final List<BigDecimal> prices = Arrays.asList( new BigDecimal("10"), new BigDecimal("30"), new BigDecimal("17"), new BigDecimal("20"), new BigDecimal("15"), new BigDecimal("18"), new BigDecimal("45"), new BigDecimal("12"));

Old version

BigDecimal totalOfDiscountedPrices = BigDecimal.ZERO;

for(BigDecimal price : prices) { if(price.compareTo(BigDecimal.valueOf(20)) > 0) totalOfDiscountedPrices = totalOfDiscountedPrices.add(price.multiply(BigDecimal.valueOf(0.9))); } System.out.println("Total of discounted prices: " + totalOfDiscountedPrices);

New Version

final BigDecimal totalOfDiscountedPrices = prices.stream() .filter(price -> price.compareTo(BigDecimal.valueOf(20)) > 0) .map(price -> price.multiply(BigDecimal.valueOf(0.9))) .reduce(BigDecimal.ZERO, BigDecimal::add);

System.out.println("Total of discounted prices: " + totalOfDiscountedPrices);

Advantages:

We avoided explicit mutation or reassignment of variables, which are often sources of bugs and result in code that’s hard to make concurrent.