



کامپیایر مقدمات

محسن هوشمند
دانشکده تکنولوژی اطلاعات و علم رایانه
دانشگاه تحصیلات تکمیلی علوم پایه زنجان

دسته‌بندی زبان‌ها

نسل اول: زبان‌های ماشین

نسل دوم: زبان‌های اسمبلی

نسل سوم: زبان‌های سطح بالاتر چون فرترن و کوبول و لیسپ و سی و سی++ و سی# و جاوا

نسل چهارم: زبان‌های کاربرد خاص

- نماد Nomad جهت تولید گزارش

- اسکیوال جهت مدیریت و جستجوی پایگاه داده

نسل پنجم: زبان‌های قید-محور و منطق‌محور پرولوگ و اُپی‌اس پنج

دسته بندی دیگر

اعلانی Declarative

- کامپیوتر «چه» انجام دهد

دستوری (امری) Imperative

- کامپیوتر «چگونه» انجام دهد

```
int g = 0;
```

```
int rt(int x) {  
    return x + 1;  
}
```

```
int ro(int x) {  
    g++;  
    return x + g;  
}  
  
wiki
```

شفافیت ارجاع referential transparency

- عدم تغییر در رفتار برنامه در صورت جانشینی عبارت با مقدار
- عدم اثر جانبی
- تابع محض
- خروجی تنها برابر با مقدار بازگشتی
- خروجی صرفا وابسته به آرگومان ورودی
- مشخص بودن کامل آرگومان‌ها قبل تولید خروجی
- کریس استراچی

- اعلانی انتزاع بیشتر
- اعلانی شبیه تر به نظر برنامه نویس و متفاوت از نگاه پیاده ساز
- بیشتر زبان‌ها دستوری
- به دلیل عملکرد بهتر

دسته بندی دیگر

اعلانی Declarative

- تابعی
- لیسپ/اسکیم، امال، هسکل
- جریان داده
- آی دی، ول
- منطق، قیدمحور
- پرولوگ، کاربرگها، اس کیوال

دستوری (امری) Imperative

- فن نویمن
- سی، آیدا، فرترن
- شی گرا
- اسمال تاک، ایفل، جاوا
- نبشتاری
- پرل، پایتون، پی اچ پی

دسته‌بندی دیگر

تابعی

- لیسپ/اسکیم، ام‌ال، هسکل
- بکارگیری مدل مبتنی بر تعریف بازگشتی توابع
- استفاده از «جبر لاندا»
- مبدع الونزو چرچ
- هم‌قدرت ماشین تورینگ
- برنامه: تابعی از ورودی به خروجی
- تعریف مبتنی بر توابع ساده‌تر
- با استفاده از فرایند تصفیه (بهبود)

جریان داده

- آی‌دی، ول
- مدل رایانش به مثابه جریان اطلاع (تکه‌ها) در میان رأس‌های توابع بدوی؟
- ذاتا موازی
- چکیده شدن رأس‌ها با دریافت تکه ورودی

دسته‌بندی دیگر

منطق، قیدمحور

- پرولوگ، کاربرگ‌ها، اس‌کیوال
- منطق محمولات یا مرتبه‌اول
- مدل رایانش به مثابه تلاشی جهت یافتن مقادیر برآورده‌کننده روابطی خاص
- استفاده از جستجویی هدف‌محور در میان فهرستی از قوانین منطقی
- بوده‌ها و کلاوزها
- نحو شامل متغیرها و ثوابت
- استنتاج‌محور

فن نویمن (رویه‌ای)

- سی، آیدا، فرترن
- معروف‌ترین و پراستفاده‌ترین
- ابزار اساسی رایانش: تغییر متغیرها
- زبان تابعی مبتنی بر عباراتی دارای مقدار
- زبان‌های فن‌نویمن مبتنی بر تخصیص‌ها که موثر بر رایانش‌های بعدی با تاثیر بر تغییر مقادیر حافظه

دسته بندی دیگر

شی گرا

- اسمال تاک، ایفل، جاوا
 - ریشه در سیمولا ۶۷
 - مشابه زبان معماری فن نویمن
 - مدل ساخت یافته تر و توزیعی تر حافظه و رایانش
 - تعامل بین اشیاء نیمه مستقل
 - رد و بدل پیام
 - به جای دیدن رایانش چون عملیات پردازنده روی حافظه
- ## هر شی
- دارای حالت داخلی مختص به خود
 - زیررویه های مدیریت حالت
- ## اسمال تاک
- مومن ترین!

دسته‌بندی دیگر

نیشتری

- پرل، پایتون، پی‌اچ‌پی

- چسب‌أهو

- تفسیر زمان اجرا

- bash و csh

- جهت کنترل وظیفه‌ها

- پی‌اچ‌پی و جاوااسکریپت

- تولید محتوای وب پویا

- لوا Lua

- جهت کنترل بازی‌های رایانه‌ای

- پرل و پایتون و روبی

- همه‌منظوره

دسته‌بندی دیگر - مقایسه

یافتن بزرگترین مقسوم‌علیه مشترک

- مثلاً a و b
- هم تاثیر بر طرز کد، هم نحوه تفکر برنامه‌نویس
- فن‌نوین
- برابری a و b را بررسی کن؛ اگر برابر بودندی یک را چاپ کن و پایان برنامه؛ وگرنه عدد بزرگتر را با تفاضل آن‌ها جانشین کن و مرحله قبل را تکرار کن.

```
int gcd(int a, int b) { // C
    while (a != b) {
        if (a > b) a = a - b;
        else b = b - a;
    }
    return a;
}
```

دسته‌بندی دیگر - مقایسه

یافتن بزرگترین مقسوم‌علیه مشترک

▪ تابعی: تاکید بر روابط ریاضی خروجی به ورودی‌ها

▪ ب‌م‌م a و b این‌گونه تعریف می‌شود:

▪ $a - 1$ وقتی a و b برابر باشند

▪ $a - 2$ ب‌م‌م b و $a - b$ اگر a بزرگتر از b

▪ $a - 3$ ب‌م‌م a و $b - a$ اگر a کوچکتر از b ، و عدد c وجود دارد به‌طوری که $c = a - b$ و $\text{gcd}(c, a, b)$ درست است

▪ جهت یافتن ب‌م‌م ادامه و ساده‌کردن تعریف تا رسیدن به پایان

```
let rec gcd a b = (* OCaml *)
  if a = b then a
  else if a > b then gcd b (a - b)
  else gcd a (b - a)
```

دسته‌بندی دیگر - مقایسه

یافتن بزرگترین مقسوم‌علیه مشترک

▪ منطقی

▪ گزاره $\text{gcd}(a, b, g)$ درست است اگر

▪ $a - 1$ و b و g برابر باشند

▪ $a - 2$ بزرگتر از b ، و عدد c وجود دارد به طوری که $c = b - a$ و $\text{gcd}(c, b, g)$ درست است

▪ $a - 3$ کوچکتر از b ، و عدد c وجود دارد به طوری که $c = a - b$ و $\text{gcd}(c, a, g)$ درست است

▪ جهت یافتن دو عدد داده‌شده به دنبال عدد g ، و c -های مختلف بطوری که یکی از قوانین فوق درستی $\text{gcd}(a, b, g)$ را نشان دهد

$\text{gcd}(A, B, G) :- A = B, G = A. \% \text{ Prolog}$

$\text{gcd}(A, B, G) :- A > B, C \text{ is } A - B, \text{gcd}(C, B, G).$

$\text{gcd}(A, B, G) :- B > A, C \text{ is } B - A, \text{gcd}(C, A, G).$

دسته بندی دیگر - مقایسه

```
int gcd(int a, int b) { // C
    while (a != b) {
        if (a > b) a = a - b;
        else b = b - a;
    }
    return a;
}
```

یافتن بزرگترین مقسوم علیه مشترک

```
let rec gcd a b = (* OCaml *)
if a = b then a
else if a > b then gcd b (a - b)
else gcd a (b - a)
```

```
gcd(A,B,G) :- A = B, G = A. % Prolog
gcd(A,B,G) :- A > B, C is A-B, gcd(C,B,G).
gcd(A,B,G) :- B > A, C is B-A, gcd(C,A,G).
```

چرائی مطالعه زبان‌های برنامه‌نویسی

کنجکاوی

انتخاب درست در موعد

شبهات‌ها جهت یادگیری

احیاناً بهبود قابلیت‌ها

▪ شرکت در پیشرفت علم و فن

Com.pil.er: 1- someone who collects different pieces of information to be used in a book, report, or list.

2- (*technical*) a set of instructions in a computer that changes a computer language known to the computer user into the form needed by the computer
2- Longman Dictionary

Com·pi·la·tion: 1- [*countable*] a book, list, record etc which consists of different pieces of information, songs etc.

2- [*uncountable*] the process of making a book, list, record etc from different pieces of information, songs etc.

com·pile: 1- to make a book, list, record etc, using different pieces of information, music etc The document was compiled by the Department of Health

2- (*technical*) to put a set of instructions into a computer in a form that it can understand and use

compile (1300-1400) Old French compiler, from Latin compilare “to seize together, steal”

منابع

[کوپر]

[ازدرها]

[فیشر]

[اسکات]

“Compiler Design Tutorial: What is, Types, Tools, Example,” <https://www.guru99.com/compiler-design-tutorial.html>

“C++ Program to implement Symbol Table,” <https://www.geeksforgeeks.org/cpp-program-to-implement-symbol-table/?ref=rp>

“Functional, Declarative, and Imperative Programming,” <https://stackoverflow.com/questions/602444/functional-declarative-and-imperative-programming>