

```

import java.util.Iterator;
//import java.util.Iterator;

//public
class MenuItem
{
    String name;
    String description;
    boolean vegetarian;
    double price;

    public MenuItem(String name, String description, boolean
vegetarian, double price)
    {
        this.name = name;
        this.description = description;
        this.vegetarian = vegetarian;
        this.price = price;
    }

    public String getName()
    {
        return name;
    }

    public String getDescription()
    {
        return description;
    }

    public double getPrice()
    {
        return price;
    }

    public boolean isVegetarian()
    {
        return vegetarian;
    }
}

public interface Menu {
    public Iterator<?> createIterator();

    String name;
    public String getName() {
        return name;
    }
}

//public
class PancakeHouseMenu implements Menu

```

```

{
    ArrayList<MenuItem> menuItems;

    public PancakeHouseMenu()
    {
        name = "BREAKFAST";
        menuItems = new ArrayList<MenuItem>();

        addItem("K&B's Pancake Breakfast",
                "Pancakes with scrambled eggs, and toast",true,
2.99);

        addItem("Regular Pancake Breakfast", "Pancakes with
fried eggs, sausage", false,2.99);

        addItem("Blueberry Pancakes","Pancakes made with fresh
blueberries, and blueberry syrup",true,3.49);

        addItem("Waffles","Waffles, with your choice of
blueberries or strawberries",true,3.59);
    }

    public void addItem(String name, String description, boolean
vegetarian, double price)
    {
        MenuItem menuItem = new MenuItem(name, description,
vegetarian, price);
        menuItems.add(menuItem);
    }

    public ArrayList<MenuItem> getMenuItems()
    {
        return menuItems;
    }

    public Iterator<MenuItem> createIterator()
    {
        return menuItems.iterator();
    }

    // other menu methods here
}

```

```

//import java.util.Iterator;

```

```

//public
class DinerMenu implements Menu
{
    static final int MAX_ITEMS = 6;
    int numberOfItems = 0;
    MenuItem[] menuItems;

```

```

public DinerMenu()
{
    name = "LUNCH";
    menuItems = new MenuItem[MAX_ITEMS];

    addItem("Vegetarian BLT",
        "(Fakin') Bacon with lettuce & tomato on whole
wheat", true, 2.99);
    addItem("BLT",
        "Bacon with lettuce & tomato on whole wheat",
false, 2.99);
    addItem("Soup of the day",
        "Soup of the day, with a side of potato salad",
false, 3.29);
    addItem("Hotdog",
        "A hot dog, with saurkraut, relish, onions,
topped with cheese",
        false, 3.05);
    addItem("Steamed Veggies and Brown Rice",
        "Steamed vegetables over brown rice", true,
3.99);
    addItem("Pasta",
        "Spaghetti with Marinara Sauce, and a slice of
sourdough bread",
        true, 3.89);
}

    public void addItem(String name, String description, boolean
vegetarian, double price)
    {
        MenuItem menuItem = new MenuItem(name, description,
vegetarian, price);
        if (numberOfItems >= MAX_ITEMS) {
            System.err.println("Sorry, menu is full!  Can't
add item to menu");
        } else {
            menuItems[numberOfItems] = menuItem;
            numberOfItems = numberOfItems + 1;
        }
    }

    public MenuItem[] getMenuItems()
    {
        return menuItems;
    }

    public Iterator<MenuItem> createIterator()
    {
        return new DinerMenuIterator(menuItems);
        //return new AlternatingDinerMenuIterator(menuItems);
    }

    // .public

    // other menu methods here
}

```

```

//public
class DinerMenuIterator implements Iterator<MenuItem>
{
    MenuItem[] list;
    int position = 0;

    public DinerMenuIterator(MenuItem[] list)
    {
        this.list = list;
    }

    public MenuItem next()
    {
        MenuItem menuItem = list[position];
        position = position + 1;
        return menuItem;
    }

    public boolean hasNext() {
        if (position >= list.length || list[position] == null)
        {
            return false;
        } else {
            return true;
        }
    }

    public void remove()
    {
        if (position <= 0) {
            throw new IllegalStateException
                ("You can't remove an item until you've
done at least one next()");
        }
        if (list[position-1] != null) {
            for (int i = position-1; i < (list.length-1);
i++) {
                list[i] = list[i+1];
            }
            list[list.length-1] = null;
        }
    }
}

```

```

//public
class Waitress

```

```

{
    ArrayList<Menu> menus;

    public Waitress(ArrayList<Menu> menus)
    {
        this.menus = menus;
    }

    public void printMenu() {
        Iterator<?> menuIterator = menus.iterator();

        System.out.print("MENU\n---\n");
        while(menuIterator.hasNext())
        {
            Menu menu = (Menu)menuIterator.next();
            System.out.print("\n" + menu.getName() + "\n");
            printMenu(menu.createIterator());
        }
    }

    void printMenu(Iterator<?> iterator)
    {
        while (iterator.hasNext())
        {
            MenuItem menuItem = (MenuItem)iterator.next();
            System.out.print(menuItem.getName() + ", ");
            System.out.print(menuItem.getPrice() + " -- ");
            System.out.println(menuItem.getDescription());
        }
    }
}

//public
class MenuTestDrive
{
    public static void main(String args[])
    {
        PancakeHouseMenu pancakeHouseMenu = new
PancakeHouseMenu();
        DinerMenu dinerMenu = new DinerMenu();
        ArrayList<Menu> menus = new ArrayList<Menu>();
        menus.add(pancakeHouseMenu);
        menus.add(dinerMenu);
        Waitress waitress = new Waitress(menus);
        waitress.printMenu();
    }
}

```