

Ekstraopgave 1

```
public class Fraction implements Comparable {
    public Fraction(int a, int b) {
        if (b == 0)
            throw new RuntimeException("Fraction: 0 in denominator");
        if (a == 0) { this.a = 0; this.b = 1; return; }
        if (b < 0) { a = -a; b = -b; }
        int d = gcd(Math.abs(a), b);
        this.a = a/d; this.b = b/d;
    }

    public Fraction(int a) { this.a = a; this.b = 1; }

    public Fraction add(Fraction f)
        { return new Fraction(a * f.b + b * f.a, b * f.b); }

    public Fraction subtract(Fraction f)
        { return new Fraction(a * f.b - b * f.a, b * f.b); }

    public Fraction multiply(Fraction f)
        { return new Fraction(a * f.a, b * f.b); }

    public Fraction divide(Fraction f)
        { return new Fraction(a * f.b, b * f.a); }

    public int compareTo(Object obj) {
        return subtract((Fraction) obj).a;
    }

    public boolean equals(Object obj) {
        Fraction f = (Fraction) obj;
        return a == f.a && b == f.b;
    }

    public String toString() {
        return a == 0 || b == 1 ? "" + a : a + "/" + b;
    }

    private int a, b;

    private int gcd(int u, int v) {
        return v == 0 ? u : gcd(v, u % v);
    }

    public static void main(String args[]) {
        Fraction sum = new Fraction(0);
        for (int i = 1; i <= 10; i++)
            sum = sum.add(new Fraction(1, i));
        System.out.println(sum);
    }
}
```

```

import java.math.BigInteger;

public class Fraction implements Comparable {
    public Fraction(long a, long b) {
        this(BigInteger.valueOf(a), BigInteger.valueOf(b));
    }

    public Fraction(long a) {
        this(a, 1);
    }

    public Fraction(BigInteger a, BigInteger b) {
        if (a == null)
            throw new RuntimeException("Fraction: numerator is null");
        if (b == null)
            throw new RuntimeException("Fraction: denominator is null");
        if (b.compareTo(BigInteger.ZERO) == 0)
            throw new RuntimeException("Fraction: 0 in denominator");
        if (a.compareTo(BigInteger.ZERO) == 0)
            { this.a = BigInteger.ZERO; this.b = BigInteger.ONE; return; }
        if (b.compareTo(BigInteger.ZERO) < 0) { a.negate(); b.negate(); }
        BigInteger d = a.gcd(b);
        this.a = a.divide(d); this.b = b.divide(d);
    }

    public Fraction add(Fraction f)
        { return new Fraction(a.multiply(f.b).add(b.multiply(f.a)),
                             b.multiply(f.b)); }

    public Fraction subtract(Fraction f)
        { return new Fraction(a.multiply(f.b).subtract(b.multiply(f.a)),
                             b.multiply(f.b)); }

    public Fraction multiply(Fraction f)
        { return new Fraction(a.multiply(f.a), b.multiply(f.b)); }

    public Fraction divide(Fraction f)
        { return new Fraction(a.multiply(f.b), b.multiply(f.a)); }

    public int compareTo(Object obj)
        { return subtract((Fraction) obj).a.compareTo(BigInteger.ZERO); }

    public boolean equals(Object obj) { return compareTo(obj) == 0; }

    public String toString() {
        return a.compareTo(BigInteger.ZERO) == 0 ||
            b.compareTo(BigInteger.ONE) == 0 ?
            "" + a : a + "/" + b;
    }

    private BigInteger a, b;

```

```
public static void main(String args[]) {  
    Fraction sum = new Fraction(0);  
    for (int i = 1; i <= 100; i++)  
        sum = sum.add(new Fraction(1, i));  
    System.out.println(sum);  
}  
}
```

Udskriften fra de to programmer bliver henholdsvis

7381/2520

og

14466636279520351160221518043104131447711/2788815009188499086581352357412492142272