

Lösningförslag, tentamen EDAA10 Programmering i Java

2011-01-10

```
1. public class Period {
    private Date start;
    private Date finish;

    public Period(Date date1, Date date2) {
        if (date1.before(date2)) {
            start = date1;
            finish = date2;
        } else {
            start = date2;
            finish = date1;
        }
    }

    public Date getStartDate() {
        return start;
    }

    public Date getFinishDate() {
        return finish;
    }

    public boolean contains(Date d) {
        return !d.before(start) && !d.after(finish);
    }
}

2. public class BirdObservationList {
    private ArrayList<BirdObservation> observations;

    public BirdObservationList() {
        observations = new ArrayList<BirdObservation>();
    }

    public void addObservation(String birdName, String location, Date date) {
        observations.add(new BirdObservation(BirdData.getNbr(birdName), location, date));
    }

    public void addObservation(int birdNbr, String location, Date date) {
        observations.add(new BirdObservation(birdNbr, location, date));
    }

    public void printAll() {
        for (int i = 0; i < observations.size(); i++) {
            System.out.println(observations.get(i));
        }
    }
}
```

```

public ArrayList<BirdObservation> getAllInPeriod(Period period) {
    ArrayList<BirdObservation> res = new ArrayList<BirdObservation>();
    for (int i = 0; i < observations.size(); i++) {
        if (period.contains(observations.get(i).getDate())) {
            res.add(observations.get(i));
        }
    }
    return res;
}

public ArrayList<BirdObservation> getAllAt(String location) {
    ArrayList<BirdObservation> res = new ArrayList<BirdObservation>();
    for (int i = 0; i < observations.size(); i++) {
        if (observations.get(i).getLocation().equals(location)) {
            res.add(observations.get(i));
        }
    }
    return res;
}
}

```

```

3. public class BirdCount {
    private int[] count;

    public BirdCount(ArrayList<BirdObservation> list) {
        count = new int[BirdData.NBR_BIRDS + 1];
        for (int i = 0; i < list.size(); i++) {
            count[list.get(i).getBirdNbr()]++;
        }
    }

    public int getCount(int birdNbr) {
        return count[birdNbr];
    }

    public ArrayList<SpeciesCount> allObserved() {
        ArrayList<SpeciesCount> sc = new ArrayList<SpeciesCount>();
        for (int i = 0; i < count.length; i++) {
            if (count[i] > 0) {
                sc.add(new SpeciesCount(i, count[i]));
            }
        }
        return sc;
    }

    public ArrayList<SpeciesCount> allObservedInCountOrder() {
        ArrayList<SpeciesCount> sc = new ArrayList<SpeciesCount>();
        for (int i = 0; i < count.length; i++) {
            if (count[i] > 0) {
                int pos = sc.size() - 1;
                while (pos >= 0 && sc.get(pos).getCount() < count[i]) {
                    pos--;
                }
                sc.add(pos + 1, new SpeciesCount(i, count[i]));
            }
        }
        return sc;
    }
}

```

```
4. public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        BirdObservationList bol = new BirdObservationList();
        boolean stop = false;
        while (!stop) {
            int year = scan.nextInt();
            if (year < 0) {
                stop = true;
            } else {
                int month = scan.nextInt();
                int day = scan.nextInt();
                String location = scan.next();
                String name = scan.next();
                bol.addObservation(name, location, new Date(year, month, day));
            }
        }
        ArrayList<BirdObservation> list = bol.getAllAt("Falsterbo");
        BirdCount bc = new BirdCount(list);
        System.out.println("De vanligaste fåglarna i Falsterbo är i ordning:");
        ArrayList<SpeciesCount> counts = bc.allObservedInCountOrder();
        int n = Math.min(counts.size(), 10);
        for (int i = 0; i < n; i++) {
            System.out.println(BirdData.getName(counts.get(i).getBirdNbr()));
        }
    }
}

5. private static ArrayList<SpeciesCount> intersection(ArrayList<SpeciesCount> obs1,
    ArrayList<SpeciesCount> obs2) {
    ArrayList<SpeciesCount> res = new ArrayList<SpeciesCount>();
    int i = 0;
    int j = 0;
    while (i < obs1.size() && j < obs2.size()) {
        SpeciesCount sc1 = obs1.get(i);
        SpeciesCount sc2 = obs2.get(j);
        if (sc1.getBirdNbr() < sc2.getBirdNbr()) {
            i++;
        } else if (sc1.getBirdNbr() > sc2.getBirdNbr()) {
            j++;
        } else {
            res.add(new SpeciesCount(sc1.getBirdNbr(), sc1.getCount() + sc2.getCount()));
            i++;
            j++;
        }
    }
    return res;
}
```
