

Separator (programowanie)

Separator (programowanie) – rodzaj ogranicznika zdefiniowanego w składni określonego języka programowania i stanowiącego element kodu źródłowego rozdzielający w ciągu znaków kodu źródłowego poszczególne jednostki leksykalne.

Spis treści

Funkcje separatora

Rodzaje separatorów

Separatory w wyrażeniach

Separatory w językach programowania

Zobacz też

Bibliografia

Funkcje separatora

Kod źródłowy składa się na podstawowym poziomie z ciągu znaków, jednak z punktu widzenia programisty i programu translatora, z jednostek leksykalnych, jako podstawowych jednostek tekstu mających określone znaczenie (interpretację) w danym języku. Aby poszczególne jednostki leksykalne mogły zostać przez translator wyodrębnione z ciągu znaków stanowiącego kod źródłowy, muszą zostać rozdzielone ustalonymi w definicji języka jednostkami nazywanymi separatorami lub ogranicznikami.

Tak więc separatory w swojej podstawowej funkcji umożliwiają podział tekstu na jednostki leksykalne. Oprócz tej funkcji, separator może:

- stanowić pewną jednostkę leksykalną o określonym znaczeniu w kodzie źródłowym (np. operator),
- stanowić znak interpunkcyjny wyodrębniający większe jednostki kodu o określonym znaczeniu (np. terminator lub separator instrukcji, separator listy itd.),
- nie mieć znaczenia w kodzie poza funkcją separatora, ale może zostać wprowadzony nadmiarowo w celu formatowania tekstu kodu źródłowego (np. kilka spacji na początku wiersza lub znak tabulacji, w celu wymuszenia wcięcia na wydruku kodu źródłowego)
- pełnić wyłącznie funkcję separatora.

Pewne pary jednostek leksykalnych nie wymagają rozdzielenia separatorem. Na przykład identyfikator zmiennej i operator (który sam stanowić może separator) zapisany w wyrażeniu w większości języków nie musi być w typowym przypadku rozdzielany separatorem, gdyż same zasady notacji tych jednostek leksykalnych umożliwiają ich wyodrębnienie z kodu źródłowego.

Rodzaje separatorów

Różne języki programowania mogą mieć zdefiniowany różny zestaw separatorów i własne zasady ich stosowania. Do typowych separatorów należą:

- białe znaki
 - spacja
 - znak nowej linii
 - znak powrotu karetki
 - znak tabulacji
- komentarze
- operatory

- nawiasy
 - w wyrażeniach
 - w indeksowaniu struktur danych
 - w specyfikacji list parametrów lub argumentów podprogramów
 - w zapisie wartości złożonych (np. tablicowych, rekordowych, zbiorowych, listowych itd.)
- znaki interpunkcyjne
 - terminator instrukcji (np. średnik ";" w języku C)
 - separator instrukcji (np. średnik ";" w języku Pascal)
 - separator listy (parametrów, argumentów, literałów prostych w zapisie wartości złożonych, np. przecinek ",")

Separatory w wyrażeniach

W większości uniwersalnych języków programowania wyrażenia mogą być zapisywane w notacji zbliżonej do matematycznej, bez konieczności stosowania wewnątrz zapisu wyrażeń dodatkowych znaków rozdzielających poszczególne jednostki leksykalne występujące w wyrażeniu, tj. literałów (np. liczbowych, znakowych, łańcuchowych i innych), identyfikatorów zmiennych, wywołań funkcji, operatorów itd. Z tego względu funkcję separatorów przejmują zastosowane w wyrażaniu operatory i nawiasy, jak np.:

```
a:=d+15*func_t(b-c,x);
```

gdzie znak operatora "+" jest separatorem dla identyfikatora "d" i literału liczbowego "15", a nawias otwierający "(" dla identyfikatora funkcji "func_t" i identyfikatora zmiennej "b", itd. W pewnych przypadkach nie jest wymagane w ogóle rozdzielanie elementów wyrażenia separatorem, znany jest przykład w języku C dotyczący operatorów ++ i -- w połączeniu z operatorami + i -, np.:

```
a=b+++c;
```

W powyższym przykładzie wyrażenia zapisano ciąg znaków "+++", który w języku C nie został zdefiniowany. Zdefiniowane są za to operator + i operator ++. Ponieważ nie zastosowano żadnego separatora, to interpretacja tego napisu zostanie przeprowadzona zgodnie z zasadą przyjętą w tym języku, że w tekście kodu źródłowego za kolejną jednostkę leksykalną przyjmuje się najdłuższy ciąg znaków, który może taką jednostkę stanowić. Oznacza to, że powyższy zapis zostanie zinterpretowany jak zapis:

```
a=(b++)+c;
```

Aby zapisać to wyrażenie jako

```
a=b+(++c);
```

oprócz powyższej konstrukcji z nawiasami można też zastosować separator, np. spację:

```
a=b+ ++c;
```

a nawet mało czytelny separator w formie komentarza pustego:

```
a=b/**/++c;
```

Separatory w językach programowania

języki programowania	białe znaki						komentarze	operatory	nawiasy	znaki interpunkcyjne
	spc=32	tab=9	nl=10	cr=13	inne	w wyrażeniach				
<u>Forth</u>	+	-	-	+	null=0	wymagane	-	-	-	-
<u>Turbo Pascal</u>	+	+	+	+	FF=12	dopuszczalne	{ }, (* *)	+	() , []	<ul style="list-style-type: none"> separator instrukcji: <code>;"</code> separator list: <code>;"</code> terminator programu, modułu: <code>;"</code>
<u>Turbo Prolog</u>	+	+	+	+	-	dopuszczalne	<code>/* */</code>	+	() , []	, ; :
<u>Snobol</u>	+	+	+	+	-	wymagane	-	-	()	; :

Uwaga:

- + : oznacza możliwość stosowania jako separatora
- - : brak możliwości stosowania jako separatora

Zobacz też

- jednostka leksykalna

Bibliografia

1. Bielecki Jan, *Język FORTH*, Wydawnictwa Naukowo-Techniczne, Warszawa 1988, seria: Mikrokomputery, ISBN 83-204-0930-6
2. Bielecki Jan, *Turbo C z grafiką na IBM PC*, Wydawnictwa Naukowo-Techniczne, Warszawa 1990, seria: Mikrokomputery, ISBN 83-204-1101-7
3. Marciniak Andrzej, *Borland Pascal 7.0*, Nakom, Poznań 1994 r., ISBN 83-85060-53-7, ISSN 0867-6011
4. Nicholls John E., *Struktura języków programowania*, Wydawnictwa Naukowo-Techniczne, Warszawa 1980, seria: Informatyka, ISBN 83-204-0246-8
5. Ruszczyk Jan, *Poznajemy FORTH*, SOETO Warszawa 1987 r., seria: Informatyka mikrokomputerowa
6. Szajna Janusz, Adamski Marian, Kozłowski Tomasz, *Turbo Prolog. Programowanie w języku logiki.*, Wydawnictwa Naukowo-Techniczne, Warszawa 1991, seria: Mikrokomputery. ISBN 83-204-1395-8

Źródło: „[https://pl.wikipedia.org/w/index.php?title=Separator_\(programowanie\)&oldid=51346735](https://pl.wikipedia.org/w/index.php?title=Separator_(programowanie)&oldid=51346735)”

Tę stronę ostatnio edytowano 26 gru 2017, 11:51. Tekst udostępniany na licencji Creative Commons: uznanie autorstwa, na tych samych warunkach (<http://creativecommons.org/licenses/by-sa/3.0/deed.pl>), z możliwością obowiązywania dodatkowych ograniczeń. Zobacz szczegółowe informacje o warunkach korzystania (http://foundation.wikimedia.org/wiki/Warunki_korzystania).