



LẬP TRÌNH JAVA

CHƯƠNG 4: GIAO DIỆN VÀ GÓI

- ✓ Định nghĩa, cài đặt giao diện
- ✓ Định nghĩa gói
- ✓ Tạo và sử dụng các gói
- ✓ Những thành phần của gói
java.lang
- ✓ Những thành phần của gói
java.util



Giới thiệu

- ✓ Những thành phần cơ bản của một chương trình Java:
 - Gói (Packages)
 - Lớp (Class)
 - Giao diện (Interfaces)
- ✓ Một chương trình Java bao gồm:
 - Lệnh khai báo gói(**package**)
 - Lệnh chỉ định gói được dùng (Lệnh **import**)
 - Khai báo lớp public (một file java chỉ chứa 1 lớp public class)
 - Các lớp khác (classes private to the package)



Interfaces

- ✓ Chương trình Java chỉ có thể kế thừa một lớp duy nhất, nhưng có thể dẫn xuất nhiều Interface
- ✓ Tính chất của interface:
 - Tất cả phương thức trong interface phải là **public**.
 - Không được phép có những phương thức cụ thể (concrete methods)
 - Các phương thức phải được định nghĩa trong lớp dẫn xuất giao diện đó.



Gói (Packages)

- ✓ Tương tự như thư mục lưu trữ những lớp, interface và các gói con khác.
- ✓ Khai báo:
 - Package TênGói;
- ✓ Sử dụng gói:
 - Import TênGói.TênLớp;



Gói (Packages)

✓ Những ưu điểm khi dùng gói (Package):

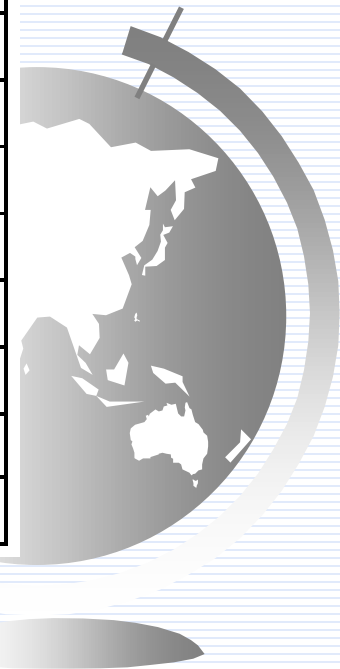
- Cho phép tổ chức các lớp vào những đơn vị nhỏ hơn
- Giúp tránh được tình trạng trùng lặp khi đặt tên.
- Cho phép bảo vệ các lớp đối tượng
- Tên gói (Package) có thể được dùng để nhận dạng chức năng của các lớp.



Gói java.lang

- ✓ Mặc định thì bất cứ chương trình Java nào cũng import gói **java.lang**
- ✓ Những lớp cho các kiểu dữ liệu nguyên thủy:

Data type	Wrapper class
boolean	Boolean
byte	Byte
char	Character
double	Double
float	Float
int	Integer
long	Long
short	Short



✓ Phương thức khởi tạo (Constructor):

- **String str1 = new String();**
- **String str2 = new String("Hello World");**
- **Char[] ch = {"A","B","C","D","E"};**
- **String str3 = new String(ch);**
- **String str4 = new String(ch,0,2);**



Những phương thức của lớp String

- ✓ **charAt()**
- ✓ **startsWith()**
- ✓ **endsWith()**
- ✓ **copyValueOf()**
- ✓ **toCharArray()**
- ✓ **indexOf()**
- ✓ **toUpperCase()**
- ✓ **toLowerCase()**
- ✓ **trim()**
- ✓ **equals()**



Lớp StringBuffer

- ✓ Cung cấp những phương thức khác nhau để thao tác trên đối tượng string
- ✓ Những phương thức của lớp

StringBuffer:

- **append()**
- **insert()**
- **charAt()**
- **setCharAt()**
- **setLength()**
- **getChars()**
- **reverse()**



Lớp java.lang.Math

✓ **abs()**

✓ **ceil()**

✓ **floor()**

✓ **max()**

✓ **min()**

✓ **round()**

✓ **random()**

✓ **sqrt()**

✓ **sin()**

✓ **cos()**

✓ **tan()**



Gói java.util

- ✓ Cung cấp phần lớn những lớp Java hữu dụng và thường xuyên cần đến trong hầu hết các ứng dụng
- ✓ Giới thiệu những lớp trừu tượng sau:
 - Hashtable
 - Random
 - Vector
 - StringTokenizer



Lớp Hashtable

- ✓ Dùng để nối kết những khóa vào những giá trị cụ thể
- ✓ Phương thức khởi tạo Hashtable:
 - **Hashtable(int)**
 - **Hashtable(int, float)**
 - **Hashtable()**



Lớp Hashtable

✓ Ví dụ:

```
1. Hashtable ht = new Hashtable();
2. ht.put("Pulse", new Integer(1995));
3. ht.put("Dark Side of the Moon", new Integer(1973));
4. ht.put("Wish You Were Here", new Integer(1975));
5. ht.put("Animals", new Integer(1997));
6. ht.put("Ummagumma", new Integer(1969));
7. System.out.println("Initailly: "+ht.toString());
8. //kiểm tra cho bất kỳ tập ảnh nào từ 1969
9. if(ht.contains(new Integer(1969)))
10.     System.out.println("An album from 1969 exists");
11. //kiểm tra cho tập ảnh các con thú
12. if(ht.containsKey("Animals"))
13.     System.out.println("Animals was found");
14. //Tìm ra
15. Integer year = (Integer)ht.get("Wish You Were Here");
16. System.out.println("Wish you Were Here was released in"+year.toString());
17. //Xoá một tập ảnh
18. System.out.println("Removing Ummagumma\r\n");
19. ht.remove("Ummagumma");
20. //Duyệt qua tất cả các khoá trong bảng.
21. System.out.println("Remaining:\r\n");
22. for(Enumeration enum = ht.keys(); enum.hasMoreElements();)
23.     System.out.println((String)enum.nextElement());
```



Lớp Random

- ✓ Tạo ra những số ngẫu nhiên theo thuật toán pseudo
- ✓ Phương thức khởi tạo:
 - **random()**
 - **random(long)**
- ✓ Những phương thức nhận giá trị ngẫu nhiên:
 - **nextDouble()**
 - **nextFloat()**
 - **nextGaussian()**
 - **nextInt()**
 - **nextLong()**



Lớp Vector

- ✓ Cung cấp khả năng co giãn cho mảng khi thêm phần tử vào mảng
- ✓ Lưu trữ những thành phần của kiểu Object
- ✓ Một Vector riêng rẽ có thể lưu trữ những phần tử khác nhau, đó là những instance của những lớp khác nhau
- ✓ Phương thức khởi tạo (Constructors):
 - **Vector(int)**
 - **Vector(int, int)**
 - **Vector()**



Lớp Vector

✓ Ví dụ:

```
1. Vector vect = new Vector();
2. vect.addElement("One");
3. vect.addElement("Two");
4. vect.addElement("Three");
5. vect.addElement("Four");
6. vect.addElement("Five");
7. vect.insertElementAt("Numbers In Words",0);
8. vect.insertElementAt("Four",4);
9. System.out.println("Size: "+vect.size());
10. System.out.println("Vector ");
11. for(int i = 0; i<vect.size(); i++){
12.     System.out.println(vect.elementAt(i)+" , ");
13. }
14. vect.removeElement("Five");
15. System.out.println("");
16. System.out.println("Size: "+vect.size());
17. System.out.println("Vector ");
18. for(int i = 0; i<vect.size(); i++) {
19.     System.out.print(vect.elementAt(i)+ " , ");
20. }
```



Lớp StringTokenizer

- ✓ Có thể được dùng để tách một chuỗi thành những thành phần cấu thành của nó.
- ✓ Ký tự phân cách có thể được chỉ định khi một đối tượng **StringTokenizer** được khởi tạo.
- ✓ Phương thức khởi tạo (Constructors):
 - **StringTokenizer(String)**
 - **StringTokenizer(String, String)**
 - **StringTokenizer(String, String, Boolean)**
- ✓ Lớp **StringTokenizer** sử dụng giao diện liệt kê



Lớp StringTokenizer

✓ Ví dụ:

1. `String mathExpr = "4*3+2/4";`
2. `StringTokenizer st1 = new
StringTokenizer(mathExpr, "*+/-", true);`
3. `System.out.println("Tokens of mathExpr: ");`
4. `while(st1.hasMoreTokens())`
5. `System.out.println(st1.nextToken());`
6. `String commas = "field1,field2,field3,and field4";`
7. `StringTokenizer st2 = new
StringTokenizer(commas, ",", false);`
8. `System.out.println("Comma-delimited tokens : ");`
9. `while (st2.hasMoreTokens())`
10. `System.out.println(st2.nextToken());`
11. `}`



- ✓ 1. Viết một chương trình hiển thị bình phương của các số lớn nhất và nhỏ nhất của một tập các số được nhập vào bởi người sử dụng hoặc tạo ngẫu nhiên (lớp Math).
- ✓ 2. Viết chương trình đoán số:
 - Tạo số ngẫu nhiên thuộc $[1, 100]$
 - Người dùng đoán tối đa n lần, nếu đoán sai thì xuất câu “lớn hơn” hay “nhỏ hơn” tương ứng.



Bài tập (tt)

- ✓ 3. Hãy tạo ra sổ ghi nhớ của chính bạn, nơi mà những con số được nhập vào như sau:

Joy	34543
Jack	56765
Tina	34567

Yêu cầu:

1. Kiểm tra xem số 3443 có tồn tại trong sổ ghi nhớ của bạn hay không.
2. Kiểm tra xem mẫu tin của Jack có hiện hữu trong sổ ghi nhớ của bạn hay không.
3. Hiển thị số điện thoại của Tina.
4. Xóa số điện thoại của Joy.
5. Hiển thị các mẫu tin còn lại.

