



Bài 5. Quản lý bộ nhớ và tổ chức lớp

Lý thuyết và ngôn ngữ hướng đối tượng
(bài tập)

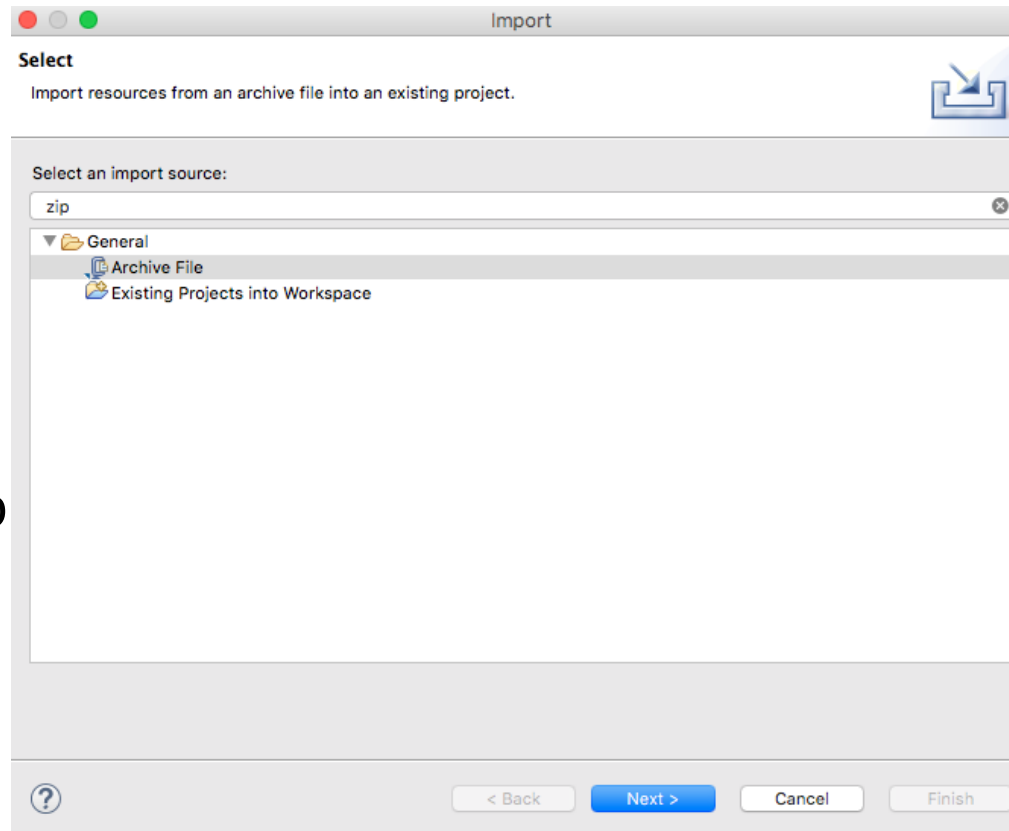


Lab's Objectives

- In this lab, you will practice with:
 - Creating packages to manage classes in Eclipse
 - Using some common packages/classes of Java API, e.g. Wrapper classes, Math, System
 - Practicing memory management with String and StringBuffer and other cases

1. Import/export a project

- You can import/export a project from/to an archive file
 - Open Eclipse
 - Open File -> Import
 - Type zip to find Archive File if you have exported as a zip file before. You may choose Existing Projects into Workspace if you want to open an existing project in your computer.





2. Re-organize your projects

- Rename project, use packages and re-organize classes
 - You can create sub-packages for more efficiently organizing your classes in both projects and all listing packages.

+ AimsProject

```
hust.soict.ictglobal.aims.disc.DigitalVideoDisc
hust.soict.ictglobal.aims.order.Order
hust.soict.ictglobal.aims.Aims
```

+ OtherProjects

```
hust.soict.ictglobal.lab01
hust.soict.ictglobal.lab02
hust.soict.ictglobal.date
```



3. Search a dvd in AimsProject project

- In the **DigitalVideoDisc** class, write a **boolean search(String title)** method which finds out (case insensitive) if the corresponding disk of the current object contains the **title**.
- Remember that if the **title** has multiple tokens (e.g. "Harry Potter"), the method still returns **true** if the disc has a **title** including all the tokens (e.g. both two tokens "Harry" and "Potter") regardless their order and their distance (so **true** for all title including a token "Harry" and a token "Potter" in any position in the **title**)



Xâu (String)

- Kiểu String là một lớp và không phải là kiểu dữ liệu nguyên thủy
- Một String được tạo thành từ một dãy các ký tự nằm trong dấu nháy kép:

```
String a = "A String";
```

```
String b = "";
```

- Đối tượng String có thể khởi tạo theo nhiều cách:

```
String c = new String();
```

```
String d = new String("Another String");
```

```
String e = String.valueOf(1.23);
```

```
String f = null;
```



Các phương thức của chuỗi

```
String name = "Joe Smith";  
name.toLowerCase();           // "joe smith"  
name.toUpperCase();           // "JOE SMITH"  
"Joe Smith ".trim();           // "Joe Smith"  
"Joe Smith".indexOf('e');       // 2  
"Joe Smith".length();           // 9  
"Joe Smith".charAt(5);           // 'm'  
"Joe Smith".substring(5);       // "mith"  
"Joe Smith".substring(2,5);     // "e S"
```



So sánh hai chuỗi

- `oneString.equals(anotherString)`

- Kiểm tra tính tương đương
- Trả về `true` hoặc `false`

```
String name = "Joe";  
if ("Joe".equals(name))  
    name += " Smith";
```

- `oneString.equalsIgnoreCase(anotherString)`

- Kiểm tra KHÔNG xét đến ký tự hoa, thường

```
boolean same = "Joe".equalsIgnoreCase("joe");
```

- So sánh `oneString == anotherString` sẽ gây nhập nhằng

- So sánh 2 đối tượng



3. Search a dvd in AimsProject project

- In the **Order** class, write a method **DigitalVideoDisc getALuckyItem()** which randomly pick out (remember to use **Math.random()**) an item for free.
- Remember to update and test the methods for listing of dvds and total cost of an order (specifying a lucky and free item).



4. String, StringBuilder and StringBuffer

- In the **OtherProjects** project, create a new package **hust.soict.ictglobal.garbage**
- Create a new class **ConcatenationInLoops** to test the processing time to construct **String** using **+** operator, **StringBuffer** and **StringBuilder**.



4. String, StringBuilder and StringBuffer

```
1  public class ConcatenationInLoops {
2      public static void main(String[] args) {
3          Random r = new Random(123);
4          long start = System.currentTimeMillis();
5          String s = "";
6          for (int i = 0; i < 65536; i++)
7              s += r.nextInt(2);
8          System.out.println(System.currentTimeMillis() - start); // This prints roughly 4500.
9
10         r = new Random(123);
11         start = System.currentTimeMillis();
12         StringBuilder sb = new StringBuilder();
13         for (int i = 0; i < 65536; i++)
14             sb.append(r.nextInt(2));
15         s = sb.toString();
16         System.out.println(System.currentTimeMillis() - start); // This prints 5.
17     }
18 }
```



StringBuffer

- **String**: nội dung của chuỗi không thay đổi – String object là immutable (đối tượng sẽ không thay đổi sau khi khởi tạo).
- **StringBuilder** (Chỉ từ Java 1.5 mới có lớp này): Khi nội dung của chuỗi sẽ thay đổi. Chỉ có 1 thread truy cập đến nội dung của chuỗi (Not threadsafe). Bù lại, tốc độ nhanh.
- **StringBuffer**: Khi nội dung của chuỗi sẽ thay đổi. Có thể có nhiều thread truy cập đến nội dung của chuỗi (threadsafe). Hạn chế tốc độ chậm.



StringBuffer

- String là kiểu bất biến:
 - Đối tượng không thay đổi giá trị sau khi được tạo ra → Các xâu của lớp String được thiết kế để không thay đổi giá trị.
 - Khi các xâu được ghép nối với nhau một đối tượng mới được tạo ra để lưu trữ kết quả → Ghép nối xâu thông thường rất tốn kém về bộ nhớ.
- StringBuffer là kiểu biến đổi:
 - Đối tượng có thể thay đổi giá trị sau khi được tạo ra



StringBuffer

- Nếu tạo xâu thông qua vòng lặp thì sử dụng **StringBuffer**

```
StringBuffer buffer = new StringBuffer(15);  
buffer.append("This is ") ;  
buffer.append("String") ;  
buffer.insert(7," a") ;  
buffer.append(' . ');  
System.out.println(buffer.length());           // 17  
System.out.println(buffer.capacity());          // 32  
String output = buffer.toString() ;  
System.out.println(output); // "This is a String."
```



4. String, StringBuilder and StringBuffer

- Create a new class **GarbageCreator** in the package **hust.soict.ictglobal.garbage**.
- Create “garbage” as much as possible and observe when you run a program (it should let the program stop working when too much “garbage”).
- Write another class **NoGarbage** to solve the problem.
- Some suggestions:
 - Read a text file to a **String** without using **StringBuffer** to concatenate String (only use **+** operator). Observe and capture your screen when you choose a very long file
 - Improve the code using **StringBuffer**