

ЛЕКЦИЯ 15 МАКРОАПАРАТ

- ⌚ **Необходимост и същност**
- ⌚ **Видове макроапарат**
- ⌚ **Макродефиниции**
- ⌚ **Макроизвиквания**
- ⌚ **Макроразширения**
- ⌚ **Процедури и макроси**

ПРОГ_15

1/18

НЕОБХОДИМОСТ

Проблемите, които се разглеждат в тази лекция **не са свързани с програмирането**.

Те са свързани с писането на текст.

Написването на **голям текст** изисква **доста време**, което често не е на разположение.

Има само един начин за **ускоряване на писането**: използване на **съкращения**.

Съкратеният текст **се възприема по-лесно** от тези, които **знаят съкращенията**, но **не може да се прочете от останалите (не знаещите)**.

ПРОГ_15

2/18

СЪЩНОСТ

Съкратеният текст **би станал четим** за всеки, **чрез списък на използваните съкращения**, стига той бъде добавен към написаното.

Още **по-добро решение** е чрез този списък **съкращенията да бъдат премахнати**, но тогава **ще се загуби лекотата** на четене.

Това е и **идеята на макроапарата**: бързо писане и леко четене чрез **определяне и използване на съкращения**.

Програмите са частен случай на текст, но се пишат **от хора, умеещи да програмират**.

ПРОГ_15

3/18

ТЕРМИНОЛОГИЯ

Гръцката дума **макро** (**μακροζ**) означава **голям**.

От гръцки тя преминава **в английския език** и **на програмистки жаргон** macro (мн. ч. macros) **означава съкращение** (съкратен запис).

Директното използване на **макро в българския език** ще доведе до **проблеми при мн. ч.**: **макри** хич не звучи добре на български.

Затова на български за **ед. ч.** се използва английското мн. ч. – **макрос**, чието българско **мн. ч. – макроси**, вече звучи доста добре.

ПРОГ_15

4/18

ДРУГИ ТЕРМИНИ

Макроапарат е (програмен) механизъм, осигуряващ дефиниране и използване на съкращения (макроси).

Макродефиниция е начина за определяне кое съкращение какво ще означава (кой макрос с какъв текст ще се заменя).

Макроизвикване е използването на съкращение (макрос) в написания текст.

Макроразширение е текстът, който заменя използваното съкращение въз основа на неговата дефиниция.

ПРОГ_15

5/18

ВИДОВЕ МАКРОАПАРАТ

Макроапаратът може да бъде реализиран и използван по два начина:

- ❶ **Оперативен (диалогов)** път: щом завърши писането на съкращение, то незабавно се заменя с желаня несъкратен текст. Използва се при текстообработка.
- ❷ **Пакетен (опосредствен)** път: текст, който съдържа дефиниции и съкращения се обработва от програма, която премахва използваните съкращения. Използва се при програмиране.

ПРОГ_15

6/18

СРАВНЕНИЕ

ОПЕРАТИВЕН ПЪТ

😊 може да се използва и за премахване на правописни грешки;

😊 не е необходим компютър.

😞 затруднява четенето: съкращенията намаляват обема и подобряват възприемането.

ПАКЕТЕН ПЪТ

😊 не е нужно да се чете разгънатият текст.

😊 облекчава четенето: малък обем, бързо възприемане.

😊 многократна замяна.

😞 реализира се само при използване на компютър, защото е необходима специална програма.

ПРОГ_15

7/18

МАКРОАПАРАТ И ПРОГРАМИРАНЕ

Езиците от ниско равнище (асемблери) изискват много писане, и при тях много често има възможност за използване на макроапарат (макроасемблери).

При езиците от високо равнище макроапаратът е рядкост (PL/1, Си, Си++). По-често за сигурност, удобство и еднаквост част от текста на програмата може да бъде изнесена „пред скоби“ (Кобол и др.).

ПРОГ_15

8/18

МАКРОАПАРАТЪТ НА СИ

От използваните в примерите езици **само Си притежава макроапарат.**

Всеки **транслатор** от **Си** е длъжен **първо** да **елиминира съкращенията** в текста и **едва след това да преведе** получената програма.

Частта, която обслужва **макроапарата**, е известна като **предпроцесор** на **Си**.

Елементите на макроапарата, обслужван от предпроцесора, се записват **на отделни редове**, които **започнат със знака диез (#)**.

ПРОГ_15

9/18

КОПИРАНЕ НА ТЕКСТ

В много езици, чиито транслатори осигуряват разделна компилация, **има необходимост различни програмни части да съдържат еднакъв текст** (дефиниции и др.).

Вместо този текст да се дублира във всяка програмна част, **по-удобно е** той да бъде записан в **отделен файл** и всяка програмна част да посочва **включване** на този файл.

Така при програмиране **имаме:**

- 1 **гаранция, че текстът ще бъде еднакъв;**
- 2 **удобство при модификация.**

ПРОГ_15

10/18

ПРИМЕР: СИ

Предварително подготвен текст се включва чрез директивата на предпроцесора **# include** <знак> <име на файл> <знак> ➡

Знаците, които ограждат името на файла, **могат да бъдат <> или " "**, и **определят справочника**, в който е описан този файл: стандартен или текущия.

По традиция на езика Си **файловете**, които съдържат прототипи на функции, полезни макродефиниции и др. **имат разширение .h** от думата заглавие (**header**).

ПРОГ_15

11/18

МАКРОДЕФИНИЦИЯ

Макродефиницията определя името, чрез което ще се използва един **макрос**, и **текстът**, с който ще бъде заменян той.

Фиксираната замяна на един текст с друг **носи доста ползи.**

Те се повишават многократно, ако можем да **параметризираме** заменящия текст.

Във втория случай **заедно с името на макроса** се определят и **имената**, с които ще се отбелязват **параметризираните участъци.**

ПРОГ_15

12/18

ПРИМЕР: СИ

Макрос се дефинира с директива **define**:

#define <име> <заменящ текст> ↵

#define <име> (<формални параметри>)
<заменящ текст> ↵

Имената на **параметрите** в списъка се **разделят със запетая (,)**. Те са **произволен брой** и **отбелязват в заменящия текст** участъците, които подлежат на **промяна при всяко използване** на макроса.

Всяка **дефиниция** може да бъде **отменена** с

#undef <име> ↵

ПРОГ_15

13/18

МАКРОИЗВИКВАНЕ

Макроизвикване се нарича **използването на име на вече дефиниран макрос**.

При използване на **параметризирана макродефиниция** трябва да се зададат и **желаните фактически параметри**.

Всеки макроапарат има **собствени правила** за идентификация на макроизвикването и неговите фактически параметри.

Общото е, че **параметрите са текстове**, т. е. редици от знаци (знакови низове).

ПРОГ_15

14/18

МАКРОРАЗШИРЕНИЕ

Текстът, който се получава при замяна на макроизвикване с дефиницията на даден макрос се нарича **макроразширение**.

В написания текст се премахва името на използвания макрос и зададените като текстове **фактически параметри** и на тяхно място се поставя текстът, получен от **макродефиницията**, като в него на мястото на имената на **формалните параметри** се записват **низове – фактически параметри**.

ПРОГ_15

15/18

ПРАВИЛАТА НА СИ

Всеки ред, който не започва с **#** (т. е. не е директива на предпроцесора), **се преглежда за думи**, дефинирани като **макроси**.

Когато открито **име завършва със скоба ()** в скобите **се откриват** разделените със запетая **низове – фактически параметри**.

Извършва се замяна като **съответствието на параметрите е по позиция**.

Проверката се повтаря докато могат да се реализират замени.

ПРОГ_15

16/18

ПРОЦЕДУРИ И МАКРОСИ

При макроапарата на **Си** използването на макрос и функция **си** приличат външно.

Когато се използва **функция**:

- ❶ кодът е на едно място (**печалба на ОП**);
- ❷ използването изисква изпълнение на МИ за връзка с подпрограмата и за предаването на фактическите параметри (**загуба на време**).

Когато се използва **макрос**:

- ❶ модифицираният код се появява при всяко използване на макроса (**загуба на ОП**);
- ❷ използването не изисква допълнителни МИ и може да бъде оптимизиран на място (**бързина**).

прог_15

17/18

**БЛАГОДАРЯ ВИ
ЗА ВНИМАНИЕТО!**

**БЪДЕТЕ С МЕН И В
ПОСЛЕДНАТА ЛЕКЦИЯ,
КОЯТО ЩЕ НИ ОТВЕДЕ
В НЕВЕРОЯТНИЯ СВЯТ НА
УСЛОВНАТА
ТРАНСЛАЦИЯ**