```java
/**
 * Τηρεί τα στοιχεία ανθρώπου που συμμετέχει σε πλειστηριασμό
 */
public class Person {

  private final String name; //το όνομά του

  /**
   * Κατασκευάζει άνθρωπο με συγκεκριμένο όνομα.
   * @param name Το όνομα του ανθρώπου
   */
  public Person(String name) {
    this.name = name; }

  /**
   * @return Το όνομα του ανθρώπου
   */
  public String getName() {
    return name; }
} //end class Person
```

```java
/**
 * Κλάση για προσφορές επί αγαθών ενός πλειστηριασμού
 */
public class Bid {

    private final Person bidder; // Ο άνθρωπος που κάνει την προσφορά
    private final long value;    // Η αξία της προσφοράς

  /**
   * Κατασκευάζει μια προσφορά.
   * @param bidder Ο άνθρωπος που κάνει την προσφορά
   * @param value Η αξία της προσφοράς
   */
  public Bid(Person bidder, long value) {
    this.bidder = bidder;
    this.value = value;
  }

  /**
   * @return Ο άνθρωπος που κάνει την προσφορά
   */
  public Person getBidder() {
    return bidder; }

  /**
   * @return Η αξία της προσφοράς
   */
  public long getValue() {
    return value; }
} //end class Bid
```

```java
/**
 * Τηρεί τα στοιχεία για τα αγαθά που εκπλειστηριάζονται
 */
public class Lot {

  private final int number;   // Κωδικός αριθμός
  private String description; // Περιγραφή
  private Bid highestBid;     // Μεγαλύτερη προσφορά για το αγαθό

  /**
   * Κατασκευάζει ένα αγαθό με συγκεκριμένο αριθμό και περιγραφή.
   * @param number Ο κωδικός του αγαθού
   * @param description Η περιγραφή του αγαθού
   */
  public Lot(int number, String description) {
    this.number = number;
    this.description = description;
    this.highestBid = null;
  }

  /**
   * @return Ο κωδικός αριθμός του αγαθού
   */
  public int getNumber() {
    return number; }

  /**
   * @return Η περιγραφή του αγαθού
   */
  public String getDescription() {
    return description; }

  /**
   * @return Η καλύτερη προσφορά για το αγαθό.
   * null αν δεν έχει γίνει καμιά προσφορά.
   */
  public Bid getHighestBid() {
    return highestBid; }

  /**
   * @return Τα στοιχεία της προσφοράς για εκτύπωση
   */
  public String toString() {
    String details = number + ": " + description;
    if (highestBid != null) details += "    Bid: " +  highestBid.getValue();
    else                    details += "    (No bid)";
    return details;
  }

  /**
   * Κάποιος έκανε μια προσφορά. Δες αν είναι η καλύτερη μέχρι στιγμής.
   * @param bid Η νέα προσφορά
   * @return true εάν είναι η καλύτερη, false διαφορετικά
   */
  public boolean bidFor(Bid bid) {
    if((highestBid == null) || (bid.getValue() > highestBid.getValue())) {
      highestBid = bid; // θέσε την ως την καλύτερη προσφορά για το αγαθό
      return true;
    }
    else
      return false;
  }
} //end class Lot
```

```java
import java.util.ArrayList;
import java.util.Iterator;

/**
 * Προσομοιώνει πλειστηριασμό που περιλαμβάνει αγαθά για τα οποία γίνονται προσφορές.
 */
public class Auction {

  private ArrayList lots;    // Η συλλογή των αγαθών που εκπλειστηριάζονται
  private int nextLotNumber; //Αριθμός που θα δοθεί στο επόμενο αγαθό που θα εισαχθεί

  /**
   * Κατασκευάζει έναν πλειστηριασμό.
   */
  public Auction() {
    lots = new ArrayList(); // αρχικά κανένα αγαθό
    nextLotNumber = 1;  // το επόμενο αγαθό θα πάρει αριθμό 1
  }

  /**
   * Εισάγει αγαθό στον πλειστηριασμό
   * @param description Η περιγραφή του υλικού
   */
  public void enterLot(String description) {
    lots.add(new Lot(nextLotNumber, description));
    nextLotNumber++;
  }

  /**
   * Προβάλλει όλα τα αγαθά του πλειστηριασμού
   */
  public void showLots() {
    Iterator it = lots.iterator();
    while ( it.hasNext() ) {
      Lot lot = (Lot) it.next();
      System.out.println(lot.toString());
    } // end while
  }

  /**
   * Δημιουργεί προσφορά . Εμφανίζει μήνυμα όταν είναι επιτυχής ή όχι.
   * @param number Ο αριθμός του αγαθού για το οποίο γίνεται η προσφορά
   * @param bidder Ο άνθρωπος που κάνει την προσφορά
   * @param value Η αξία της προσφοράς
   */
  public void bidFor(int lotNumber, Person bidder, long value) {
    Lot selectedLot = getLot(lotNumber);
    if (selectedLot != null)
      if (selectedLot.bidFor(new Bid(bidder, value)))
        System.out.println("Επιτυχής προσφορά για το αγαθό " + lotNumber);
      else
        System.out.println("Το αγαθό " + lotNumber + " έχει ήδη προσφορά " +
                           selectedLot.getHighestBid().getValue() );
  }

/**
 * Επιστρέφει την προσφορά με συγκεκριμένο αριθμό ή null αν δεν υπάρχει.
 * @param lotNumber Ο αριθμός της προσφοράς
 */
  public Lot getLot(int lotNumber) {
    if ( (lotNumber >= 1) && (lotNumber < nextLotNumber) ) { //υπάρχει το αγαθό
      Lot selectedLot = (Lot) lots.get(lotNumber - 1); // στην προηγούμενη θέση
      return selectedLot;
    } //end if
    else {
      System.out.println("Δεν υπάρχει αγαθό με αριθμό " + lotNumber);
```

```
67          return null;
68        } //end else
69      }
70  } //end class Auction
```