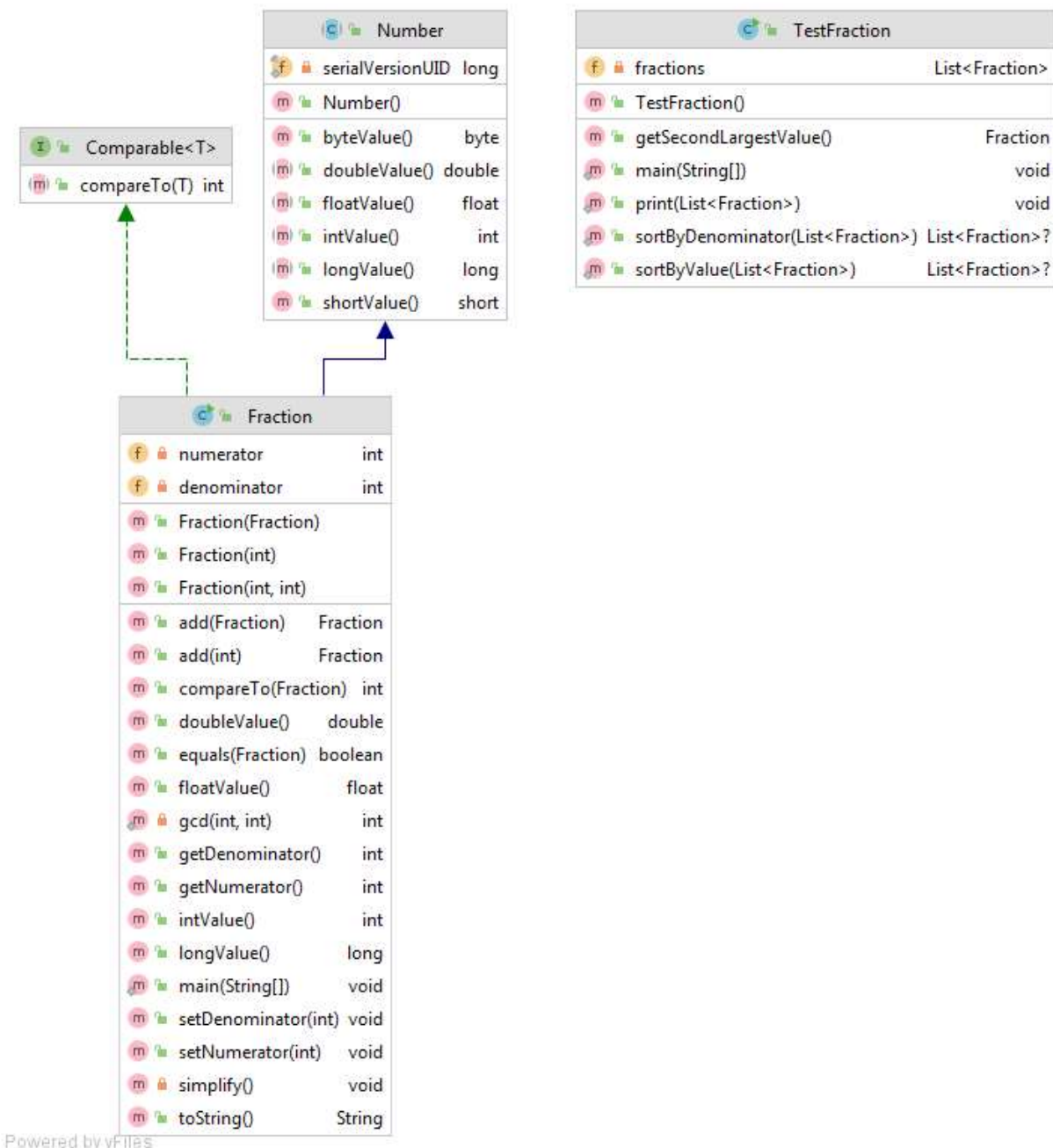


## Đề 1

**Câu 1.** Một chương trình Java được thiết kế như biểu đồ dưới đây.



Viết code cho các lớp của chương trình. Các lớp thuộc package “com.fraction”.

*Chú thích:*

Lớp service **Fraction** mô tả một phân số:

- Lớp **Fraction** thừa kế từ lớp **java.lang.Number** và implements giao diện **Comparable<Fraction>**.
- Lớp **Fraction** có 2 thuộc tính là *numerator* và *denominator*.
- Hàm dựng *Fraction(int)* khởi tạo tử số theo tham số truyền vào, khởi tạo mẫu số bằng 1.
- Hàm *add(Fraction)* cộng hai *Fraction*, trả về *Fraction* mới là kết quả của cộng *Fraction* hiện tại với *Fraction* truyền vào.
- Hàm *add(int)* cộng hai *Fraction*, trả về *Fraction* mới là kết quả của cộng *Fraction* hiện tại với *Fraction* có tử số là tham số truyền vào, mẫu số là 1.
- Hàm *compareTo(Fraction)* override hàm *compareTo* của *interface Comparable*, dùng để so sánh hai *Fraction*. *Fraction* A lớn hơn *Fraction* B nếu tử số của A nhân với mẫu số của B lớn hơn mẫu số của A nhân với tử số của B.
- Hàm *equals(Fraction)* kiểm tra hai *Fraction* có bằng nhau không. *Fraction* A bằng *Fraction* B nếu tử số của A nhân với mẫu số của B bằng mẫu số của A nhân với tử số của B.
- Hàm *gcd(int, int)* tính ước số chung lớn nhất của hai số. Nên dùng thuật toán Euclid để tính ước chung lớn nhất của hai số.
- Hàm *simplify()* rút gọn phân số đã cho về phân số tối giản.
- Hàm *toString()* trả về String mô tả phân số theo định dạng “Fraction[numerator/denominator]”.

Lớp test driver **TestFraction** thực hiện các yêu cầu sau:

- Tạo ngẫu nhiên 30 cặp số nguyên có giá trị nằm trong khoảng [1, 1000] để tạo ra 30 phân số *Fraction*, lưu các phân số này vào một List có tên là *fractions*.
- In ra các phân số trong list *fractions*; in ra các phân số theo thứ tự có giá trị tăng dần (theo sắp thứ tự của *Comparable*); in ra các phân số có mẫu số tăng dần; tìm phân số có giá trị lớn thứ hai trong list *fractions*. Mỗi thông tin được in trên một dòng.
- Việc sắp xếp các phân số có thể dùng các thuật toán sắp xếp đã học, ví dụ Selection Sort, Bubble Sort, ..., hoặc dùng các giao diện *Comparable* và *Comparator*.

**Bài 2.** Một chương trình Java được thiết kế như biểu đồ dưới đây.