

```

//Клас "Обикновена дроб"
public class Fraction {
// Свойства: числител и знаменател
private long num,den;

//Частни методи

//НОД на числителя и знаменателя
private long gcd() {
//променливи n и d - абсолютните стойности на числителя и знаменателя
//(методът НЕ ПРОМЕНЯ свойствата, работим с копия от тях)
long n=num>=0?num:-num,
d=den>0?den:-den,
t;
//Невъзможният резултат 0 подсказва, че поне едно от числата е било 0.
// (По принцип това ще е осигурено преди повикването, но като правим инструменти,
// предпочитаме да са по-всеобхватни.)
if (n==0||d==0) return 0;
//Оставяме в n не по-малкото от двете.
if (n<d){
t=n;
n=d;
d=t;
}
}
//Алгоритъм на Евклид с деление
do{
t=n%d;
if (t==0) return d;
n=d;
d=t;
}while(true);
}
//"Нормализиране" на дроб
private void normalize(){
//Проверка за знаменател 0
if (den==0) return;
//Проверка за числител 0
if (num==0){
den=1;
return;
}
//Знакът да се държи от числителя
if (den<0){
den=-den;
num=-num;
}
//Опростяване на дробта
long d=gcd();
num/=d;
den/=d;
}

//Публични методи

//"Празен" конструктор (без параметри): да създава дроб със стойност 0
public Fraction(){
num=0;
den=1;
}
//Конструктор с един параметър - превръща цяло в "обикновена дроб"
public Fraction (long n){
num=n;
den=1;
}
//Конструктор с два параметъра - числител и знаменател
public Fraction (long n,long d){
//Запомняне на входните параметри в свойствата
num=n;
den=d;
}

```

```

// "Нормализиране" на дробта
    normalize();
}
// "Гетъри": позволяват на външния потребител да ЧЕТЕ свойствата
public long getNum(){
    return num;
}
public long getDen(){
    return den;
}
// "Сетъри": позволяват на външния потребител да ПРОМЕНЯ свойствата
// ВНИМАНИЕ за адекватността на промените!
public void setNum(long n){
    num=n;
    normalize();
}
public void setDen(long d){
    den=d;
    normalize();
}
// Създаване на низ за визуализация на обект от новия клас
public String toString(){
    if (den==0) return "NaN";
    if (den==1) return Long.toString(num);
    return num+"/"+den;
}
// Действия с обекти от новия клас
// Събиране на дроби
public Fraction add(Fraction f){
    return new Fraction(this.num*f.den+this.den*f.num, this.den*f.den);
}
// Изваждане на дроби
public Fraction sub(Fraction f){
    return new Fraction(this.num*f.den-this.den*f.num, this.den*f.den);
}
// Умножение на дроби
public Fraction mul(Fraction f){
    return new Fraction(this.num*f.num, this.den*f.den);
}
// Деление на дроби (ВНИМАНИЕ при деление на нула!)
public Fraction div(Fraction f){
    if (f.num==0) return new Fraction(1,0); // Връща NaN
    return new Fraction(this.num*f.den, this.den*f.num);
}
// Намиране на реципрочна на дадена дроб
public Fraction reciprocal(){
    // 0 и NaN нямат реципрочни
    if (num==0 || den==0) return new Fraction(1,0); // Връща NaN
    return new Fraction(den,num);
}
// Повдигане на степен на дроб
public Fraction power(int p){
// Няма нулева степен на 0 и NaN
    if (p==0 && (num==0 || den==0)) return new Fraction(1,0);
    // Временна променлива t, равна на текущия обект
    // "Суматор" r, равен на 1 (ще трупаме произведение)
    Fraction t=this, r=new Fraction(1);
    // Ако p е отрицателно, сменяме t с неговото реципрочно
    // и правим p положително
    if (p<0) {
        p=-p;
        t=t.reciprocal();
    }
    // Цикъл за натрупване на произведението в r
    for (int i=0; i<p; i++) r=r.mul(t);
    return r;
}
}

```