

```

public class Menu {
    private String name;
    private Dish arrDish[];
    private int nbDish;

    public Menu(String s) {
        name = s;
        arrDish = new Dish[20];
        nbDish = 0;
    }

    public Menu(Menu m) {
        name = m.name;
        arrDish = new Dish[m.arrDish.length];

        /*
        nbDish = 0;
        for (int i = 0; i < m.nbDish; i++) {
            addDish(m.arrDish[i]);
        }
        */

        for (int i=0; i < m.nbDish; i++) {
            arrDish[i] = m.arrDish[i];
        }
        nbDish = m.nbDish;
    }

    public boolean addDish(Dish d) {
        if (nbDish < arrDish.length) {
            arrDish[nbDish] = d;
            nbDish++;
            return true;
        }
        else
            return false;
    }

    public void display() {
        System.out.println(name);
        for (int i = 0; i < nbDish; i++) {
            arrDish[i].display();
        }
    }

    public Dish cheapestDish() {
        Dish res = arrDish[0];

        for (int i=1; i < nbDish; i++)
            if (arrDish[i].getPrice() < res.getPrice())
                res = arrDish[i];

        return res;
    }

    public double avgDish() {
        double p = 0.0;

        for (int i=0; i < nbDish; i++)
            p += arrDish[i].getPrice();
    }
}

```

```
        if (nbDish > 0)
            return p / nbDish;
        else
            return 0.0;
    }
}

public class Dessert extends Menu {
    private String type;

    public Dessert(String n, String t) {
        super(n);
        type = t;
    }

    public Dessert(Dessert d) {
        super(d);
        type = d.type;
    }

    public String getType() {
        return type;
    }

    public void display() {
        super.display();
        System.out.println(type);
    }
}
```

```

public class Restaurant {

    private String name, location, contact;
    private Menu arrMenu[];
    private int nbMenu;

    public Restaurant(String n, String l, String c, int size) {
        name = n;
        location = l;
        contact = c;

        arrMenu = new Menu[size];
        nbMenu = 0;
    }

    public void addMenu(Menu m) {
        if (nbMenu < arrMenu.length) {
            if (m instanceof Lunch)
                arrMenu[nbMenu++] = new Lunch((Lunch) m);
            else
                if (m instanceof Dessert)
                    arrMenu[nbMenu++] = new Dessert((Dessert) m);
                else
                    arrMenu[nbMenu++] = new Menu(m);
        }
    }

    public Menu[] selectLunch(double a, int t) {
        Menu res [] = new Menu[arrMenu.length];
        int j = 0;

        for (int i = 0; i < nbMenu; i++) {
            if ( arrMenu[i] instanceof Lunch &&
                arrMenu[i].avgDish() < a &&
                ((Lunch)arrMenu[i]).getStartTime() == t)
                res[j++] = arrMenu[i];
        }
        return res;
    }

    public Menu bestLunchOffer() {
        Menu res = arrMenu[0];

        for (int i = 1; i < nbMenu; i++) {
            if ( arrMenu[i].cheapestDish().getPrice() <
                res.cheapestDish().getPrice() )
                res = arrMenu[i];
        }
        return res;
    }

    public Menu bestLunchOfferExpected() {
        Menu res = null;

        for (int i = 0; i < nbMenu; i++) {
            if ( arrMenu[i] instanceof Lunch ) {
                if ( res == null ||
                    arrMenu[i].cheapestDish().getPrice() <

```

```
        res.cheapestDish().getPrice() )
    res = arrMenu[i];
    }
}
return res;
}
}
```