



## درس طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

موضوع: نوع داده اولیه

ارائه: صادق سلیمانی

[www.Bioinformation.ir](http://www.Bioinformation.ir)

جلسه‌ی چهارم

### نوع عناوین جلسه

- یادآوری شش جزء اصلی هر کامپیوتر
- داده
  - شیء داده
  - نوع داده
  - پیچیدگی‌های پیاده‌سازی عملگرها بر نوع داده
  - نحوه‌ی پیاده‌سازی نوع داده
- اعلان
- کنترل نوع
- تبدیل نوع

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع  
یادآوری

● شش جزء اصلی یک کامپیوتر

1. Data
2. Primitive Operations
3. Sequence Control
4. Data Control
5. Storage Management
6. Operating Environment

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع  
Data

● شیء داده (Data Object)

● مکانی برای نگهداری Data Value

● انواع شیء داده‌ها

1. Programmer Defined

● مانند متغیرها، ثوابت، آرایه، ...

2. System Defined

● مانند:

پشته‌های زمان اجرا، رکوردهای فعالیت، زیربرنامه‌ها، لیست فضای آزاد، بافر فایل‌ها

● برنامه‌نویس از وجود آن‌ها بی‌خبر است و با آن‌ها کار نمی‌کند

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل  
نوع  
طول عمر شیء داده

---

● Data Object Life time

- فاصله‌ی زمانی از وقتی که شیء داده ساخته می‌شود تا وقتی که منهدم می‌شود
- بیشتر شیء داده‌ها، طول عمری برابر با طول اجرای برنامه دارند
- اما گاهی چنین نیست! مثال:
- Malloc در زبان C

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل  
نوع  
Binding های شیء داده

---

● انواع Binding ها در طی life time یک شیء داده

- اسم:
- در زمان کامپایل منتسب می‌شود
- ممکن است تغییر یابد
- مثال:
- با رفت و برگشت به عنوان پارامتر در زیر برنامه
- مقدار:
- در زمان اجرا

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## Binding های شیء داده

• انواع Binding ها در طی life time یک شیء داده

• Type:

- در زمان کامپایل منتسب می شود
- تا پایان اجرا عوض نمی شود

• حافظه ی حقیقی

- به وسیله ی روتین های مدیریت حافظه
- در زمان اجرا

A:

(a) **Data object:** A location in computer memory with the name A.

10001 = 17

(b) **Data value:** A bit pattern used by the translator whenever the number 17 is used in a program

A:

(c) **Bound variable:** Data object bound to data value 17

طراحی و پیاده سازی زبان های برنامه سازی

7 از 26

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## سایر اشیای داده

• تعریف متغیر (Variable)

• شیء داده ایی که برنامه نویس تعریف می کند و به آن مقدار می دهد

• تعریف ثابت (Constant)

• شیء داده ایی که مقدار آن در حین برنامه عوض نمی شود

• Literal

• شیء داده ایی که نامش نماینده ی مقدارش است

• مثال:

const x=13 یا x='Tehran'

طراحی و پیاده سازی زبان های برنامه سازی

8 از 26

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع  
سایر اشیای داده

مثال:

```
const MAX=34;
var n:integer;
...
n=345;
n=n+max;
```

ثابت

متغیر

Literal

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع  
نوع داده (Data Type)

تعریف

- یک کلاس از شیء داده‌ها، همراه با عملیات لازم برای ساختن و دستکاری (manipulation) آن‌ها
- اقسام نوع داده:
  1. ابتدایی (Elementary)
    - مانند boolean, real, int
  2. ساخت‌یافته (Structured)
    - مانند Union, آرایه، رکورد،

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## مطالعه‌ی نوع داده‌ها (Data Types)

### 1. Specifications

1. Attribute: خصیصه‌های نوع داده
2. Value: مقادیری که می‌توان به نوع داده داد
3. Operation: عملیاتی که بر روی نوع داده تعریف می‌شود

### 2. Implementation

1. Storage Representation: طرز نمایش در حافظه
2. Algorithms: نحوه‌ی پیاده‌سازی Operationها، با نرم‌افزار یا سخت‌افزار؟ و با چه الگوریتمی

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## مطالعه‌ی نوع داده‌ها (Data Types)

### • نکته:

- Specificationها بر Implementation اثر می‌گذارند
- Attributeها روی نمایش در حافظه اثر می‌گذارند
- مثلا:
- int یا Long int نحو نمایش متفاوتی دارند، در اینجا Attribute در مکانی ذخیره نشده
- اما در آرایه، Attها روی نمایش در حافظه مؤثر هستند و به صورت Descriptor در کنار فضای آرایه ذخیره می‌شوند
- طراحی باید دقیقا Operationها را تعریف نماید
- تعداد، ترتیب، نوع آرگومان‌ها، نام، عملکرد دقیق Operationها

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## مثال از نوع داده‌ها (Data Types)

• مثال:

+ : integer\*integer -> integer

Sqrt: real -> real

==: real \*real -> Boolean

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## پیچیدگی‌های تعریف عملیات بر نوع داده‌ها (Data Types)

• عوامل مؤثر بر پیچیده شدن عملیات بر نوع داده:

1. عملیات به ازای برخی ورودی‌ها تعریف نشده است

• مثال:

تقسیم بر صفر، تعیین محدوده‌ی مقادیر آرگومان‌ها که نتیجه، سرریز خواهد شد

2. آرگومان‌های ضمنی (Implicit):

• مقادیری که جزو آرگومان‌های صریح نیستند و Operation از طریق قواعد

Scope به آن‌ها دسترسی دارد

• مثال:

متغیرهای سراسری

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## پیچیدگی‌های تعریف عملیات بر نوع داده‌ها (Data Types)

• عوامل مؤثر بر پیچیده شدن عملیات بر نوع داده:

3. نتایج ضمنی (Implicit Results) یا اثر جانبی (Side Effect):

- عملیات ممکن است علاوه بر وظیفه‌ی خود کار دیگری نیز انجام دهد
- به صورت غیرصریح روی متغیرهای سایر برنامه‌ها تأثیر گذارد
- مثال:

تابعی که علاوه بر مقدار بازگشتی، آرگومان‌ها ورودی خود را نیز تغییر دهد

4. حساسیت به سابقه‌ی اجرا

- عملیاتی مانند تولید اعداد تصادفی
- نتیجه‌ی آن در هر بار اجرا متفاوت است

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## پیاده‌سازی عملگرها

1. مستقیماً از طریق سخت‌افزار

2. به عنوان یک تابع یا زیربرنامه، از طریق شبیه‌سازی نرم‌افزار

• مثال:

تابع جذر

3. In-Line Code

• کد زیر برنامه فراخوانی شده، در نقطه‌ی فراخوانی، داخل برنامه‌ی فراخوان کپی می‌شود.

$ABS(x) = \text{if } x < 0 \text{ then } -x \text{ else } x$

a) Fetch value of x from memory

b) If  $x > 0$  skip next instruction

c) Set  $x = -x$

d) Store x in memory

• مستقیماً از طریق سخت‌افزار

پیاده‌سازی می‌شود



عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## اعلان (Declaration) در زبان‌های برنامه‌نویسی

### • تعریف

• دستوری که در خدمت کامپایلر قرار می‌گیرد تا اطلاعاتی در مورد شیء داده، به شرح زیر بیان کند:

- نام شیء داده
- نوع آن
- مقدار اولیه
- طریقه‌ی ذخیره در حافظه
- طول عمر آن

### • انواع اعلان

1. صریح

2. ضمنی

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## انواع اعلان

### • اعلان صریح

• تعریف شده توسط برنامه‌نویس

• مثال:

`int I;`

### • اعلان ضمنی

• تعریف شده توسط زبان برنامه‌سازی

• مثال:

• در `Qbasic`

• در فرترن متغیرهای `n, ..., j, i` از نوع صحیح در نظر گرفته می‌شود

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## محاسن استفاده از اعلان

### 1. Storage Representation

- کامپایلر دقیقاً می‌داند برای شیء داده، چقدر فضا در نظر بگیرد
- نحوه‌ی نمایش شیء داده چگونه باشد
  - تسریع کار کامپایلر

### 2. Storage Management

- روتین‌های مدیریت حافظه، بر میزان حافظه‌ی مورد نیاز آگاهی دارند
- نتیجه:
  - هر بار در حین اجرا، ناچار به کم و زیاد نمودن حافظه‌ی برنامه نیست

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## محاسن استفاده از اعلان

### 3. Generic Operation (Polymorphism یا Overloading)

- بعضی Operation‌ها با اسم یکسان، مفاهیم مختلف دارند
  - مفهوم آن‌ها وابسته به نوع آرگومان‌هایشان است
- مثال:
  - عمل جمع (+)
  - جمع دو integer یا جمع دو real
  - الحاق دو رشته
  - OR کردن دو مقدار بولی
- مزیت اعلان در این زمینه:
  - اگر نوع، در زمان کامپایل مشخص شود، مفهوم Operation هم معلوم می‌شود
  - کامپایلر کد کاراتری تولید می‌کند

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## محاسن استفاده از اعلان

### 4. کنترل نوع ایستا (Static Type checking)

- مقایسه‌ی نوع و تعداد آرگومان‌های عملیات، در داخل برنامه، با آنچه که در زبان تعریف شده است

### • انواع کنترل نوع

1. ایستا: بررسی در زمان کامپایل صورت می‌گیرد
2. پویا: بررسی در زمان اجرا صورت می‌گیرد

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## مقایسه انواع کنترل نوع

- |  |                                    |
|--|------------------------------------|
| • مزایای کنترل نوع ایستا   | • مزایای کنترل نوع پویا            |
| • سرعت اجرای برنامه بالا می‌رود                                  | • انعطاف زیاد در برنامه‌سازی       |
| • عدم نیاز به نگهداری اطلاعات در مورد نوع                        | • تبدیل نوع در زمان اجرا           |
| • تمام مسیرهای اجرایی از جهت نوع چک می‌شوند، اشکال زدایی ساده‌تر | • عدم نیاز به اعلان برای هر متغییر |

- افزایش سرعت اجرا

- افزایش انعطاف

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## مقایسه انواع کنترل نوع

- معایب کنترل نوع ایستا
  - انعطاف پذیری کم
  - مشکلاتی در کامپایل مجزای برنامه
  - نیاز به اعلان برای تمام اشیای داده
  - کنترل عملیات چندریختی
  - ترجمه‌ی مستقل زیربرنامه‌ها
- معایب کنترل نوع پویا
  - زمانبر
  - حافظه اضافی نیاز دارد (توصیف‌گر)
  - اشکال زدایی برنامه‌ها مشکل می‌شود
  - سخت‌افزار به‌ندرت از کنترل نوع پویا حمایت می‌کند

عناوین ارائه: شیء داده - نوع داده - عملیات - اعلان - کنترل نوع - تبدیل نوع

## تبدیل نوع

- در زمان Type Checking دو حالت روی خواهد داد:
  1. تطبیق
    - مشکلی وجود نخواهد داشت
  2. عدم تطبیق
    1. پیام خطا (Type Mismatch)
    2. تبدیل نوع (Type Conversion)
      1. صریح
        - برنامه‌نویس، خود با فراخوانی روتین‌هایی، نوع را تبدیل می‌کند
        - مثال: `inttooreal(x);` یا `(int)X` برای یک متغیر `Float`
      2. ضمنی
        - تبدیل نوع، به وسیله‌ی زبان انجام می‌شود

## تبدیل نوع

- در زمان کامپایل
- مقایسه‌ی نوع‌ها در داخل جدول نمادها (Symbol Table) یا جدولی مرتبط با Typeها
- و تشخیص عدم تطبیق و انجام عمل تطبیق
- در زمان اجرا
- کامپایلر با تولید کدی مفصل، همه‌ی موارد عدم تطبیق را پیش‌بینی می‌نماید
- برحسب هر مورد، تبدیل مناسب را کد می‌کند
- در زمان اجرا با توجه به محتوی توصیف‌گر، تبدیل انجام می‌گیرد

پرسش؟