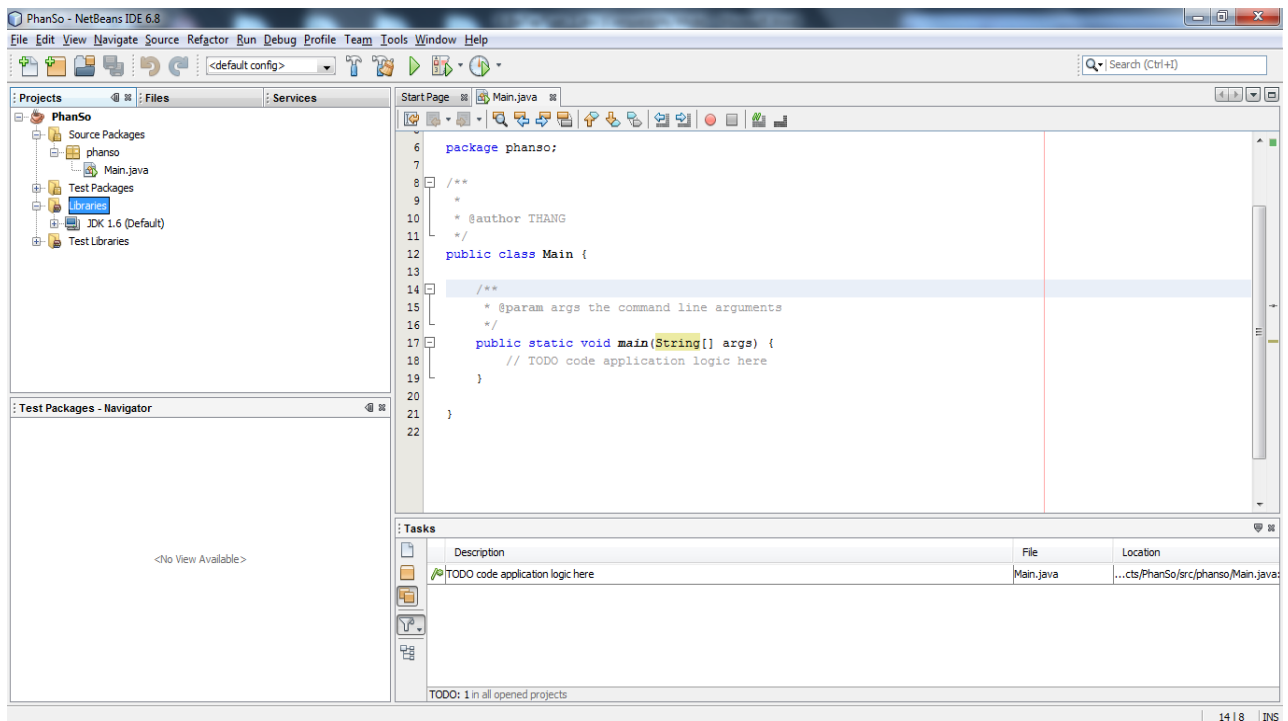


# CHUYÊN ĐỀ JAVA

## Hướng Dẫn TUẦN 1: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG JAVA

### 1. Phân số:

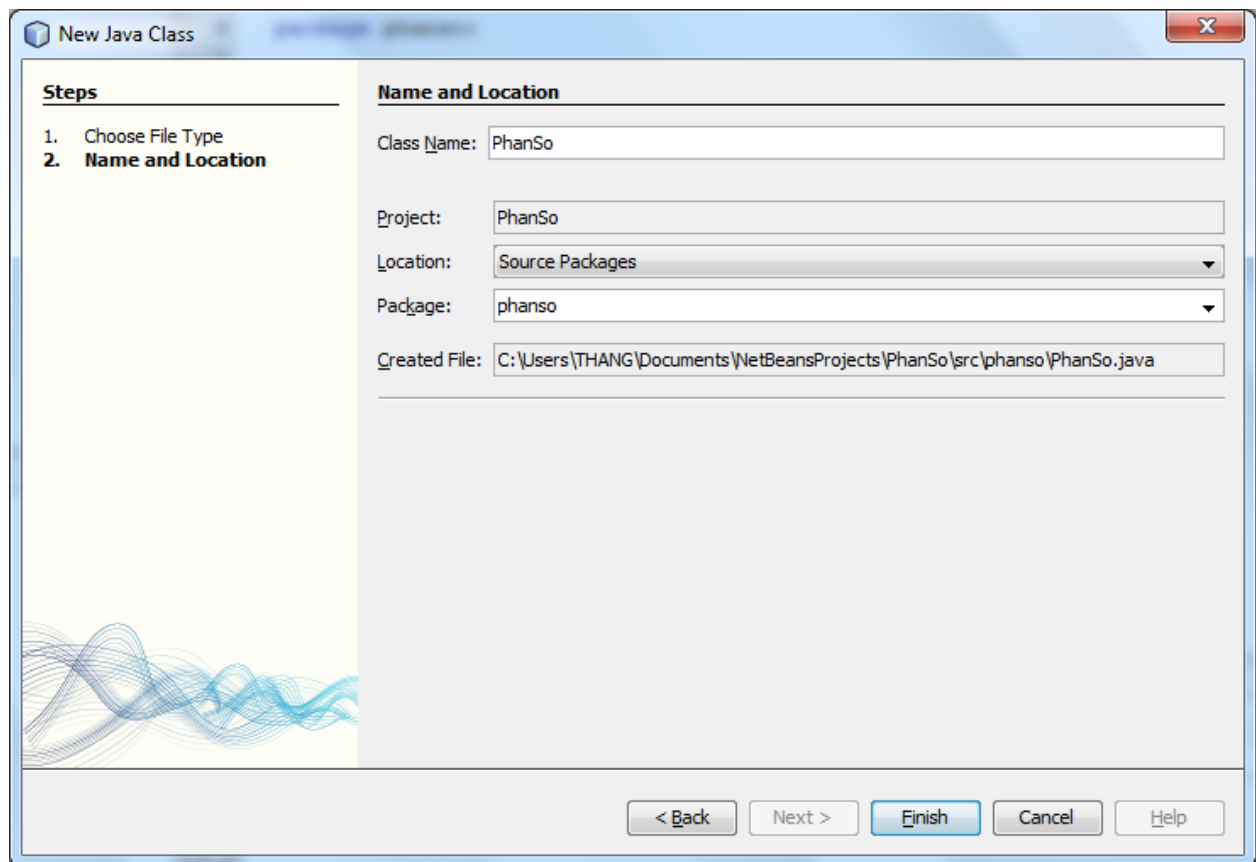
1. Tạo một project dạng java application và đặt tên PhanSo sử dụng netbeans:



Sau khi tạo project netbeans sẽ tự động tạo một package phanso tương ứng và file Main chứa class Main chứa các lệnh thực thi chương trình.

2. Right- click vào package phanso chọn new java class: đặt tên là PhanSo

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH  
KHOA CÔNG NGHỆ PHẦN MỀM  
MÔN CHUYÊN ĐỀ JAVA

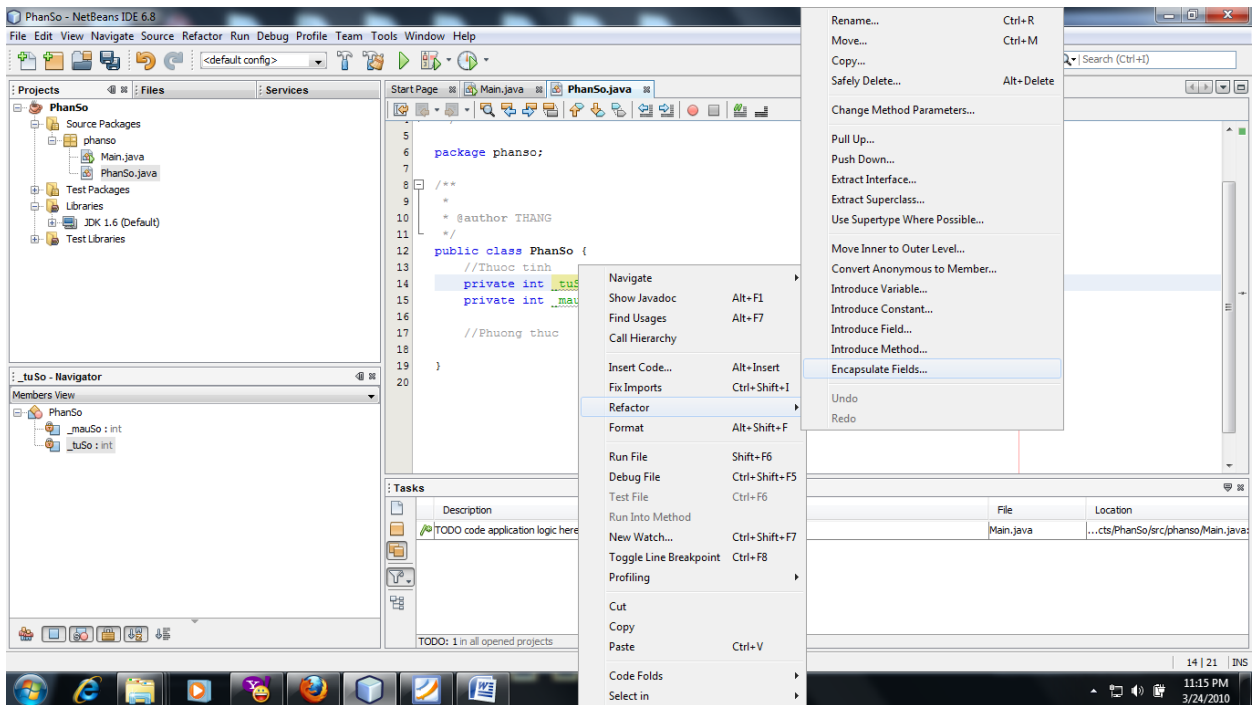


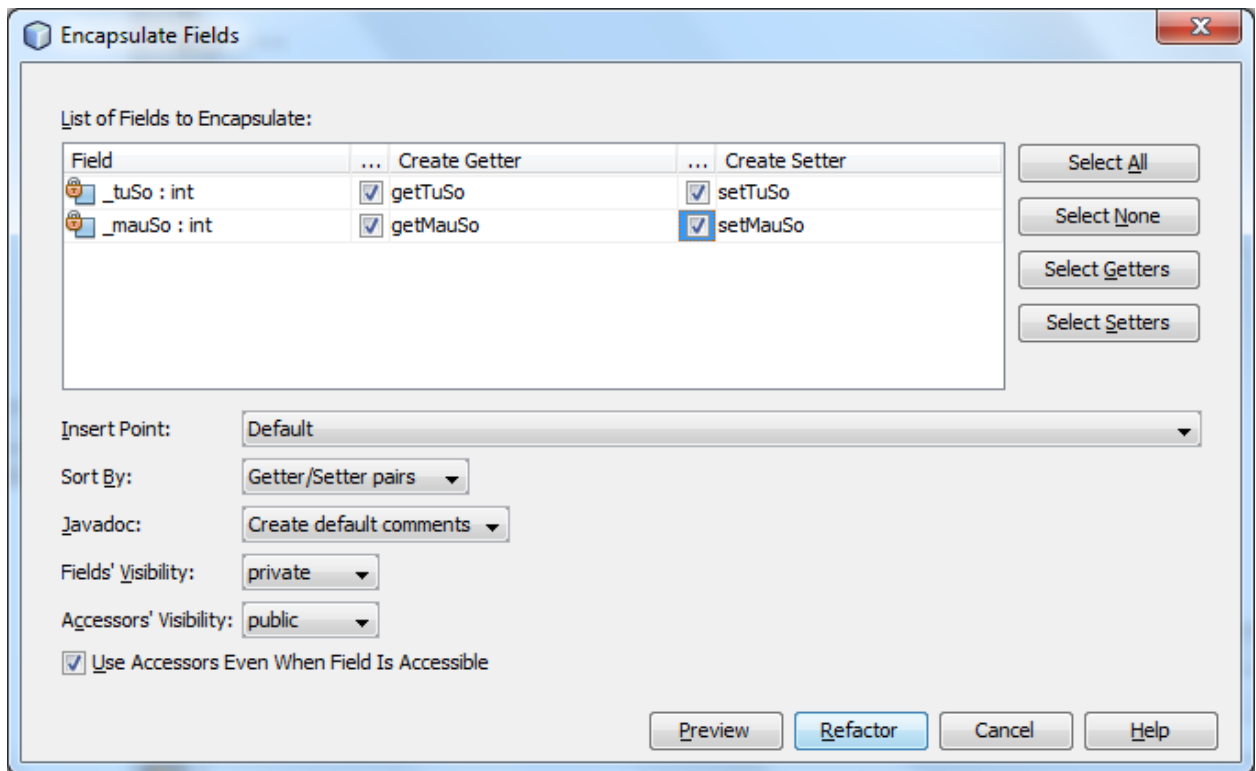
3. Khi đó bạn định nghĩa class phân số như sau:

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH  
KHOA CÔNG NGHỆ PHẦN MỀM  
MÔN CHUYÊN ĐỀ JAVA

```
public class PhanSo {  
    //Thuộc tính  
    private int tuSo;  
    private int mauSo;  
  
    /**  
     * @return the _tuSo  
     */  
    public int getTuSo() {  
        return _tuSo;  
    }  
  
    /**  
     * @param tuSo the _tuSo to set  
     */  
    public void setTuSo(int tuSo) {  
        this._tuSo = tuSo;  
    }  
}
```

4. Right-click vào thuộc tính: \_tuSo chọn Refactor-> Encapsulate Field...





Sau đó nhấn Refactor -> lúc này NetBeans sẽ tự động Generate code cho các hàm get/set cho từng thuộc tính private của lớp.

Lưu ý sau này khi thiết kế lớp các bạn sử dụng refactor để tạo Property ( các hàm get/set) cho từng thuộc tính của lớp.

Từ đây, mọi sự truy xuất vào thuộc tính lớp đều thông qua phương thức get/set này.

##### 5. Viết hàm nhập và xuất tham số:

```
public PhanSo() {  
    _tuSo = 0;  
    _mauSo = 1;  
}  
public PhanSo(int tu, int mau) {  
    _tuSo = tu;  
    _mauSo = mau;  
}  
public void nhapPhanSo() {  
    Scanner in = new Scanner(System.in);  
    System.out.print("Tu so = ");  
    _tuSo = in.nextInt();  
    System.out.print("Mau so = ");  
    _mauSo = in.nextInt();  
}  
public void xuatPhanSo() {  
    System.out.print(_tuSo + "/" + _mauSo);  
}
```

6. Viết hàm rút gọn phân số:

```
public int uocChungLonNhatCuaTuMau() {  
    int tu = Math.abs(_tuSo);  
    int mau = Math.abs(_mauSo);  
    if(tu == 0)  
        return mau;  
    while(tu*mau != 0) {  
        if(tu > mau)  
            tu = tu - mau;  
        else  
            mau = mau - tu;  
    }  
  
    return (tu + mau);  
}  
public PhanSo rutGon() {  
    PhanSo kq = new PhanSo();  
    kq._tuSo = _tuSo/uocChungLonNhatCuaTuMau();  
    kq._mauSo = _mauSo/uocChungLonNhatCuaTuMau();  
    return kq;  
}
```

7. Lưu ý trong ngôn ngữ java thì nạp chồng toán tử (overloading operator) không hỗ trợ nên từ câu 12 các bạn không cần làm. Các bạn tham khảo sách core java trang

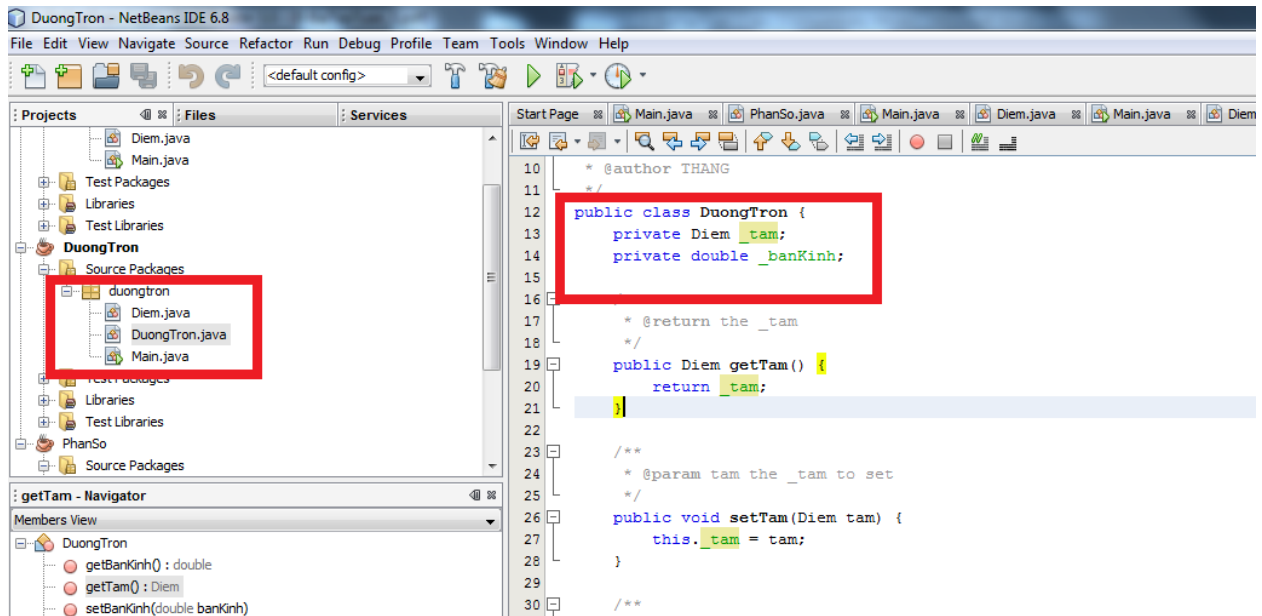
102 để biết lý do vì sao không hỗ trợ.

8. Các câu còn lại các bạn tự luyện.

## 2. Lớp điểm:

```
public class Diem {  
    private double _x;  
    private double _y;  
  
    /**  
     * @return the _x  
     */  
    public double getX() {  
        return _x;  
    }  
  
    /**  
     * @param x the _x to set  
     */  
    public void setX(double x) {  
        this._x = x;  
    }  
}
```

## 3. đường tròn:



#### 4. Tam giác:

```
public class TamGiac {  
    private Diem _A;  
    private Diem _B;  
    private Diem _C;
```