Структура на програма в C

Част 3 - декларации, изрази, оператори, клаузи

Иван Георгиев, Христо Иванов, Христо Стефанов

Технологично училище "Електронни системи",
Технически университет, София

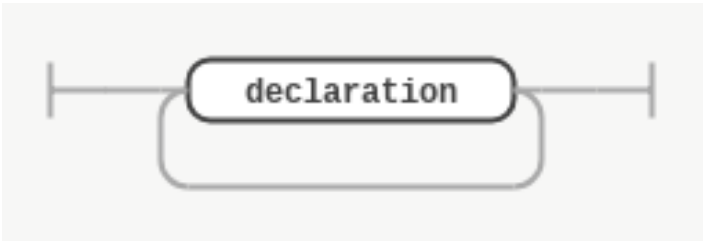
13 март 2019 г.

Съдържание

Съдържание

Декларации

- В C, за да се използват данни или подпрограми, те трябва първо да бъдат *декларирани*
- *Декларацията* служи за описване на различни атрибути, нужни на компилатора, свързани с определени данни или подпрограми
- Например, един от основните атрибути за данни е *типът данни*
- Програмите в C са поредица от декларации



Фигура 1: Синтаксис на програма в C

Декларации на променливи

- Данните в C се наричат *променливи*
- Променливите имат основни атрибути - идентификатор (име), тип, начална стойност.
- Ако началната стойност не е посочена, то тя е произволна

Фигура 2: Декларация на променлива. Непълна синтактична диаграма

- Съществуват и *временни променливи*, които нямат име, но имат тип и зададена начална стойност. Те не могат да бъдат декларирани или променяни и се създават от компилатора за запазване на междинни резултати

Съдържание

Изрази

- *Израз* (*expression*) е последователност от операции върху променливи, резултата от които е запазен във временна променлива
- За *стойност на израза* се счита стойността запазена във временната променлива
- За *тип на израза* се счита типа на временната променлива
- *Първичен израз* се нарича израз, който съдържа само една константа, един литерал или един идентификатор.
  - Ако съдържа идентификатор, стойността и типът на временната променлива са тези на променливата посочена от идентификатора
  - Ако съдържа константа/литерал, стойността и типът на временната променлива са тези на константата/литерала

Оператори

- Начинът, по който се обработват данни в C, е чрез *оператори*
- Операторите служат за обозначаване на операция върху една или повече *операнди*
- Операндите най-често са изрази[1]
- Операторите най-често се отбелязват с пунктуатор(и)
- Начинът на прилагане на оператор върху операнди варира

```
op1 + op2, -op1, op1 << op2, op1 >= op2,
op1 * op2, op1 / op2, sizeof(op1)
```

Фрагмент 1: Примери за оператори в C

[1] Например могат да бъдат и имена на типове, които се използват от операцията по определен начин

Оператори

- Броят на операндите зависи от операцията, която се обозначава от оператора
- Например:
  - операция събиране - 2 операнди
  - операция смяна на знак - 1 операнда
- Спрямо броя операнди операциите се разделят на *унарни* (1 операнда), *бинарни* (2 операнди) и *тернарни* (3 операнди)
- Някои операции съдържат *странични ефекти* - след изпълнението им освен резултат, може да се наблюдават промени в някои от операндите

Особености на оператори

- Съществуват оператори, които се отбелязват с един и същ пунктуатор и се различават само по броя операнди
- Например операторът за смяна на знак и операторът за изваждане се отбелязват с '-'

```
-op1, op1 - op2
```

Фрагмент 2: Оператор за смяна на знак, оператор за изваждане

Съдържание

Декларации на функции

- Подпрограмите в C се наричат *функции*
- Функциите имат следните основни атрибути
  - идентификатор (име)
  - типът на данните, които връща функцията след изпълнението си
  - типове и имена на отделните данни (*параметри*), върху които работи функцията
  - *блок* за *тяло* на функцията

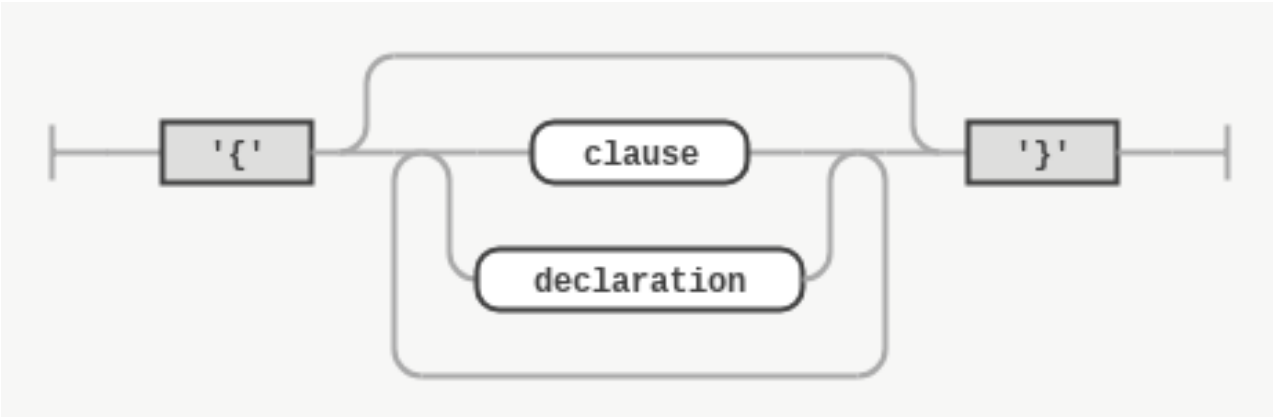
Фигура 3: Декларация на функция. Непълна синтактична диаграма

Клаузи

- *Клауза* (*твърдение*, *clause*/*statement*) е фрагмент от код, който се изпълнява последователно
- Съществуват пет вида клаузи
  - сложни клаузи
  - клаузи за изрази
  - клаузи за избор
  - клаузи за цикъл
  - клаузи за преход

Сложни клаузи. Блокове

- *Сложна клауза* (*блок*, *compound clause/block*) е клауза, която може да съдържа в себе си поредица от клаузи и/или декларации
- Сложна клауза, която не съдържа нито една клауза или декларация се нарича *празна сложна клауза/празен блок*



Фигура 4: Синтактична диаграма на сложна клауза (блок)

Клаузи за изрази

- *Клауза за израз* (*expression clause*) е клауза, която може да съдържа в себе си израз, завършващ с ';'
- Изразът може да бъде изпуснат, като в такъв случай клаузата се нарича *празна клауза*

Фигура 5: Синтактична диаграма на клауза за израз

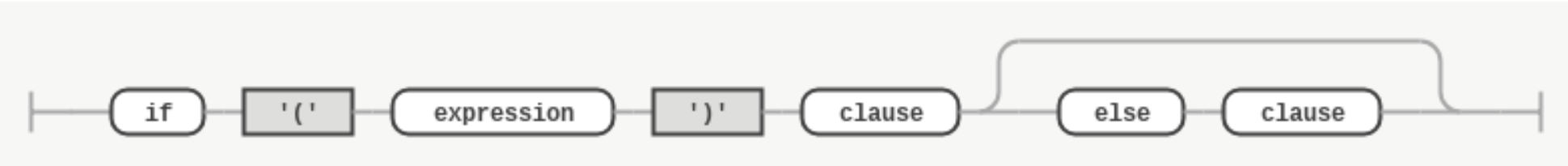
Клаузи за преход. Клауза за връщане на резултат

- *Клаузите за преход* променят реда на изпълнение на програмата
- Например *клаузата за връщане на резултат* прекратява изпълнението на тялото на функцията и оказва стойността на резултата от функцията

Фигура 6: Синтактична диаграма на клауза за връщане на резултат

Клаузи за избор. if клауза

- *Клаузите за избор* променят реда на изпълнение на програмата като избират между две клаузи, с които да се продължи изпълнението
- Например *if клаузата* избира на базата на стойността на някакъв израз (наречен *условие*) дали да изпълни една или друга клауза
- За да се избере първата клауза, стойността на условието трябва да бъде различна от 0. Ако стойността на условието е равна на 0 се изпълнява втората клауза, ако има такава



Фигура 7: Синтактична диаграма на if клауза

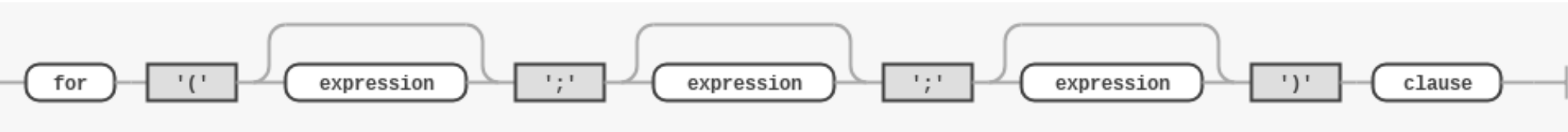
Клаузи за цикъл. while клауза

- *Клаузите за цикъл* променят реда на изпълнение на програмата като повтарят дадена клауза. Може дадената клауза да не бъде изпълнена нито веднъж
- Например *while клаузата* изпълнява дадена клауза (наречена *тяло*) докато стойността на някакъв израз (наречен *условие*) е различна от 0
- Ако стойността на условието първоначално е равна на 0, тялото не се изпълнява нито веднъж

Фигура 8: Синтактична диаграма на while клауза

for клауза

- *for клаузата* е клауза за цикъл. Първо се изпълнява израз (наречен *инициализация*), след което се изпълнява дадена клауза (наречена *тяло*) докато стойността на някакъв израз (наречен *условие*) е различна от `0`. Между изпълненията на клаузата се изпълнява друг израз (наречен *стъпка*)
- Ако стойността на условието първоначално е равна на `0`, тялото не се изпълнява нито веднъж
- Ако изразите за стъпка или инициализация са изпуснати, просто не се изпълняват
- Ако условието е изпуснато, винаги се влиза в тялото на `for` цикъла



Фигура 9: Синтактична диаграма на for клауза