



Bài 3:

Lớp và Đối tượng trong Java

Trường ĐH Công nghệ, ĐHQG Hà Nội

Nội dung

- Khai báo Lớp
- Thao tác với các Đối tượng
- Thuộc tính, Phương thức, Kiểm soát truy cập
- Các phương thức khởi tạo

Tài liệu tham khảo



- *Giáo trình Lập trình HĐT, Chương 3, 4*
- *Java How to Program, Chapter 3, 8*

Chương trình Java



- Chương trình Java bao gồm một tập các đối tượng
- Mỗi lớp được đặc tả trong một tệp nguồn (tên tệp trùng với tên lớp)
 - Mỗi dòng lệnh trong Java đều thuộc một lớp (không tính các lệnh import)
 - Tăng tính mô-đun hóa
 - Dễ hơn cho sửa đổi mã nguồn, giảm thời gian biên dịch

khai báo Lớp
bắt đầu/kết thúc

02 lớp
trong 02 tệp

```
// GradeBook.java
public class GradeBook
{
    // display a welcome message to the GradeBook user
    public void displayMessage()
    {
        System.out.println( "Welcome to the Grade Book!" );
    }
} // end class GradeBook
```

lớp được khai báo **public** phải được
lưu trong tệp cùng tên lớp

```
// GradeBookTest.java
public class GradeBookTest
{
    // main method begins program execution
    public static void main( String args[] )
    {
        // create a GradeBook object and assign it to myGradeBook
        GradeBook myGradeBook = new GradeBook();
        // call myGradeBook's displayMessage method
        myGradeBook.displayMessage();
    }
} // end class GradeBookTest
```

main() được gọi tự động bởi máy ảo
Java khi chương trình được thực thi

tạo một thể hiện (đối tượng) của lớp và
được tham chiếu bởi myGradeBook

Welcome to the Grade Book!

Các đối tượng



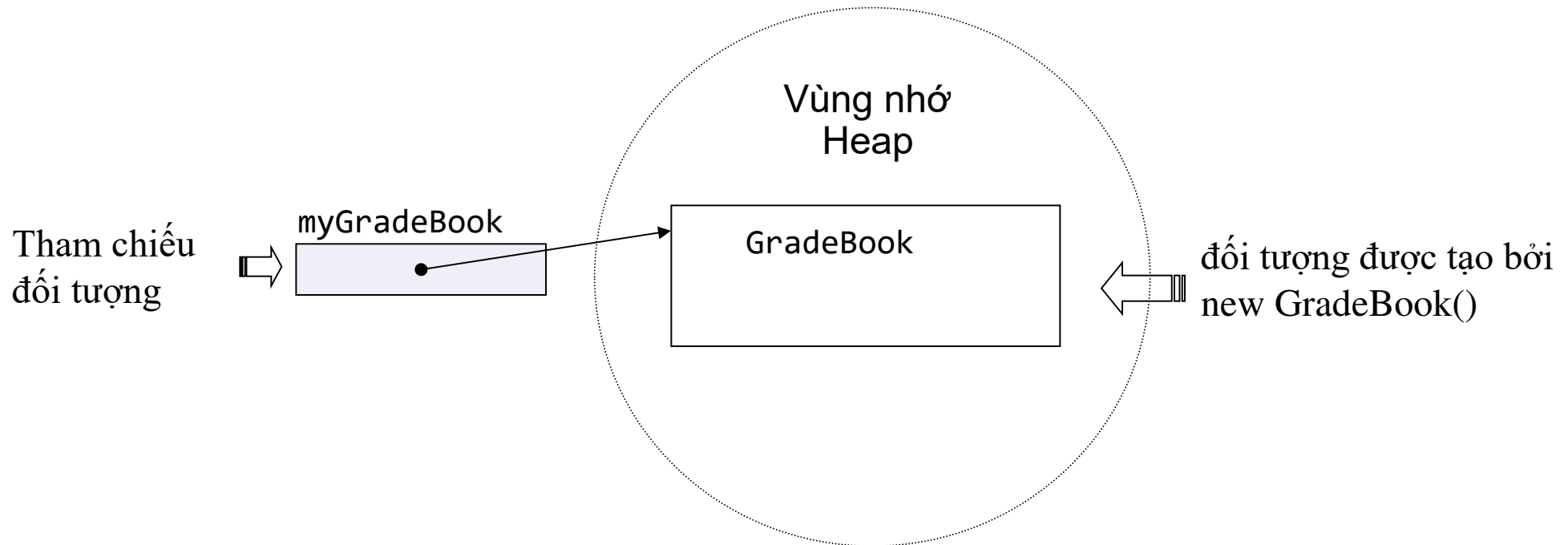
- Các đối tượng được thao tác qua các tham chiếu (references)
 - các tham chiếu đối tượng đóng vai trò giống như các con trỏ
- Các đối tượng phải được tạo tường minh bởi toán tử new

```
public class GradeBookTest {  
    // phương thức main bắt đầu thực thi chương trình  
    public static void main( String args[] ) {  
        // tạo đối tượng GradeBook và gán cho myGradeBook  
        GradeBook myGradeBook = new GradeBook();  
        // gọi phương thức displayMessage của myGradeBook  
        myGradeBook.displayMessage();  
    }  
} // kết thúc lớp GradeBookTest
```

Đối tượng và Tham chiếu đối tượng



```
...  
// tạo đối tượng GradeBook và gán cho myGradeBook  
GradeBook myGradeBook = new GradeBook();  
...
```



Thuộc tính, Phương thức và Kiểm soát truy cập



- Các phạm vi truy cập
 - Public
 - Có thể truy cập ở bất kỳ đâu và bởi bất kỳ ai
 - Protected
 - Chỉ có thể truy cập bởi lớp đó và các lớp con hoặc các lớp trong cùng gói (package)
 - Private
 - Chỉ có thể truy cập bởi lớp đó

GradeBook

- courseName : String
+ setCourseName(name : String)
+ getCourseName() : String
+ displayMessage()

private:
chỉ được truy cập
bởi các phương
thức của lớp

```
// GradeBook.java  
public class GradeBook  
{  
    private String courseName; // course name for this GradeBook
```

```
    // method to set the course name  
    public void setCourseName( String name )  
    {  
        courseName = name; // store the course name  
    }
```

```
    // method to retrieve the course name  
    public String getCourseName()  
    {  
        return courseName;  
    }
```

```
    // display a welcome message to the GradeBook user  
    public void displayMessage()  
    {  
        System.out.println( "Welcome to the Grade Book!" );  
    }  
} // end class GradeBook
```

thuộc tính (attribute):
Mỗi đối tượng **Gradebook** có biến
thể hiện tên **courseName**

các phạm vi
truy cập

các phương thức

```
class MyDate {  
    private int year, mon, day;  
    public int getYear() {  
        return year;  
    }  
    public void setYear(int y){  
        year=y;  
    }  
}
```

```
MyDate d = new MyDate();
```

```
...
```

```
d.year = 2005;    // compile error
```

```
d.setYear(2005);
```

```
System.out.println(d.getYear());
```

Nạp chồng phương thức (overloading)



Có thể định nghĩa các phương thức trùng tên nhưng phải khác danh sách tham số

```
class MyDate {  
    ...  
    public boolean setMonth(int m) { ... }  
    public boolean setMonth(String s) { ... }  
}  
...  
d.setMonth(9);  
d.setMonth("September");
```

Phương thức khởi tạo (constructor)



- Dữ liệu nên được khởi tạo trước khi sử dụng
 - lỗi khởi tạo là một trong các lỗi phổ biến
- Phương thức khởi tạo
 - là phương thức đặc biệt được gọi tự động *ngay sau khi đối tượng được tạo ra*
 - nhằm mục đích chính là khởi tạo cho các thuộc tính của đối tượng

Phương thức khởi tạo

- Tên trùng với tên lớp
- Không nhận giá trị trả lại
- Được gọi tự động mỗi khi đối tượng được tạo (bởi toán tử new)
 - nếu không khai báo, hệ thống sẽ gọi phương thức khởi tạo mặc định - phương thức rỗng

```
// GradeBook.java
public class GradeBook
{
    private String courseName; // course name

    // constructor initializes courseName
    public GradeBook( String name )
    {
        courseName = name; // initializes courseName 12
    }
    // method to set the course name
    public void setCourseName( String name )
    {
        courseName = name; // store the course name
    }
    // method to retrieve the course name
    public String getCourseName()
    {
        return courseName;
    }
    // display a message
    public void displayMessage()
    {
        System.out.println( "GradeBook for: " + getCourseName() );
    }
} // end class GradeBook
```

```
// GradeBookTest.java
...
// tạo đối tượng GradeBook và gán cho myGradeBook
GradeBook myGradeBook =
    new GradeBook( "CS101 Introduction to Java Programming" );
...
```

GradeBook

- courseName : String

«constructor» GradeBook(name: String)

+ setCourseName(name: String)

+ getCourseName() : String

+ displayMessage()

Ví dụ: Phương thức khởi tạo rỗng



Phương thức khởi tạo không được khai báo tường minh

```
class MyDate {  
}
```

...

```
MyDate date = new MyDate();
```

Ví dụ: Phương thức khởi tạo mặc định



Phương thức khởi tạo không nhận tham số

```
class MyDate {  
    private int day, month, year;  
    public MyDate() {  
        day=1;  
        month=1;  
        year=2014;  
        System.out.println("A new date created");  
    }  
}  
...  
MyDate d= new MyDate();
```


Ví dụ: Nhiều phương thức khởi tạo



```
class MyDate {  
    private int day, month, year;  
    public MyDate() {  
        day=1;month=1;year=2014;  
        System.out.println("A new date created");  
    }  
    public MyDate(int d,m,y) {  
        day=d;  
        month=m;  
        year=y;  
    }  
}  
  
...  
MyDate d1 = new MyDate();  
MyDate d2 = new MyDate(19, 9, 2014);
```

Ví dụ: Khi không có phương thức khởi tạo mặc định



```
class MyDate {  
    private int day, month, year;  
    public MyDate(int d,m,y) {  
        day=d;  
        month=m;  
        year=y;  
    }  
}  
  
...  
MyDate d1 = new MyDate(); //error  
MyDate d2 = new MyDate(19,9,2014);
```

Khởi tạo sao chép (copy constructor)



Khởi tạo đối tượng bằng một đối tượng khác

```
public class MyDate {  
    private int year, month, day;  
    public MyDate() {...}  
    public MyDate(MyDate d) {  
        year = d.year;  
        month = d.month;  
        day = d.day;  
    }  
    ...  
}
```

Ví dụ khởi tạo sao chép

```
MyDate d = new MyDate();  
d.setYear(2005);  
d.setMonth(9);  
d.setDay(12);  
MyDate openDay = new MyDate(d);  
MyDate dd = d;  
dd = new MyDate();
```

Cài đặt và Giao diện

- GradeBookTest: là “client” của GradeBook
- Cài đặt (Implementation)
 - mã nguồn và cấu trúc dữ liệu cài đặt lớp (biến và phương thức)
 - tập trung vào hoạt động bên trong
 - “clients” không cần biết về chi tiết cài đặt
- Giao diện (Interface)
 - Phần điều khiển lộ ra bên ngoài, cho phép “client” gửi yêu cầu thực hiện các dịch vụ (phương thức) được cài đặt
 - Giống như các nút bấm chức năng trên các hộp đen

```
// GradeBookTest.java
// tạo đối tượng GradeBook và gán cho myGradeBook
GradeBook myGradeBook =
    new GradeBook( "CS101 Introduction to Java Programming" );
```

Bao gói / Che dấu thông tin

- “Không lộ các cấu trúc dữ liệu bên trong!”
- Các đối tượng lưu giữ dữ liệu và mã lệnh
 - Không lộ ra cho cho người dùng cuối
- Các đối tượng mở ra / cung cấp các giao diện
 - dịch vụ được cung cấp bởi các đối tượng
- Độ phức tạp được ẩn bên trong đối tượng
 - Tạo thuận tiện cho các “client” (chế ngự độ phức tạp)
 - Tính mô-đun hóa cao (cục bộ hóa sự thay đổi)
 - Cài đặt có thể được thay đổi mà không ảnh hưởng bên ngoài (thành phần khác)
 - Ít phát sinh lỗi

Tổng kết



- Khai báo Lớp
- Thao tác với các Đối tượng
- Thuộc tính, Phương thức, Kiểm soát truy cập
- Các phương thức khởi tạo