

به نام خدا



درس طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

موضوع: **Encapsulation, Information Hiding, Abstraction**

ارائه: صادق سلیمانی

www.Bioinformation.ir

جلسه‌ی هشتم

عناوین جلسه

- Abstraction
- Abstract Data Type
- Subprogram
- Type Definition

عناوین ارائه: تجريد - Abstract Data Type - زیربرنامه - تعريف نوع

تجريد (Abstraction)

- تعريف خلاصه‌سازی (تجريد)
 - به مقدار زيادی از اطلاعات تنها با بردن يك نام رجوع كنيم
 - مثال:
 - if ... then ... else ...
 - فلوچارت يك خلاصه‌سازی از برنامه است
 - در سلسله‌مراتب كامپيوترها، هر كامپيوتر يك خلاصه‌سازی برای سطح قبلی است
 - الگوریتم يك خلاصه‌سازی از فلوچارت است

طراحی و پياده‌سازی زبان‌های برنامه‌سازی

3 از 26

عناوین ارائه: تجريد - Abstract Data Type - زیربرنامه - تعريف نوع

انواع تجريد (Abstraction)

- انواع تجريد که زبان برای برنامه‌نویس فراهم می‌کند
 1. امکان تعريف Data Type جديد
 - برنامه‌نویس باید Attributeها، Operationها و نمایش در حافظه را برای آن نوع داده، مشخص نماید
 - نیز باید بتواند آن را Encapsulate کند
 2. نوشتن زیربرنامه
 - مثلاً نوشتن زیربرنامه‌ای برای محاسبه‌ی فاکتوریل يك عدد

طراحی و پياده‌سازی زبان‌های برنامه‌سازی

4 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

تعریف و نحوه ایجاد

- نوع داده‌ی انتزاعی
 - یک نوع داده به همراه مجموعه‌ایی از عملیات بر روی آن، که از سوی برنامه‌نویس و توسط امکاناتی که زبان در اختیار آن قرار می‌دهد، تعریف و ایجاد می‌گردد
 - همچنین برنامه‌نویس، یک سری عملیات نیز بر آن تعریف می‌کند
- Abstract Data Type Creation
 1. Data Object
 2. Abstract Operations
 3. Encapsulation

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

5 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

مثال از A.D.T.

- تعریف یک نوع داده‌ی جدید در سیستم ثبت‌نام دانشجویان دانشگاه برای نگهداری مشخصات دروس ارائه شده
- یک طرح نمونه

مکان تشکیل کلاس	} رکورد
زمان تشکیل کلاس	
نام مدرس	
ظرفیت کلاس	} آرایه
شماره دانشجویی	
...	

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

6 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

مثال از A.D.T.

• ادامه مثال

• Attribute ها

• چهار فیلد اول

• Value

• شماره دانشجویی

• Operation ها

1. Def (Physic, Att)

• یک ساختمان داده مطابق طراحی پیشین ایجاد می‌نماید و فضا به آن اختصاص می‌دهد

• اشاره گر به آن را در متغیر Physic قرار می‌دهد

• با استفاده از اطلاعات فیلد Att، چهار جزء اول را مقداردهی اولیه می‌کند

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

7 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

مثال از A.D.T.

• ادامه مثال

• Operation ها

2. Add (Physic, No)

• افزودن دانشجو به لیست درس

3. Delete (Physic, No)

• حذف دانشجو از لیست درس

4. Purge (Physic)

• در صورت عدم ثبت نام هیچ دانشجویی، فضا آزاد شود

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

8 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

مثال از A.D.T.

• ادامه مثال

• Encapsulation

- تغییرات در آن Type، جز به وسیله‌ی Operation های تعریف شده، امکان نداشته باشد
- معمولاً از امکانات خود زبان است

• تعریف Encapsulation

- امکانی برای برنامه‌نویس در جهت گروه‌بندی داده و Subroutine ها در یک مکان، برای انجام عملیات روی آن‌ها
- و مخفی کردن جزئیات اضافی، از دید استفاده کننده‌ی آن

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

9 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

Information Hiding و Encapsulation

• در مقایسه با Encapsulation

• Information Hiding

- در هر زبانی امکان‌پذیر است و هر برنامه‌نویس با کاربرد روتین‌هایی مثل از بین بردن Source برنامه یا رمزدار کردن آن می‌تواند اطلاعات را مخفی کند
- این موارد از مخفی‌کاری، با تلاش قابل خنثی شدن هستند
- اما در Encapsulation دسترسی به اجزا هرگز امکان ندارد
- مثال

تابع Sqrt()

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

10 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

Encapsulation

مثال

<pre> struct DateStruct { int nMonth; int nDay; int nYear; } int main() { DateStruct sDate; sDate.nMonth = 10; sDate.nDay = 14; sDate.nYear = 2020; return 0; } </pre>	<pre> class Date { int mMonth; int mDay; int mYear; }; int main() { Date cDate; cDate.mMonth = 10; cDate.mDay = 14; cDate.mYear = 2020; return 0; } </pre>
--	--

```

class Date
{
public:
    int mMonth; // public
    int mDay; // public
    int mYear; // public
};

int main()
{
    Date cDate;
    cDate.mMonth = 10;
    cDate.mDay = 14;
    cDate.mYear = 2020;

    return 0;
}

```

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

11 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

زیربرنامه‌ها

• مکانیزم دیگر محصورسازی یا encapsulation

• تعریف

• تابعی که هر مجموعه از آرگومان‌های ورودی را به یک مجموعه از آرگومان‌های خروجی نگاشت می‌کند

• مثال

Function fact1(x,y:integer;T:real): boolean;

Var m,n;

begin

...

end;

Fact1:integer×integer×real -> boolean

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

12 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

طراحی زیربرنامه

1. اسم برای زیربرنامه
2. امضای زیربرنامه (Subprogram Assignment)
 1. تعداد آرگومان‌ها
 2. نوع آرگومان‌ها
 3. ترتیب آرگومان‌ها
 4. تعداد نتایج
 5. نوع نتایج
 6. ترتیب نتایج
3. عملکرد زیربرنامه

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

13 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

مشکلات تعریف زیربرنامه

- تفاوت‌های زیر برنامه با توابع ریاضی
 1. شرایط خطا
 - زیربرنامه برای بعضی مقادیر، پارامتر محاسباتی نداشته باشد و تعریفی هم نداشته باشد
 2. آرگومان‌های ضمنی
 - به شکل غیر محلی؛ مثلاً متغیرهای سراسری
 3. نتایج ضمنی (اثر جانبی)
 - مثلاً آرگومان‌های ورودی، خروجی را تغییر دهد
 4. وابستگی به تاریخچه
 - نتایج آن به آرگومان‌هایی وابسته باشد که در فراخوانی‌های قبلی به آن ارسال شده‌اند

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

14 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

مشکلات تعریف زیربرنامه

- تفاوت‌های زیر برنامه با توابع ریاضی
1. مثال از آرگومان‌های ضمنی

```
int y;

int Add(int x)
{
    return x+y;
}

main()
{
    y=6;
    Cout<<Add(4);
    y=10;
    Cout<<Add(14);
}
```

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

15 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

مشکلات تعریف زیربرنامه

- تفاوت‌های زیر برنامه با توابع ریاضی
1. مثال از نتایج ضمنی (اثر جانبی)

```
int Func(int &x)
{
    x+=5;
    return x*x;
}

main()
{
    int a=3;
    cout<<a*Func(a)+a;
}
```

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

16 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

مشکلات تعریف زیربرنامه

- تفاوت‌های زیر برنامه با توابع ریاضی
1. مثال از وابستگی به تاریخچه

```
int Func(int x)
{
    static int y=0;
    y++;
    return x+y;
}

main()
{
    cout<<Func(2);
    cout<<Func(3);
    cout<<Func(2);
}
```

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

17 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

پیاده‌سازی و اجرای زیربرنامه

- دو بخش باید مورد نظر باشد:
- Subprogram Definition
 - محتویات زیربرنامه، که در زمان کامپایل به کار می‌آید
 - متغیرها و دستورات
- Subprogram Activation
 - فعالیت‌های زیربرنامه در حین اجرای (فراخوانی) آن

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

18 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

پیاده‌سازی و اجرای زیربرنامه

- فضایی که به زیربرنامه هنگام فراخوانی اختصاص داده می‌شود

1. Static Part or Code Segment

- فضا برای کد زیربرنامه و مقادیر ثابت

2. Dynamic Part or Activation Record

- فضا برای آدرس برگشت، نتایج بازگشتی، متغیرهای محلی و آرگومان‌های ورودی

- مزیت

- به ازای چندین بار فراخوانی زیربرنامه، تنها یک بار، به بخش ایستای آن حافظه اختصاص داده می‌شود
- به ازای هر بار فراخوانی برنامه، یک بخش پویا ایجاد می‌گردد

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

19 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

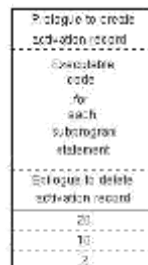
پیاده‌سازی و اجرای زیربرنامه

- مثال

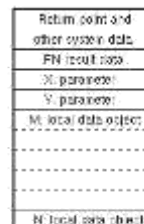
```
float FN( float X, int Y)
{const initial=2;
#define finalval 10
float M(10); int N;
N = initial;
if(N<finalval){ ... }
return (20 * X + M(N)); }
```

- Prologue و Epilogue کدهایی هستند که

کامپایلر ضمیمه‌ی برنامه می‌کند



Code segment for subprogram FN



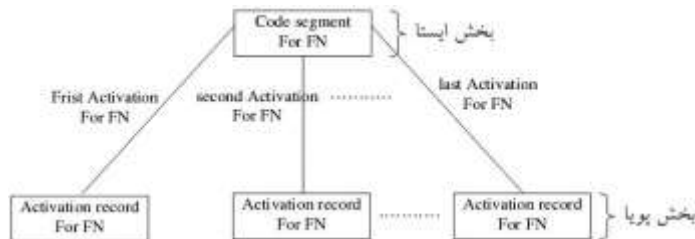
Activation record for FN (Template)

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

20 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

پیاده‌سازی و اجرای زیربرنامه



Prologue

- کد برای تنظیم رکورد فعالیت، ایجاد ارتباطات غیرمحلی و انتقال پارامترها به داخل برنامه

Epilogue

- کدهای برگرداندن نتایج به زیربرنامه‌ی اصلی و آزادسازی فضای پویا

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

21 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

زیربرنامه‌های Generic

- چندین زیربرنامه با اسامی یکسان ولی تعداد پارامترها، نوع و ترتیب مختلف برای پشتیبانی از چنین امکانی فقط کافی است

مثال

- مورد اول: ثبت نام دانشجو در یک Section
- مورد دوم: افزودن یک Section به لیست کلاس‌ها

```
procedure ENTER(student: integer; SECT: var Section)
begin
end;

procedure ENTER(st: section; TAB: var Classlist)
begin
end;
```

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

22 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

تعریف نوع

- مکانیزم‌هایی برای توصیف دسته‌ایی از اشیاء
 - عمل جایگزینی را کامپایلر انجام خواهد داد
 - مثال
- ```
typedef int key_type;
key_type key1, key2;

struct rational_number
{int numerator, denominator;};

typedef rational_number rational; A.D.T. تعریف نوع با تعریف
```
- تفاوت تعریف نوع با تعریف A.D.T.
  - در تعریف نوع، عملیات نیز تعریف نمی‌شود
- ```
rational r1, r2;
```

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

23 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

معادل بودن دو نوع

- مشکلاتی که بر اثر تعریف نوع ممکن است بروز کند:
- ```
Type VECT1: Array[1...10] of Real;
VECT2: Array[1...10] of Real;
Var X, Y: VECT1;
Y: VECT2;
procedure SUB (A: VECT1);
Begin
:
End
Begin
X := Y;
SUB(Y);
END.
```
- معادل بودن اسمی
  - دو نوع با هم یکسان هستند اگر نام آن‌ها یکسان باشد
  - نتیجه:
  - هر شیء دارای یک نوع داده‌ی نسبت داده شده است
  - معادل بودن ساختاری
  - دو نوع با هم مساوی هستند اگر ساختار آن‌ها یکسان باشد
  - نتیجه:
  - انعطاف‌پذیری بالای زبان برنامه‌سازی
  - هزینه‌ی بالاتر برای کامپایلر

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

24 از 26

عناوین ارائه: تجرید - Abstract Data Type - زیربرنامه - تعریف نوع

## نوع داده‌ی پارامتریک

- تعریف نوع داده، به برخی پارامتر وابسته خواهد شد
  - پارامترها به کاربران اجازه می‌دهند که مثلاً اندازه‌ی نوع داده‌ی مورد نیاز را مشخص کنند.
  - مثال:
 

```
type Section(Maxsize : integer) is
 record
 Room : integer;
 Instructor : integer;
 ClassSize : integer range 0..MaxSize;
 ClassRoll : Array (1..MaxSize) of Stud_ID;
 end record;
```
  - اندازه‌ی آرایه‌ها
  - پیاده‌سازی
  - در زمان کامپایل
  - تکمیل تعریف نوع در زمان استفاده
- X : Section(100);  
Y : Section(50);

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

26 از 25

پرسش؟

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

26 از 26