```java
import components.map.Map;
import components.map.Map1L;

/**
 *
 * @author Jahnavi Acharya
 */
public final class ToDoList {

    private String event;

    private Map<String, Integer> allEventsImp;
    private Map<String, Integer> allEventsDate;

    /**
     * No argument constructor--private to prevent instantiation.
     */
    public ToDoList() {
        this.event = "";
        this.allEventsImp = new Map1L<String, Integer>();
        this.allEventsDate = new Map1L<String, Integer>();
    }

    //requires event to not be an already existing not completed event in the to-do-list
    public void addEvent(String event) {
        if (this.allEventsImp.hasKey(event)
                && this.allEventsImp.value(event) == -1) {
            this.allEventsImp.replaceValue(event, 0);
            this.allEventsDate.replaceValue(event, 0);
        } else {
            this.allEventsImp.add(event, 0);
            this.allEventsDate.add(event, 0);
        }
    }

    //requires that importance is a number between 0-3 and date is a valid date in the format yearmonthday
    //requires event to not be an already existing not completed event in the to-do-list
    public void addEvent(String event, int importance, int date) {
        if (this.allEventsImp.hasKey(event)
                && this.allEventsImp.value(event) == -1) {
            this.allEventsImp.replaceValue(event, importance);
            this.allEventsDate.replaceValue(event, date);
        } else {
            this.allEventsImp.add(event, importance);
            this.allEventsDate.add(event, date);
        }
    }

    //requires that event is valid for all the next 4 methods
    public int getImportance(String event) {
        return this.allEventsImp.value(event);
    }

    public int getDate(String event) {
        return this.allEventsDate.value(event);
    }

    //requires that imp is between 0-3
    public void addImportance(String event, int imp) {
        this.allEventsImp.replaceValue(event, imp);
    }

    //requires that date is a valid date in the format YearMonthDay
    public void addDate(String event, int date) {
        this.allEventsDate.replaceValue(event, date);
    }

    //requires date is a valid date in the format YearMonthDay
    //requires that event already exists in the to-do list
    public void removeEvent(String event, int dateRem) {
        this.allEventsImp.replaceValue(event, -1);
        this.allEventsDate.replaceValue(event, dateRem);
    }

    //requires i to be a valid importance level (-1, 3)
    public String getStringImportance(int i) {
        String imp = "";
        if (i == 0) {
            imp = "No Importance";
        } else if (i == 1) {
            imp = "Low Importance";
        } else if (i == 2) {
            imp = "Medium Importance";
        } else if (i == 3) {
            imp = "High Importance";
        } else {
            imp = "Completed";
        }
        return imp;
    }

    public String getStringDate(int i) {
        String date = "";
        if (i == 0) {
            date = "No Date";
        } else {
            int dateInt = i;
            int year = dateInt / 10000;
            dateInt = dateInt % 10000;
            int month = dateInt / 100;
            dateInt = dateInt % 100;
            int day = dateInt;
            date = month + "/" + day + "/" + year;
        }
        return date;
    }

    //
    @Override
    public String toString() {
        String ans = "To-Do-List \nEvent, Importance or Completed, Date Completed or Date to be Completed\n";
        String imp = "";
        String date = "";
        for (Map.Pair<String, Integer> i : this.allEventsImp) {
            imp = this.getStringImportance(i.value());
            date = this.getStringDate(this.allEventsDate.value(i.key()));
            ans = ans + i.key() + ", " + imp + ", " + date + "\n";
        }
        return ans;
    }



    /**
     * Main method.
     *
     * @param args
     *            the command line arguments; unused here
     */
    public static void main(String[] args) {
        ToDoList check = new ToDoList();
        //Adds first event without any importance level or date to be completed by
        check.addEvent("Clean");
        //Adds second event with importance level and date to be completed by
        check.addEvent("Attend Meeting", 3, 240423);
        //Adds third event with date, no importance level
        check.addEvent("Have fun", 0, 240530);
        //completes Clean event and adds date when it was completed
        check.removeEvent("Clean", 240418);

        System.out.println(check.toString());

    }

}

```