```java
package hyperDap.base.helpers;

/**
 * An abstract helper class for frequently used number comparisons that exceed one line. Its purpose
 * not only to simplify but rather to streamline programming and ensure comparisons in the project
 * follow well defined standards.
 *
 * @author soenk
 *
 */

public final class Comparator {

  private Comparator() {}

  public static boolean equalProportionate(double a, double b, double fractionalPrecision)
      throws IllegalArgumentException {
    if (fractionalPrecision < 0 || fractionalPrecision > 1) {
      throw new IllegalArgumentException(
          String.format("%s.equalProportionate has been passed %s as precision!", Comparator.class,
              fractionalPrecision));
    }
    if (b < (a * (1 - fractionalPrecision))) {
      return false;
    }
    if (b > (a * (1 + fractionalPrecision))) {
      return false;
    }
    return true;
  }

  /**
   * Evaluate whether with the chosen precision the two values are equal, i.e. if {@code a} plus or
   * minus {@code precision} contains {@code b}.
   *
   * @param a
   * @param b
   * @param precision
   * @return
   */
  public static boolean equalApprox(double a, double b, double precision) {
    if (b < (a - precision)) {
      return false;
    }
    if (b > (a + precision)) {
      return false;
    }
    return true;
  }

  /**
   * An encapsulation of {@link #equalApprox(double, double, double)} making use of
   * {@link Number#doubleValue()}.
   *
   * @param a
   * @param b
   * @param precision
   * @return
   */
  public static boolean equalApprox(Number a, Number b, Number precision) {
    return Comparator.equalApprox(a.doubleValue(), b.doubleValue(), precision.doubleValue());
  }

}
```