

```

public interface MedicineAdministration { ---- 1      ---- /2

    public double calculateDose(int weight); ---- 1
}

public      abstract class Medicine ---- 1      ---- /5
           implements MedicineAdministration, Serializable { ---- 1

    private String name;
    private String expiry;
    protected double perKGDose;

    public Medicine(String s, String d, double pkgD) {
        name =s;
        expiry = d;
        perKGDose = pkgD;
    }

    public Medicine(Medicine m) { ---- 2
        name =m.name;
        expiry = m.expiry;
        perKGDose = m.perKGDose;
    }

    public String getExpiry() { ---- 1
        return expiry;
    }
}

public class Pill extends Medicine { ---- 1      ---- /6

    private int perPacket;
    private double perPill_mg;

    public Pill(String s, String d, double pkgD, int n, double mg) {
        super(s, d, pkgD); ---- 1
        perPacket = n;
        perPill_mg = mg;
    }

    public Pill(Pill p) { ---- 1      ---- /2
        super(p); ---- 1
        perPacket = p.perPacket;
        perPill_mg = p.perPill_mg;
    }

    public double calculateDose(int weight) { ---- 1      ---- /2

        double dose = (perKGDose * weight) / perPill_mg; ---- 1
        return dose;
    }
}

```

```

public class Pharmacy {                                ---- /39
    private int branchNo;
    private String city;

    private Medicine arMed[]; ---- 1
    private int nbMed; ---- 1

    public Pharmacy(int bn, String s, int size) { ---- /2
        branchNo = bn;
        city = s;
        arMed = new Medicine[size]; ---- 1
        nbMed = 0; ---- 1
    }

    public void addMedicine(Medicine m) ---- /9
        throws ArrayIndexOutOfBoundsException { ---- 1

        if (nbMed >= arMed.length) ---- 1
            throw new ---- 1
                ArrayIndexOutOfBoundsException("Wrong insert operation");

        if (m instanceof Pill) ---- 1
            arMed[nbMed] = new Pill((Pill) m); ---- 1+1
        else
            arMed[nbMed] = new Syrup((Syrup) m); ---- 1+1

        nbMed++; ---- 1
    }

    public Medicine[] getMedicines(String d) { ---- /7
        Medicine res[] = new Medicine[nbMed]; ---- 1
        int j = 0; ---- 1

        for (int i = 0; i < nbMed; i++) { ---- 1
            if (arMed[i].getExpiry().equals(d)) { ---- 1
                res[j++] = arMed[i]; ---- 1+1
            }
        }
        return res; ---- 1
    }

    public void saveSyrups(String fname, int v) throws IOException { --- /6
        File f = new File(fname);
        FileOutputStream fos = new FileOutputStream(f);
        ObjectOutputStream oos = new ObjectOutputStream(fos); ---- 1

        for (int i = 0; i < nbMed; i++) { ---- 1

            if ( arMed[i] instanceof Syrup && ---- 1
                ((Syrup) arMed[i]).getVolume() > v ) ---- 1+1
                oos.writeObject(arMed[i]); ---- 1
        }

        oos.close();
    }
}

```

```

public int readPill(String fname) throws IOException { ---- 1  -- /12
    File f = new File(fname);
    FileInputStream fis = new FileInputStream(f);
    ObjectInputStream ios = new ObjectInputStream(fis); ---- 1

    int n = 0; ---- 1
    Medicine m ; ---- 1

    try { ---- 1
        while (true) { ---- 1
            m = (Medicine) ios.readObject(); ---- 1+1

            if (m instanceof Pill) ---- 1
                n++; ---- 1
        }
    }
    catch (EOFException e) { ---- 1
        System.out.println("The end of file is reached");
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }

    ios.close();
    return n; ---- 1
}

public String getCity() { --- /1
    return city;
}
}

```

```

public int countPharmacies(String c) {                                --- /7
    int n = 0;                                                       ---- 1
    Node current = head;                                             ---- 1

    while (current != null) {                                         ---- 1
        if (current.getData().getCity().equals(c)) ---- 1
            n++;                                                       ---- 1

        current = current.getNext();                                   ---- 1
    }

    return n;                                                         ---- 1
}

public void split(           ListOfPharmacies riyaadhList,          --- /7
                        ListOfPharmacies jeddahList) {

    Node current = head;                                             ---- 1

    while (current != null) {                                         ---- 1
        if (current.getData().getCity().equals("Riyadh")) ---- 1
            riyaadhList.insertAtFront(current.getData()); ---- 1
        else
            if (current.getData().getCity().equals("Jeddah")) ---- 1
                jeddahList.insertAtFront(current.getData()); ---- 1

        current = current.getNext();                                   ---- 1
    }
}

```