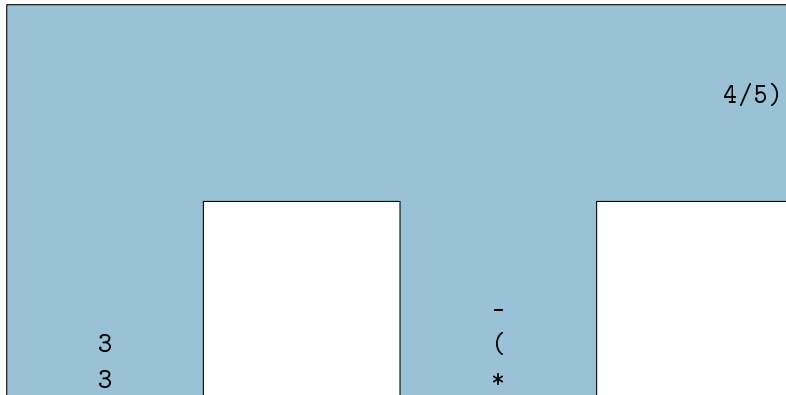
Пример: Директно пресмятане на израз



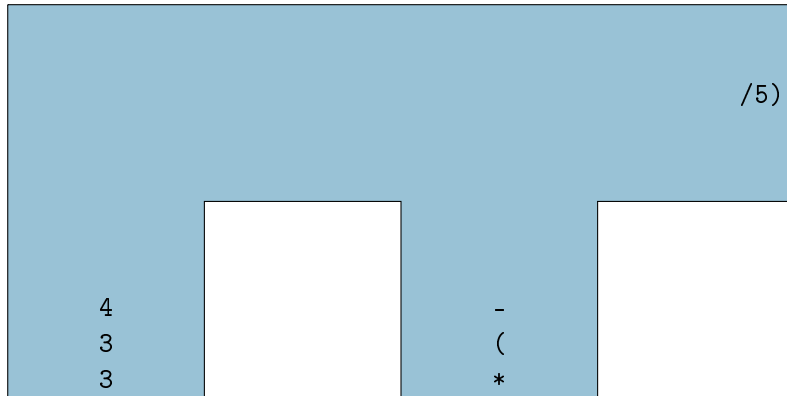
Пример: Директно пресмятане на израз



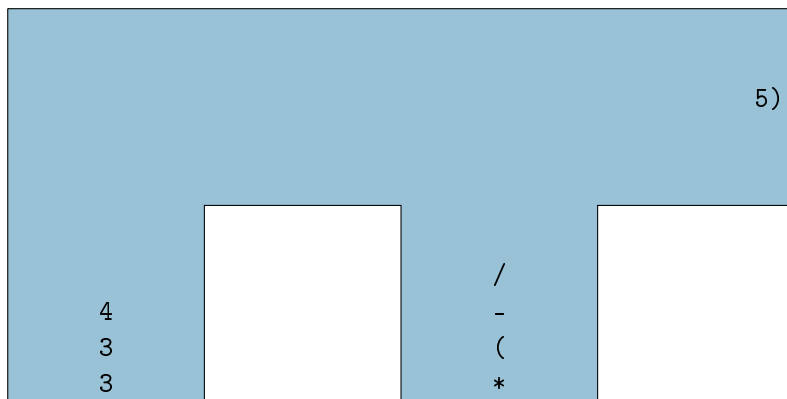
Пример: Директно пресмятане на израз



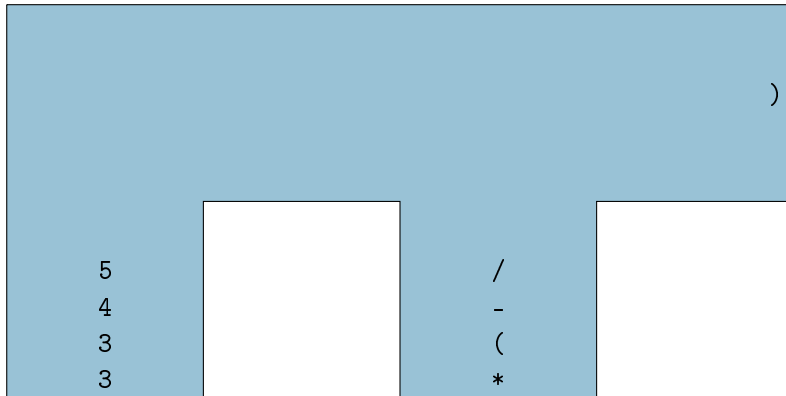
Пример: Директно пресмятане на израз



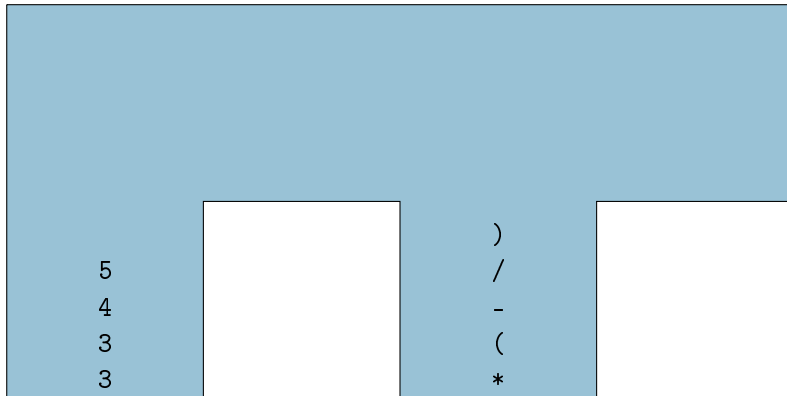
Пример: Директно пресмятане на израз



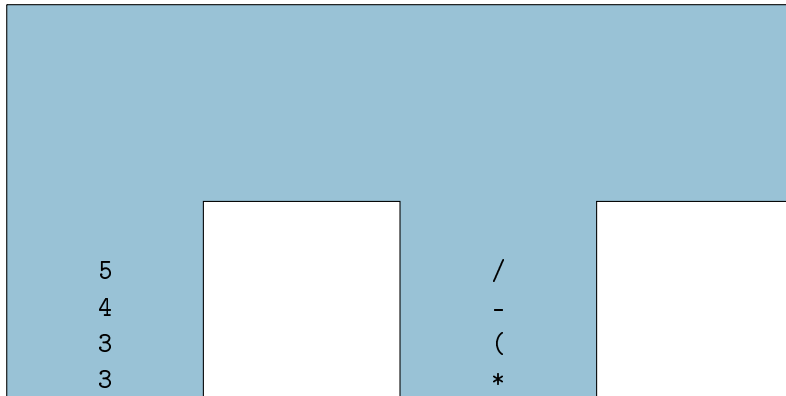
Пример: Директно пресмятане на израз



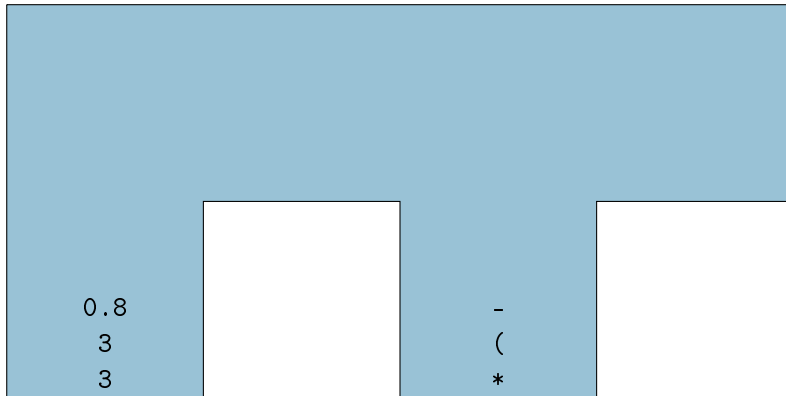
Пример: Директно пресмятане на израз



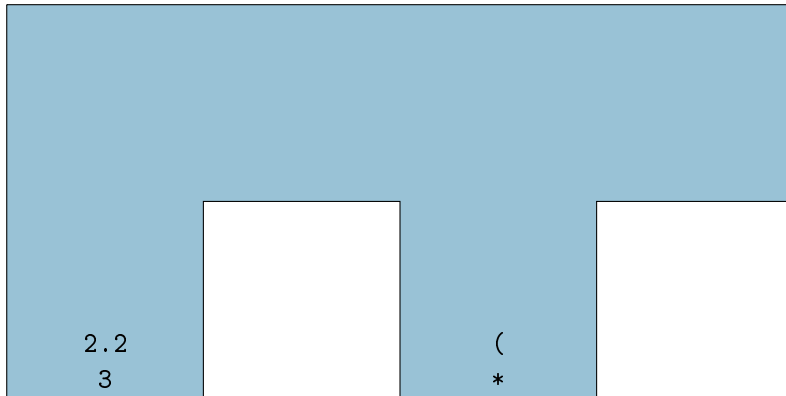
Пример: Директно пресмятане на израз



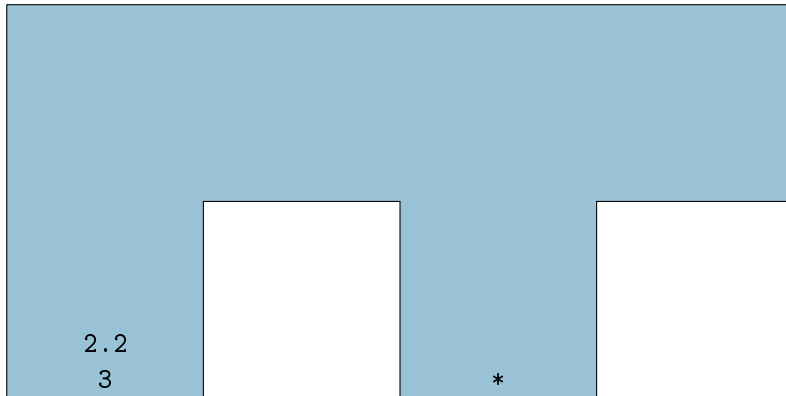
Пример: Директно пресмятане на израз



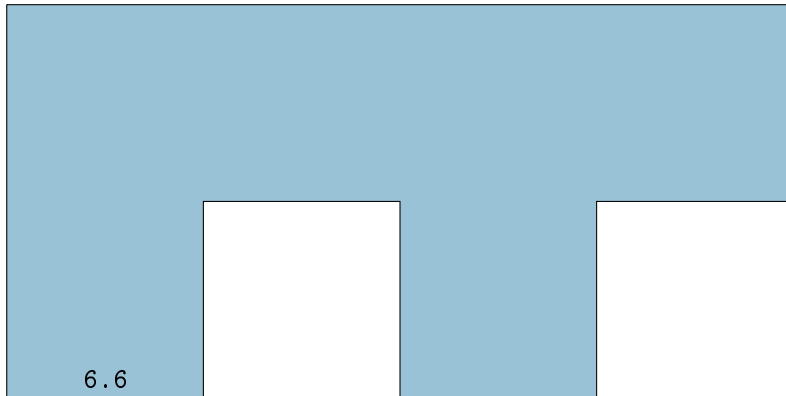
Пример: Директно пресмятане на израз



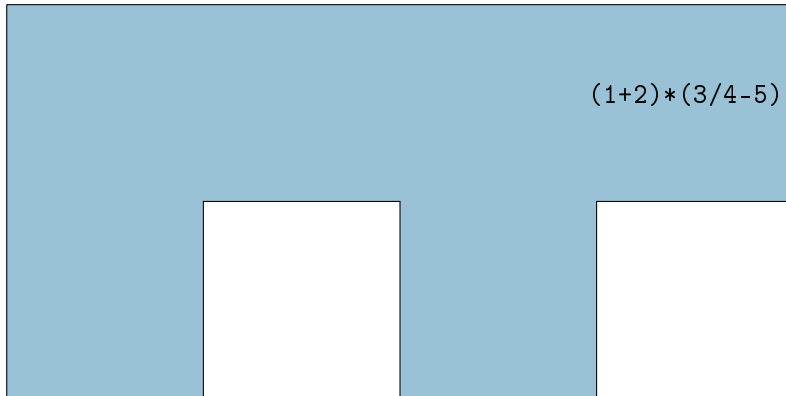
Пример: Директно пресмятане на израз



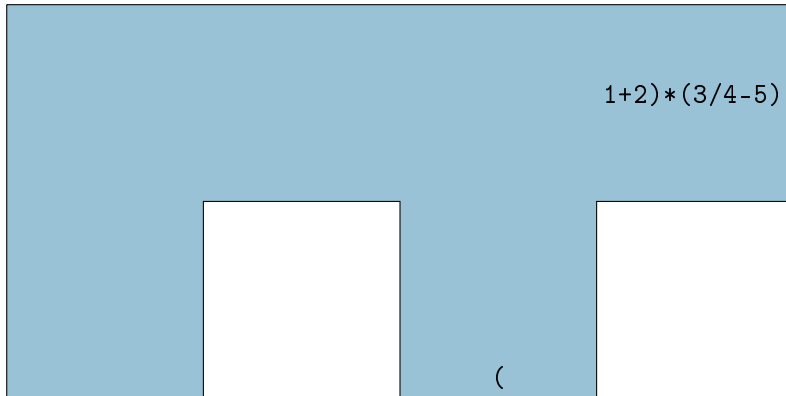
Пример: Директно пресмятане на израз



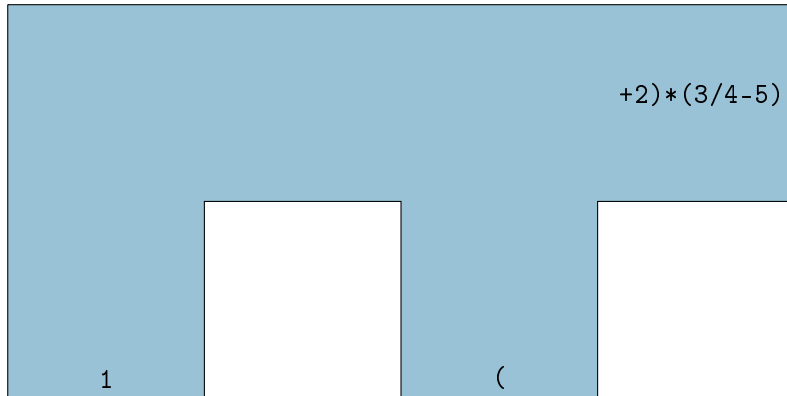
Пример 2: Директно пресмятане на израз



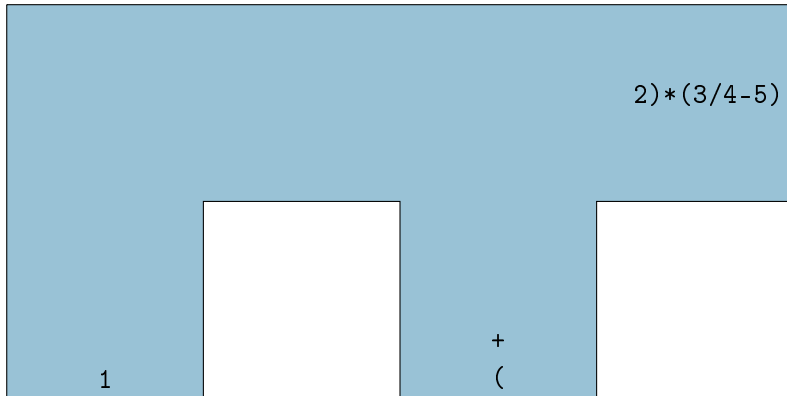
Пример 2: Директно пресмятане на израз



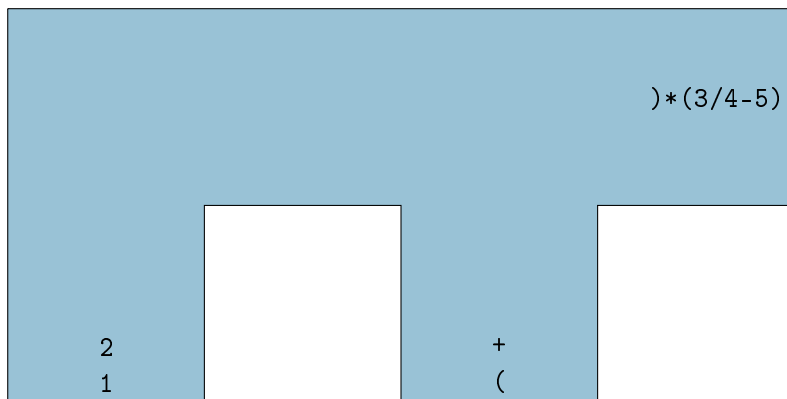
Пример 2: Директно пресмятане на израз



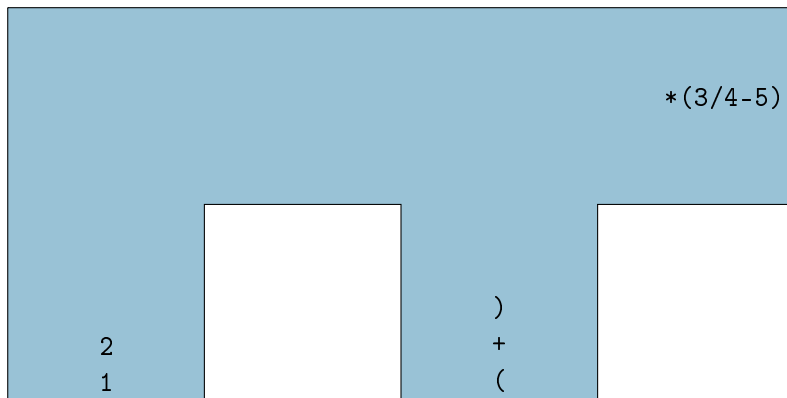
Пример 2: Директно пресмятане на израз



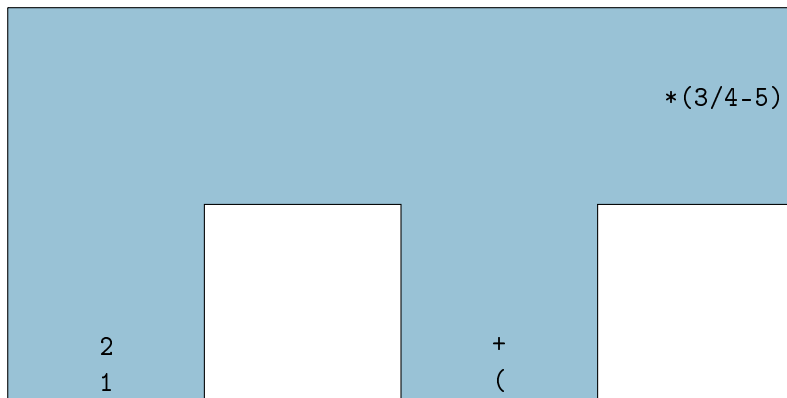
Пример 2: Директно пресмятане на израз



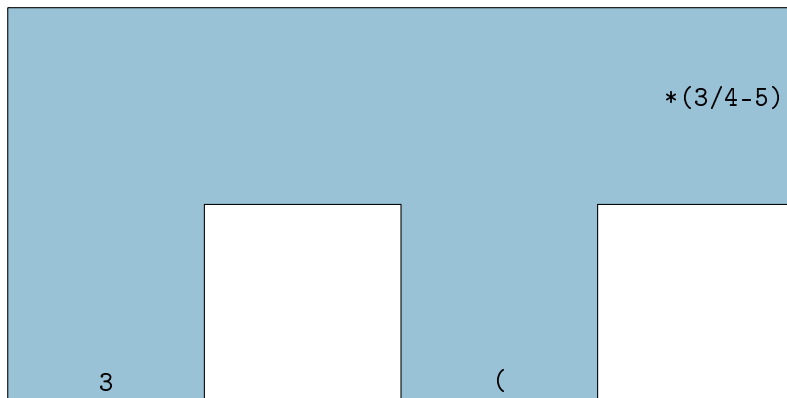
Пример 2: Директно пресмятане на израз



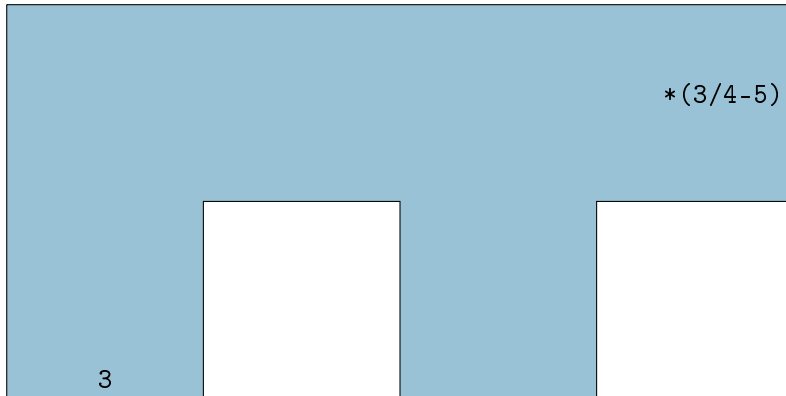
Пример 2: Директно пресмятане на израз



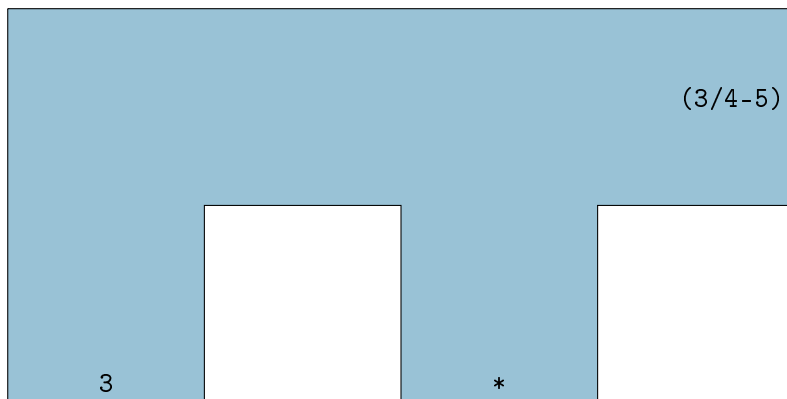
Пример 2: Директно пресмятане на израз



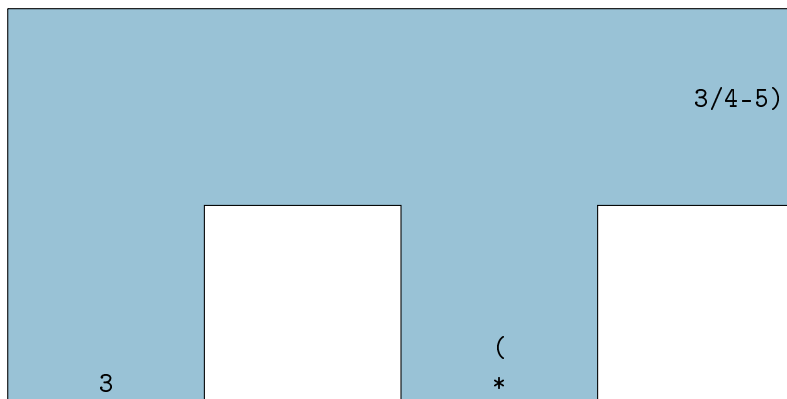
Пример 2: Директно пресмятане на израз



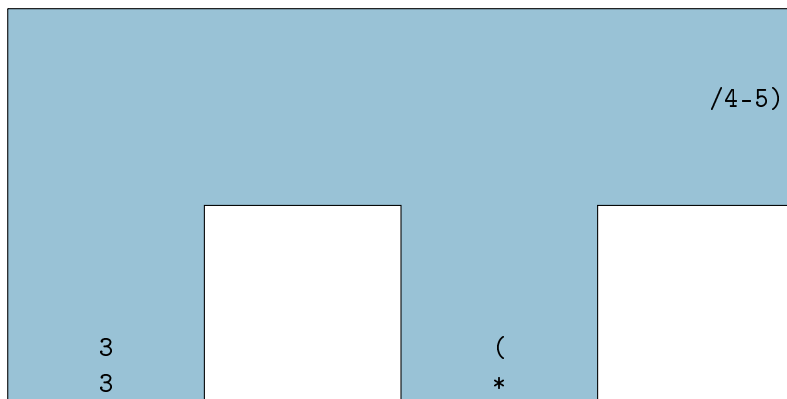
Пример 2: Директно пресмятане на израз



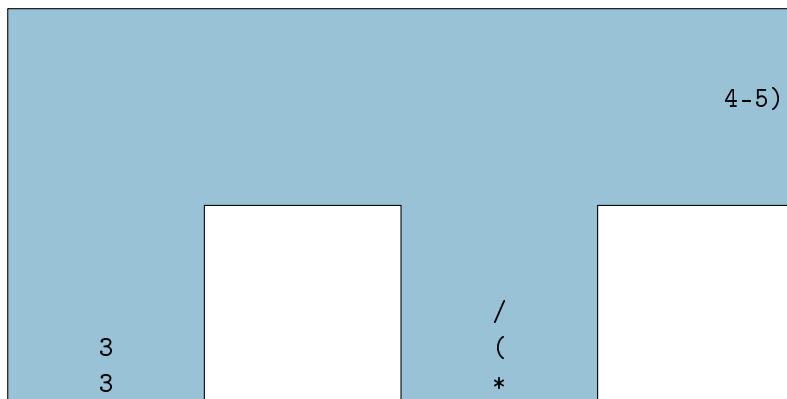
Пример 2: Директно пресмятане на израз



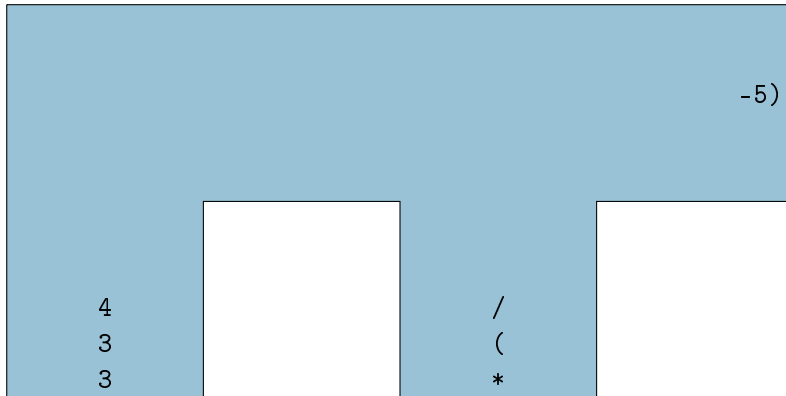
Пример 2: Директно пресмятане на израз



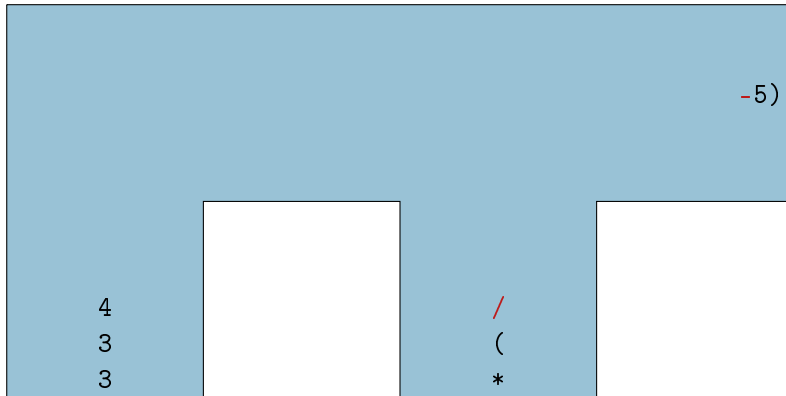
Пример 2: Директно пресмятане на израз



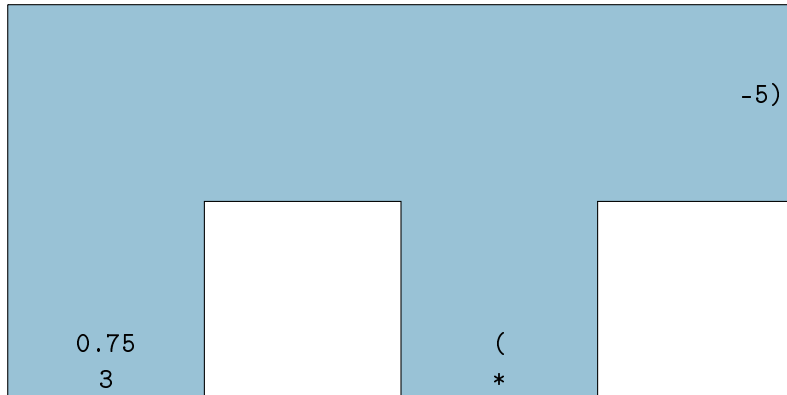
Пример 2: Директно пресмятане на израз



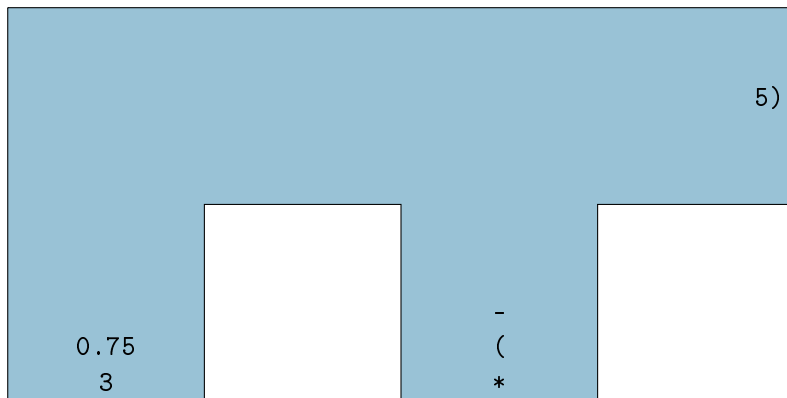
Пример 2: Директно пресмятане на израз



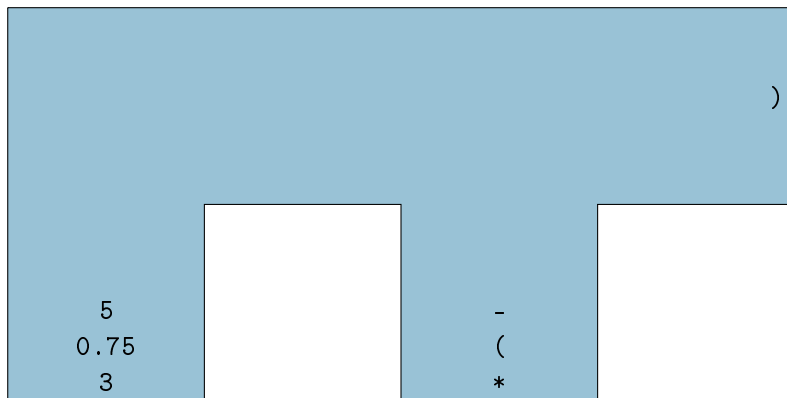
Пример 2: Директно пресмятане на израз



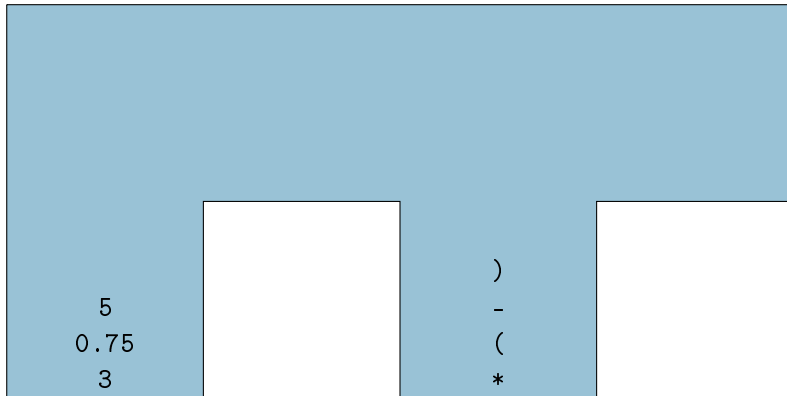
Пример 2: Директно пресмятане на израз



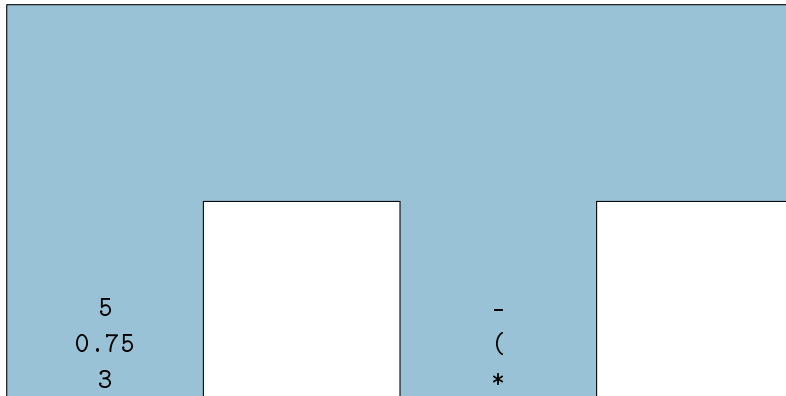
Пример 2: Директно пресмятане на израз



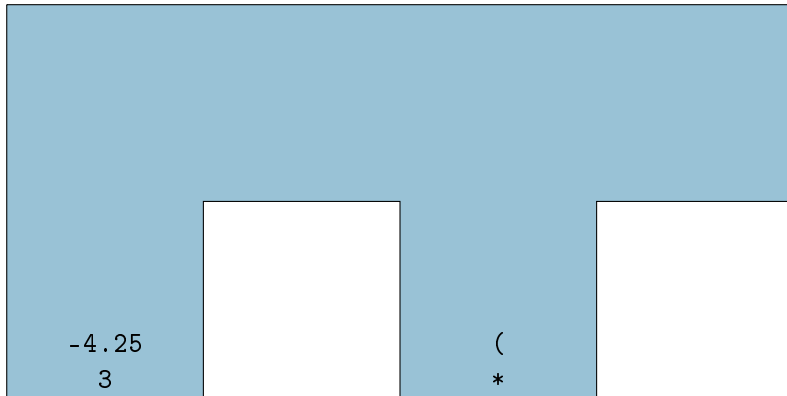
Пример 2: Директно пресмятане на израз



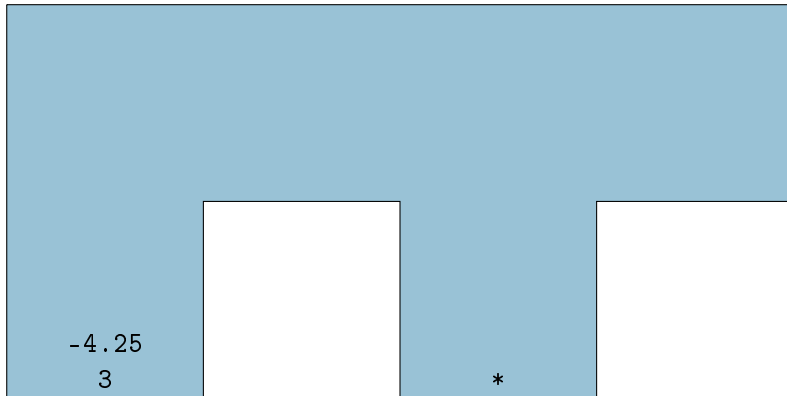
Пример 2: Директно пресмятане на израз



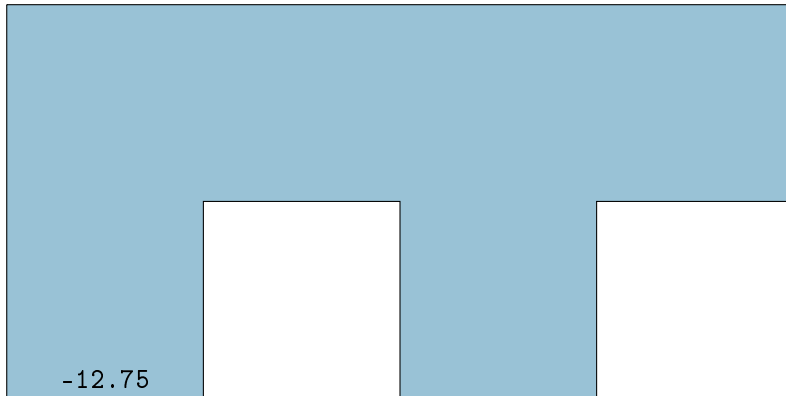
Пример 2: Директно пресмятане на израз



Пример 2: Директно пресмятане на израз



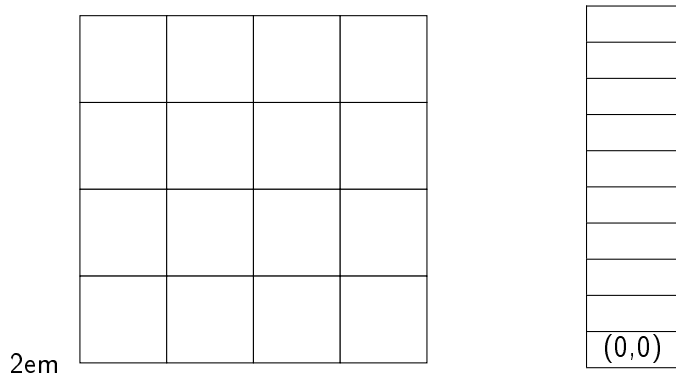
Пример 2: Директно пресмятане на израз



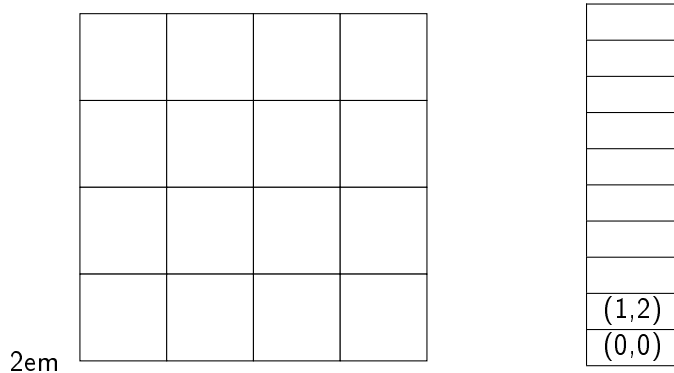
Симулиране на рекурсия

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

Пример: ход на коня

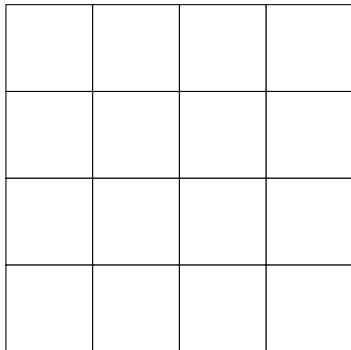


Пример: ход на коня



Пример: ход на коня

2em



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

Пример: ход на коня

2em

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

Пример: ход на коня

2em

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

Пример: ход на коня

2em

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

Пример: ход на коня

2em

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

Пример: ход на коня

2em

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

Пример: ход на коня

2em

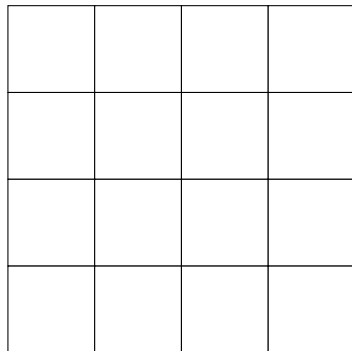
(0,2)
(2,1)
(1,3)
(3,2)
(2,0)
(1,2)
(0,0)

Пример: ход на коня

2em

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

Пример: ход на коня



2em

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

`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
- `==, !=, <, >, <=, >=` — лексикографско сравнение на два стека