

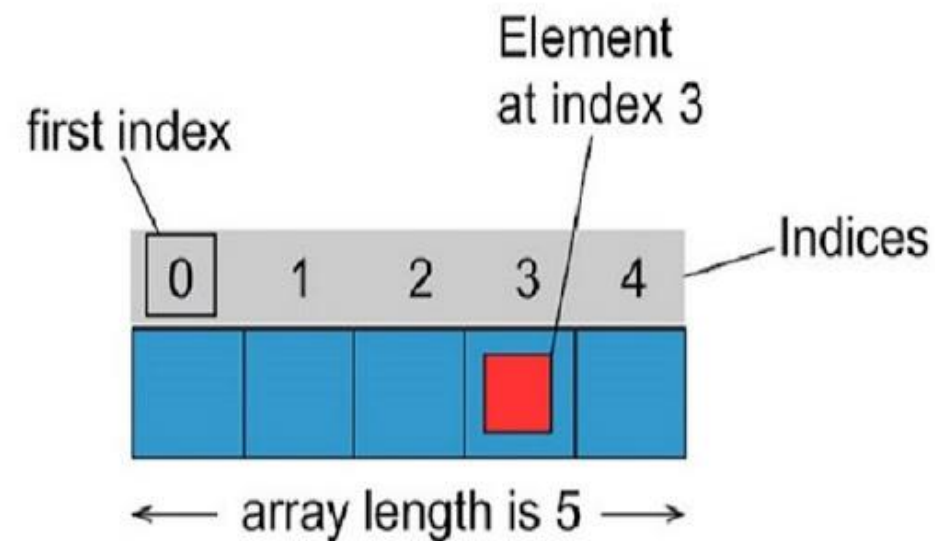


Bài 3

Mảng và chuỗi

- Giới thiệu về mảng
- Mảng đơn chiều và mảng đa chiều
- Lớp ArrayList và List interface
- Chuỗi - String
- StringBuilder và StringBuffer

- **Mảng – Array:** là tập hợp **các phần tử** có **cùng kiểu dữ liệu**, có **địa chỉ liên kề** nhau trên bộ nhớ. Mảng có **số phần tử cố định** và **không thể thay đổi kích thước** của nó
- **Các loại mảng:**
 - Mảng đơn chiều
 - Mảng đa chiều
- **Ưu điểm:**
 - Tối ưu hóa code: thu nhận và sắp xếp dữ liệu một cách dễ dàng
 - Truy cập ngẫu nhiên: có thể thao tác bất cứ dữ liệu của phần tử thông qua chỉ mục của nó
- **Nhược điểm:**
 - Không thể thay đổi kích thước: Chỉ có thể lưu trữ kích cỡ cố định số phần tử trong mảng



- Khai báo mảng

`DataType[] arrayName`

- Khởi tạo mảng

`arrayName = new DataType[size]`

- Khai báo và khởi tạo mảng

`DataType[] arrayName = new DataType[size]`

- Gán giá trị cho phần tử

`arrayName[index] = value`

- Khai báo, khởi tạo và gán giá trị cho các phần tử của mảng

`DataType[] arrayName = {value1,value2,....,valueN}`

• Ví dụ:

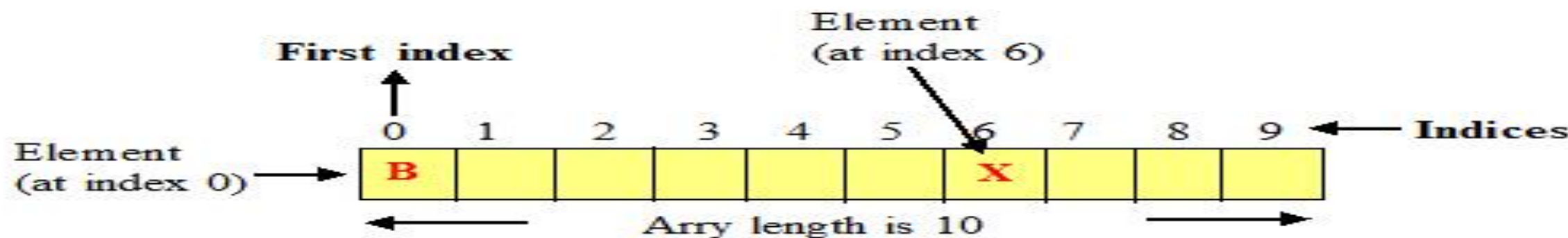
- `int[] arrAge`

- `arrAge = new int[10]`

- `int[] arrAge = new int[10]`

- `arrAge[2] = 18`

- `int[] arrAge = {18,20,19,22,25}`



- Khai báo mảng

`DataType[][] arrayName`

- Khởi tạo mảng

`arrayName = new DataType[row][col]`

- Khai báo và khởi tạo mảng

`DataType[][] arrayName = new DataType[row][col]`

- Gán giá trị cho phần tử mảng

`arrayName[rowIndex][colIndex] = value`

- Khai báo, khởi tạo và gán giá trị cho các phần tử của mảng

`DataType[][] arrayName = new DataType[][]
{ {value1,value2,...valueN},..., {value1,value2,...,valueN} }`

- Ví dụ:

- `int[][] arrAge`
- `arrAge = new int[3][2]`
- `int[][] arrAge = new int[3][2]`
- `arrAge[1][0] = 8`
- `int[][] arrAge = new int[][]
{ {5,7}, {4,9}, {8,3} }`

		Column 1	Column 2	Column 3	Column 4		
		Column indexes					
Row indexes		0	1	2	3		
Row 1	0	a[0][0] 15	a[0][1] 20	a[0][2] 25	a[0][3] 30	←	Array a[0]
Row 2	1	a[1][0] 20	a[1][1] 30	a[1][2] 40	a[1][3] 50	←	Array a[1]
Row 3	2	a[2][0] 60	a[2][1] 65	a[2][2] 70	a[2][3] 80	←	Array a[2]

- ArrayList là một lớp trong Collection có các đặc điểm sau:
 - Mềm dẻo, có khả năng tăng hoặc giảm kích thước khi cần
 - Cung cấp các phương thức hữu ích để tương tác với dữ liệu của mảng
 - Thêm/xóa dữ liệu đơn giản
 - Có thể truy cập các phần tử mảng thông qua chỉ số của nó
- Cấu trúc khai báo và khởi tạo
`List<DataType> arrayListName = new ArrayList<>()`
- Lưu ý: List và ArrayList thuộc package `java.util`

- Các phương thức:
 - **add(object o)**: Thêm một phần tử vào cuối danh sách
 - **add(int index, object o)**: Chèn một phần tử vào vị trí index
 - **get(int index)**: Lấy giá trị phần tử có chỉ số index
 - **set(int index, object o)**: Đè phần tử o vào vị trí index
 - **remove(int index)**: Xóa phần tử có chỉ số index
 - **int size()**: trả ra số phần tử của mảng
 - **clear()**: xóa tất cả các phần tử trong mảng

- String là đối tượng biểu diễn mảng các ký tự liên tục

- Khởi tạo đối tượng String

- Sử dụng **String Literal**

String strName = **"value"**

- Sử dụng từ khóa **new**

String strName = **new String**("value")

- Lưu ý: Nên sử dụng String Literal để khởi tạo chuỗi vì nó hiệu quả hơn trong việc sử dụng bộ nhớ.

```
public class String
```

String(String s)	create a string with the same value as s
int length()	number of characters
char charAt(int i)	the character at index i
String substring(int i, int j)	characters at indices i through (j-1)
boolean contains(String substring)	does this string contain substring?
boolean startsWith(String pre)	does this string start with pre?
boolean endsWith(String post)	does this string end with post?
int indexOf(String pattern)	index of first occurrence of pattern
int indexOf(String pattern, int i)	index of first occurrence of pattern after i
String concat(String t)	this string with t appended
int compareTo(String t)	string comparison
String toLowerCase()	this string, with lowercase letters
String toUpperCase()	this string, with uppercase letters
String replaceAll(String a, String b)	this string, with as replaced by bs
String[] split(String delimiter)	strings between occurrences of delimiter
boolean equals(Object t)	is this string's value the same as t's?
int hashCode()	an integer hash code

- StringBuilder và StringBuffer khá giống nhau, chỉ khác nhau ở: các phương thức của StringBuffer đồng bộ nên thích hợp khi làm việc với ứng dụng đa luồng

- **StringBuffer** – đồng bộ, **StringBuilder** – bất đồng bộ

- StringBuilder

```
StringBuilder strName = new StringBuilder("Value")
```

- StringBuffer

```
StringBuffer strName = new StringBuffer("Value")
```

Các phương thức	
append(value)	Nối chuỗi
insert(int index,String str)	Chèn chuỗi str vào vị trí index
delete(int start, int end)	Xóa từ ký tự có vị trí start đến ký tự có vị trí end
deleteCharAt(int index)	Xóa ký tự có vị trí index
reverse()	Đảo ngược chuỗi
replace(int start, int end, String str)	Thay chuỗi str vào từ vị trí start đến vị trí end
subsequence(int start, int end)	Lấy chuỗi từ vị trí start đến vị trí end

HỎI ĐÁP





TRẢI NGHIỆM THỰC HÀNH

HỆ THỐNG ĐÀO TẠO CNTT QUỐC TẾ BACHKHOA - APTECH



TRÂN TRỌNG CẢM ƠN!



238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội



0968.27.6996



tuyensinh@bachkhoa-aptech.edu.vn



www.bachkhoa-aptech.edu.vn