

פתרון שאלת מר חלבי

הפתרון מציג אפשרות של יצירת שתי מחלקות אחת למשלוח ואחת לאוסף בחנות.

בבחירה זו אין שימוש בפעולות סטטיות אלא כל הפעולות הן פעולות מחלקה.

סעיף א

הייצוג כולל שתי מחלקות

מחלקת משלוח

```
public class Delivery
{
    private string date; //תאריך אחרון לשיווק
    private int amount; //כמות במשלוח
}
```

מחלקת מכולת

הכוללת את שני אוספים מסוג תור שכל איבר בהם הוא משלוח. תור אחד למשלוחי הגבינה ותור שני

למשלוחי היוגורט. המשלוח האחרון תמיד יכנס לסוף התור, והמכירה תתבצע מתחילת התור.

```
public class Store
{
    private double cheesePrice; // מחיר הגבינה
    private double yogurtPrice; // מחיר היוגורט
    private Queue<Delivery> cheeseList; // תור של משלוחי גבינה
    private Queue<Delivery> yogurtList; // תור של משלוחי יוגורט
}
```

סעיף ב

בקשת רכישה מהלקוח, הכוללת את שם המוצר וכמותו, ומעדכנת את המאגר בהתאם.
פעולה זו שייכת למחלקת מכולת

```
public void CustomerRequest(string name, int amount)
{
    Delivery temp;
    if (name == "yogurt")
    {
        while (amount > 0)
        {
            temp = this.yogurtList.Head();
            if (temp.GetAmount() > amount)
            {
                temp.SetAmount(temp.GetAmount() - amount);
                amount = 0;
            }
            else
            {
                amount = amount - temp.GetAmount();
                this.yogurtList.Remove();
            }
        }
    }
    else //רכישת גבינה
    {
        while (amount > 0)
        {
            temp = this.cheeseList.Head();
            if (temp.GetAmount() > amount)
            {
                temp.SetAmount(temp.GetAmount() - amount);
                amount = 0;
            }
            else
            {
                amount = amount - temp.GetAmount();
                this.cheeseList.Remove();
            }
        }
    }
}
```

סעיף ב

1. פעולה לקבלת משלוח מוצרים. כתבתי שתי פעולות נפרדות אחת לגבינה ואחת ליוגורט, אפשר לכתוב פעולה אחת שתקבל גם את שם המוצר ותפעיל את אחת משתי הפעולות בהתאם.

```
public void AddCheeseDelivery(string date, int amount)
{
    Delivery d1 = new Delivery(date, amount);
    this.cheeseList.Insert(d1);
}
```

```
public void AddYogurtDelivery(string date, int amount)
{
    Delivery d1 = new Delivery(date, amount);
    this.yogurtList.Insert(d1);
}
```

2. הוצאה מהמלאי של מוצרים שפג תוקפם

```
public void RemoveOldProducts(string today)
{
    Delivery temp = cheeseList.Head();
    while (temp.GetDate().CompareTo(today) == -1)
    {
        cheeseList.Remove();
        temp = cheeseList.Head();
    }
    temp = yogurtList.Head();
    while (temp.GetDate().CompareTo(today) == -1)
    {
        yogurtList.Remove();
        temp = yogurtList.Head();
    }
}
```