

**Sau khi thực hành xong bài này, học viên có khả năng :**

- Thực hành sử dụng lớp, phương thức với mảng một chiều

Xây dựng ứng dụng Java Application thực hiện các chức năng : Lưu trữ thông tin của sinh viên , tìm thông tin của sinh viên dựa vào ID , cập nhật thông tin sinh viên và in danh sách sinh viên ra màn hình.

**Bước 1** . Mở ứng dụng NetBean , New Project | Java | Java Application , nhấp Next , trên hộp thoại "New Java Application " điền các thông tin trong hộp thoại như sau :  
Project Name : **WS\_JavaClass**; project Location : "chọn đường dẫn lưu ứng dụng" ,  
nhấn Finish để thúc .

- 1.1 Nhấp phải chuột trên **<default package>** chọn New | Java Class , đặt tên lớp : ,  
nhấp Finish để kế thúc .
- 1.2 Nhấp phải chuột trên **<default package>** chọn New | Java Class , đặt tên lớp :  
**Manager**, nhấp Finish để kế thúc.
- 1.3 Nhấp phải chuột trên **<default package>** chọn New | Java Class, đặt tên lớp :  
**Test**, nhấp Finish để kế thúc.

**Bước 2**. Phát triển các lớp để thực hiện các chức năng .

2.1 Phát triển lớp **Student**, lớp này mô tả thông tin chi tiết cho sinh viên. Viết mã cho tập tin **Student.java** như sau :

```
import java.text.SimpleDateFormat;
import java.util.*;
/*
 * @author MichaelLake
 */
public class Student {
    private String mID;
    private String mFullName;
    private Date mDoB;
    private String mPhone;
    private String mAddress;
    public Student() {}
    public Student(String ID, String FullName, Date DoB,
        String Phone, String Address) {
        this.mID = ID;
        this.mFullName = FullName;
        this.mDoB = DoB;
        this.mPhone = Phone;
        this.mAddress = Address;
    }
    public String getID() {
        return mID;
    }
    public void setID(String ID) {
        this.mID = ID;
    }
    public String getFullName() {
        return mFullName;
    }
    public void setFullName(String FullName) {
        this.mFullName = FullName;
    }
    public Date getDoB() {
        return mDoB;
    }
}
```

```

public void setDoB(Date DoB) {
    this.mDoB = DoB;
}
public String getPhone() {
    return mPhone;
}
public void setPhone(String Phone) {
    this.mPhone = Phone;
}
public String getAddress() {
    return mAddress;
}
public void setAddress(String Address) {
    this.mAddress = Address;
}

```

2.2 Phát triển lớp **StudentManager** , lớp này thực hiện các thao tác : lưu trữ danh sách các sinh viên , thêm , tìm kiếm thông tin sinh viên và in danh sách các sinh viên ra màn hình . Viết mã cho tập tin **StudentManager.java** như sau :

```

/**
 *
 * @author MichaelLake
 */
public class StudentManager {
    //Khái báo mảng chứa danh sách sinh viên
    Student studentList[] = new Student[50];
    //số sinh viên đã lưu trữ
    int numOfStudent = 0;
    //phương thức lấy sinh viên theo index
    public Student getStudentByIndex(int index){
        return studentList[index];
    }

    //thêm sinh viên mới
    public int addStudent(Student newStudent){
        if(numOfStudent < studentList.length){
            studentList[numOfStudent++] = newStudent;
            return 1;
        }
        else{
            return 0;
        }
    }
}

```

```

public void printStudentList() {
    System.out.println("Student List :");
    for(int i=0;i<numOfStudent;i++){
        System.out.println(studentList[i].toString());
    }
} //end addStudent
//Tìm sinh viên theo ID
public int findStudent(String ID){
    int result = 0;
    //code to do here
    return result;
} //end findStudent
//Cập nhật
public void updateStudent(Student newStudent){
    //code to do here
} //end updateStudent
} //end class

```

2.3 Phát triển lớp **StudentManagerTest**, lớp này hiển thị một menu cho người dùng chọn các chức năng để thao tác : thêm , cập nhật , tìm kiếm thông tin sinh viên và in danh sách các sinh viên . Viết mã cho tập tin **StudentManagerTest.java** như sau :

```

import java.text.SimpleDateFormat;
import java.util.*;
/**...4 lines */
public class StudentTest {
    public static void main(String[] args){
        int choice ;
        String ID, FullName, Address, Phone;
        Date DoB = null;
        Scanner sc = new Scanner(System.in);
        StudentManager stdManager = new StudentManager();
        do{
            System.out.println("Select a function !");
            System.out.println("1.Add new a student");
            System.out.println("2.View student list");
            System.out.println("3.Find a student by ID");
            System.out.println("4.Update a student");
            System.out.println("5.Exit");
            System.out.print("Select :");
            choice = sc.nextInt();

```

```

switch(choice){
    case 1:
        //nhap ID
        System.out.print("Enter ID:");
        ID = sc.next();
        //Nhap FullName
        System.out.print("FullName:");
        FullName = sc.next();
        //Nhap ngay sinh
        System.out.print("DoB(dd/MM/yyyy):");
        SimpleDateFormat f = new SimpleDateFormat("dd/MM/yyyy");
        try{
            DoB = f.parse(sc.next());
        }
        catch(Exception ex){
            ex.printStackTrace();
        }
        //Nhap dia chi
        System.out.print("Address:");
        Address = sc.next();
        //Nhap phone
        System.out.print("Phone:");
        Phone = sc.next();
        //tao doi tuong student moi
        Student newStudent = new Student(ID, FullName, DoB, Phone, Address);
        //them vao danh sach StdudentList
        stdManager.addStudent(newStudent);
        break;
    case 2:
        stdManager.printStudentList();
        break;
    case 3:
        break;
    case 4:
        break;
    case 5:
        System.exit(0);
        break;
    default:
        System.out.println("Please try again.");
        break;
}

}while(true);
}

```

### **Bước 3**: Chạy ứng dụng và kiểm tra kết quả