

Kwiaciarnia (uwaga: niezbędny do wykonania tego zadania materiał zawiera punkt o polimorfizmie w wykładzie, dotyczącym programowania obiektowego)

Napisać aplikację, która symuluje zakupy w kwiaciarni "samoobsługowej".

W kwiaciarni są kwiaty, kwiaty mają swoje nazwy oraz kolory. Ceny kwiatów znajdują się w cenniku.

Do kwiaciarni przychodzą klienci. Klienci mają imiona oraz dysponują jakimś zasobem pieniędzy. Wybierają kwiaty i umieszczają je na wózku sklepowym.

Następnie płacą za zawartość wózka i przepakowują ją do pudełka (jakiegoś pudełka na kwiaty :-).

Aplikacja wymaga zdefiniowania kilku klas i umiejętnego ich użycia, w taki sposób by następujący program działał poprawnie.

```
public class FloristsTest {
    // definicja metody sumowania wartosci kwiatów o podanym kolorze
    static int valueOf(Box box, String color) {
        /*<- tu trzeba wpisac kod metody */
    }

    public static void main(String[] args) {
        // Kwiaciarnia samoobsługowa
        // ustalenie cennika
        PriceList pl = PriceList.getInstance();
        pl.put("róża", 10.0);
        pl.put("bez", 12.0);
        pl.put("piwonia", 8.0);

        // Przychodzi klient janek. Ma 200 zł
        Customer janek = new Customer("Janek", 200);

        // Bierze różne kwiaty: 5 róż, 5 piwonii, 3 frezje, 3 bzy
        janek.get(new Rose(5));
        janek.get(new Peony(5));
        janek.get(new Freesia(3));
        janek.get(new Lilac(3));

        // Pewnie je umieścił na wózku sklepowym
        // Zobaczmy co tam ma
        ShoppingCart wozekJanka = janek.getShoppingCart();
        System.out.println("Przed płaceniem " + wozekJanka);

        // Teraz za to zapłaci...
        janek.pay();

        // Czy przypadkiem przy płaceniu nie okazało się,
        // że w koszu są kwiaty na które nie ustalono jeszcze ceny?
        // W takim razie zostałyby usunięte z wózka i Janek nie płaciłb
```

y za nie

```

        // Również może mu zabraknąć pieniędzy, wtedy też kwiaty są odk
ładane.

        System.out.println("Po zapłaceniu " + janek.getShoppingCart());

        // Ile Jankowi zostało pieniędzy?
        System.out.println("Jankowi zostało : " + janek.getCash() + " z
ł");

        // Teraz jakos zapakuje kwiaty (może do pudełka)
        Box pudełkoJanka = new Box(janek);
        janek.pack(pudełkoJanka);

        // Co jest teraz w wózku Janka...
        // (nie powinno już nic być)
        System.out.println("Po zapakowaniu do pudełka
" + janek.getShoppingCart());

        // a co w pudełku
        System.out.println(pudełkoJanka);

        // Zobaczymy jaka jest wartość czerwonych kwiatów w pudełku Jank
a
        System.out.println("Czerwone kwiaty w pudełku Janka kosztowały:
"
            + valueOf(pudełkoJanka, "czerwony"));

        // Teraz przychodzi Stefan
        // ma tylko 60 zł
        Customer stefan = new Customer("Stefan", 60);

        // Ale nabrał kwiatów nieco za dużo jak na tę sumę
        stefan.get(new Lilac(3));
        stefan.get(new Rose(5));

        // co ma w wózku
        System.out.println(stefan.getShoppingCart());

        // płaci i pakuje do pudełka
        stefan.pay();
        Box pudełkoStefana = new Box(stefan);
        stefan.pack(pudełkoStefana);

        // co ostatecznie udało mu się kupić
        System.out.println(pudełkoStefana);
        // ... i ile zostało mu pieniędzy
        System.out.println("Stefanowi zostało : " + stefan.getCash() +
" zł");
    }
}

```

Program ten wyprowadzi na konsolę:

```

Przed płaceniem Wózek właściciel Janek
róża, kolor: czerwony, ilość 5, cena 10.0
piwonia, kolor: czerwony, ilość 5, cena 8.0
frezja, kolor: żółty, ilość 3, cena -1.0
bez, kolor: biały, ilość 3, cena 12.0
Po zapłaceniu Wózek właściciel Janek
róża, kolor: czerwony, ilość 5, cena 10.0
piwonia, kolor: czerwony, ilość 5, cena 8.0
bez, kolor: biały, ilość 3, cena 12.0

```

Jankowi zostało : 74.0 zł
Po zapakowaniu do pudełka Wózek właściciel Janek -- pusto
Pudełko właściciel Janek
róża, kolor: czerwony, ilość 5, cena 10.0
piwonia, kolor: czerwony, ilość 5, cena 8.0
bez, kolor: biały, ilość 3, cena 12.0
Czerwone kwiaty w pudełku Janka kosztowały: 90
Wózek właściciel Stefan
bez, kolor: biały, ilość 3, cena 12.0
róża, kolor: czerwony, ilość 5, cena 10.0
Pudełko właściciel Stefan
bez, kolor: biały, ilość 3, cena 12.0
Stefanowi zostało : 24.0 zł

Uwaga: kod tego programu można zmienić tylko w miejscu zaznaczonym na zielono.

Dodawanie do programu nowych rodzajów kwiatów ma być bardzo łatwe. Przy dodaniu nowego rodzaju kwiatów nie wolno modyfikować żadnych innych klas programu.

Wymagania dodatkowe:

- należy wykorzystać klasy abstrakcyjne i **polimorfizm**
- należy zminimalizować kod klas ShoppingCart i Box
- należy zdefiniować klasę PriceList jako **singleton** (możemy mieć zawsze tylko jeden cennik)

Ważne uwagi.

- W kwiaciarni mogą być kwiaty, których zapomniano dodać do cennika. Wtedy przy płaceniu są one usuwane z naszego wózka.
- Może się okazać, że klient nie dysponuje odpowiednią kwotą pieniędzy aby zapłacić za całą zawartość wózka. Wtedy z wózka usuwane są kwiaty, za które klient nie może zapłacić (ale nie pojedynczo, tylko w kompletach np. po stefan.get(new Lilac(3)) usuwane są te trzy bzy na które Stefan nie ma pieniędzy).
- Warto zwrócić uwagę na odpowiednio zdefiniowanie metody toString() w niektórych klasach.

I na koniec: nie przejmujemy się tym, że np. róże mogą mieć wiele kolorów. Dla uproszczenia przyjęliśmy, że róże są czerwone itd.