

IK

Inżynieria Komputerowa

mgr inż. Patryk Kaczmarek
patryk.kaczmarek@ansleszno.pl

Zapoznanie z praktycznymi problemami występującymi podczas projektowania i programowania aplikacji na potrzeby biznesu

Zaprojektować (oprogramować) algorytm obliczania kosztów transportu dla zakupów w sklepie internetowym uwzględniający następujące założenia:

- a) koszt dostawy = 24zł
- b) dla zakupów powyżej 1000 zł dostawa bezpłatna
- c) dla towarów o dużych gabarytach są ustalone indywidualnie specjalne ceny transportu

Zaproponuj alternatywne rozwiązanie problemu z uwzględnieniem masy towaru.

Zapoznanie z praktycznymi problemami występującymi podczas projektowania i programowania aplikacji na potrzeby biznesu

```
165
166 // TUTAJ ROBIMY MODYFIKACJE
167
168 1 reference
169 private void InicjalizacjaDanych()
170 {
171     // tutaj dodajemy dane
172
173     Towary.Add(new Towar() { ID = 1, Nazwa = "Czajnik Bezprzewodowy", Cena = 199.99M, Waga = 1.00M });
174     Towary.Add(new Towar() { ID = 2, Nazwa = "Deska do krojenia", Cena = 25.99M, Waga = 0.50M });
175
176     TransportKurier.Add(new Transport() { Nazwa = "InPost", KosztPrzesylki = 24.00M, WagaPrzesylki = 25.00M });
177
178     BindowanieDanych();
179 }
180
181
182 1 reference
183 private void ObliczKosztTransportu()
184 {
185     decimal KosztTransportu = 0M;
186     decimal LacznaWaga = 0M;
187     // Lista wybranych: Pozycje
188     // wybrany Kurier: WybranyTransport
189
190     foreach (Pozycja p in Pozycje)
191     {
192         // tutaj operacje na pozycjach listy
193     }
194
195     WyszwietlDane(KosztTransportu, LacznaWaga);
196 }
197
```

Zapoznanie z praktycznymi problemami występującymi podczas projektowania i programowania aplikacji na potrzeby biznesu

1. Zapoznaj się z aktualnym działaniem programu.
 2. Dodaj towary do programu (dodaj kilka z flagą gabaryt)
 3. Stwórz algorytm spełniający wszystkie założenia zadania.
Modyfikuj funkcje:
-

1. Dodaj kilka nowym przesyłek kurierskich

```
TransportKurier.Add(new Transport() { Nazwa = "InPost Gabaryt", KosztPrzesylki = 299.99M, Gabaryt = true });  
TransportKurier.Add(new Transport() { Nazwa = "DPD", KosztPrzesylki = 29.00M, WagaPrzesylki = 30.00M });  
TransportKurier.Add(new Transport() { Nazwa = "Poczta Polska", KosztPrzesylki = 35.00M, WagaPrzesylki = 50.00M });  
TransportKurier.Add(new Transport() { Nazwa = "Poczta Polska Gabaryt", KosztPrzesylki = 499.00M, Gabaryt = true });
```

2. Dodaj kilka nowym towarów

Pamiętaj o zwiększaniu pola ID – dla nowych towarów – pole musi być unikalne

```
Towary.Add(new Towar() { ID = 3, Nazwa = "Worek Piasku 25kg", Cena = 15.99M, Waga = 25.00M });  
Towary.Add(new Towar() { ID = 4, Nazwa = "Worek Piasku 35kg", Cena = 20.99M, Waga = 35.00M });  
Towary.Add(new Towar() { ID = 5, Nazwa = "Drzwi drewniane STOL 10", Cena = 499.99M, Waga = 80.00M, Gabaryt = true });  
Towary.Add(new Towar() { ID = 6, Nazwa = "Lodówka FROST 15", Cena = 2599.99M, Waga = 110.00M, Gabaryt = true });  
Towary.Add(new Towar() { ID = 7, Nazwa = "Zastawa kuchenna Porcelana", Cena = 899.00M, Waga = 25.00M });|
```

Można dodać więcej...

3. Sprawdź działanie programu

Zobacz czy dodane towary i kurierzy wyświetlają się poprawnie w programie

4. Dodaj towar do kalkulacji

Dodaj towar: Worek piasku 25kg – 1 sztuka

Wybierz kuriera: Inpost

Kliknij Oblicz koszt – coś się zmieniło?

5. Dodanie funkcji liczenia wagi

Dodajemy funkcjonalność liczącą wagę towarów w koszyku:

```
private void ObliczKosztTransportu()
{
    decimal KosztTransportu = 0M;
    decimal LacznaWaga = 0M;
    // Lista wybranych: Pozycje
    // wybrany Kurier: WybranyTransport

    foreach (Pozycja p in Pozycje)
    {
        // tutaj operacje na pozycjach listy
        LacznaWaga += p.Waga;
    }

    WswietlDane(KosztTransportu, LacznaWaga);
}
```

Ponownie sprawdź czy teraz funkcja oblicz koszt coś pokazuje.

Dodaj również inne towary w celu sprawdzenia poprawności rozwiązania

Dodaj 2 worki piasku i sprawdź działanie

6. Popraw funkcję

Popraw zapis na prawidłowy i ponownie przetestuj działanie

```
1 reference
private void ObliczKosztTransportu()
{
    decimal KosztTransportu = 0M;
    decimal LacznaWaga = 0M;
    // Lista wybranych: Pozycje
    // wybrany Kurier: WybranyTransport

    foreach (Pozycja p in Pozycje)
    {
        // tutaj operacje na pozycjach listy
        LacznaWaga += p.WagaRazem;
    }

    WyswietlDane(KosztTransportu, LacznaWaga);
}
```

7. Kolejna funkcja

Napiszmy funkcję sprawdzającą czy na liście znajduje się gabaryt

```
1 reference
private void ObliczKosztTransportu()
{
    decimal KosztTransportu = 0M;
    decimal LacznaWaga = 0M;
    // Lista wybranych: Pozycje
    // wybrany Kurier: WybranyTransport

    bool gabaryt = false;

    foreach (Pozycja p in Pozycje)
    {
        // tutaj operacje na pozycjach listy

        LacznaWaga += p.WagaRazem;

        if(p.Gabaryt) gabaryt = true;
    }

    WswietlDane(KosztTransportu, LacznaWaga);
}
```

8. Nie widzimy informacji

Zmodyfikujmy istniejącą funkcję WyświetlDane – o nowy parametr gabaryt

```
157 private void WyświetlDane(decimal KosztTransportu, decimal LacznaWaga, bool gabaryt)
158 {
159     lb_KosztTransport.Text = $"{KosztTransportu} PLN";
160     lb_Waga.Text = $"{LacznaWaga} kg";
161
162     if (gabaryt) lb_Waga.Text += $" - gabaryt";
163 }
164
```

Sprawdź ponownie działanie programu

```
1 reference
private void ObliczKosztTransportu()
{
    decimal KosztTransportu = 0M;
    decimal LacznaWaga = 0M;
    // Lista wybranych: Pozycje
    // wybrany kurier: WybranyTransport

    bool gabaryt = false;

    foreach (Pozycja p in Pozycje)
    {
        // tutaj operacje na pozycjach listy

        LacznaWaga += p.WagaRazem;

        if(p.Gabaryt) gabaryt = true;
    }

    WyświetlDane(KosztTransportu, LacznaWaga, gabaryt);
}
```

9. Kolejna funkcja

Sprawdzenie czy wybrany transport obsługuje gabaryt

```
foreach (Pozycja p in Pozycje)
{
    // tutaj operacje na pozycjach listy

    LacznaWaga += p.WagaRazem;

    if(p.Gabaryt) gabaryt = true;
}

// sprawdź czy jest kurier dla gabarytu
if(gabaryt && !WybranyTransport.Gabaryt)
{
    MessageBox.Show($"Wybrany transport {WybranyTransport} - nie obsługuje gabarytu");
    WyszwietlDane(KosztTransportu, LacznaWaga, gabaryt);
    return;
}

WyszwietlDane(KosztTransportu, LacznaWaga, gabaryt);
```

9. Wyświetl koszt transportu

Wyświetlenie kosztu transportu

```
// sprawdź czy jest kurier dla gabarytu
if(gabaryt && !WybranyTransport.Gabaryt)
{
    MessageBox.Show($"Wybrany transport {WybranyTransport} - nie obsługuje gabarytu");
    WyświetlDane(KosztTransportu, LacznaWaga, gabaryt);
    return;
}

KosztTransportu = WybranyTransport.KosztPrzesylki;

WyświetlDane(KosztTransportu, LacznaWaga, gabaryt);
```

Ponownie sprawdź działanie programu – czy wszystko się zgadza

10. Testy

Dodaj 2 worki piasku 25kg i wybierz przesyłkę Inpost

Jaki jest koszt transportu?

Zadeklarowana waga kuriera zgadza się z tym co mamy w koszyku?

11. Zaawansowany mechanizm kalkulacji

Zmodyfikujmy nasz algorytm aby stworzył kilka „przesyłek” dla wagi

```
decimal brakujacaWaga = LacznaWaga;
if(LacznaWaga <= WybranyTransport.WagaPrzesylki)
{
    KosztTransportu = WybranyTransport.KosztPrzesylki;
}
else
{
    bool obliczaj = true;
    // jeśli więcej dodaj kolejną przesyłkę
    do
    {
        KosztTransportu += WybranyTransport.KosztPrzesylki;
        brakujacaWaga -= WybranyTransport.WagaPrzesylki;

        if (brakujacaWaga <= 0) { obliczaj = false; }

    } while (obliczaj);
}
```

12. Czy przyjęto poprawne założenie

Czy w podanym algorytmie uwzględniono gabaryt?

Ze względu na specyfikację danych oraz algorytm – jeśli w koszyku znajdzie się gabaryt – program się zawiesi.

Musimy zmodyfikować ponownie nasz algorytm aby obsługiwał gabaryty, dla których to nie mamy podanych wag przesyłki – dlatego nigdy nie zostanie obliczona.

13. Modyfikacja algorytmu

Zmodyfikuj algorytm

```
if (!gabaryt)
{
    decimal brakujacaWaga = LacznaWaga;
    if (LacznaWaga <= WybranyTransport.WagaPrzesylki)
    {
        KosztTransportu = WybranyTransport.KosztPrzesylki;
    }
    else
    {
        bool obliczaj = true;
        // jeśli więcej dodaj kolejną przesyłkę
        do
        {
            KosztTransportu += WybranyTransport.KosztPrzesylki;
            brakujacaWaga -= WybranyTransport.WagaPrzesylki;

            if (brakujacaWaga <= 0) { obliczaj = false; }
        } while (obliczaj);
    }
}
else
{
    KosztTransportu = WybranyTransport.KosztPrzesylki;
}
```

14. Modyfikacja algorytmu

W programie jest jeszcze jeden błąd. Jak wybierzemy towary bez gabarytu, ale wybierzmy transport gabaryt... sprawdź

14. Modyfikacja algorytmu

Naprawa błędu

```
if (!gabaryt)
{
    decimal brakujacaWaga = LacznaWaga;
    if (LacznaWaga <= WybranyTransport.WagaPrzesylki)
    {
        KosztTransportu = WybranyTransport.KosztPrzesylki;
    }
    else
    {
        bool obliczaj = true;
        // jeśli więcej dodaj kolejną przesyłkę
        do
        {
            KosztTransportu += WybranyTransport.KosztPrzesylki;
            if (WybranyTransport.WagaPrzesylki == 0) break;

            brakujacaWaga -= WybranyTransport.WagaPrzesylki;

            if (brakujacaWaga <= 0) { obliczaj = false; }

        } while (obliczaj);
    }
}
else
{
    KosztTransportu = WybranyTransport.KosztPrzesylki;
}
```

15. Darmowa dostawa powyżej 1000 PLN

Musimy dołożyć zmienną, która przechowa nam wartość koszyka

```
1 reference
private void ObliczKosztTransportu()
{
    decimal KosztTransportu = 0M;
    decimal LacznaWaga = 0M;
    decimal WartośćZamowienia = 0M;
    // Lista wybranych: Pozycje
    // wybrany Kurier: WybranyTransport
}
```

```
foreach (Pozycja p in Pozycje)
{
    // tutaj operacje na pozycjach listy

    LacznaWaga += p.WagaRazem;

    if(p.Gabaryt) gabaryt = true;

    WartośćZamowienia += p.Ilosc * p.Towar.CenaBrutto;
}
```

Sprawdź działanie programu

16. Dokładamy warunek

Musimy dołożyć warunek sprawdzający wartość zamówienia

```
bool darmowaDostawa = false;
if(WartoscZamowienia > 1000)
{
    KosztTransportu = 0;
    darmowaDostawa = true;
}

WyswietlDane(KosztTransportu, LacznaWaga, gabaryt);
```

Sprawdź działanie programu

17. Komunikat do użytkownika o darmowej dostawie

Modyfikacja funkcji WyświetlDane

```
157     2 references  
158     private void WyświetlDane(decimal KosztTransportu, decimal LacznaWaga, bool gabaryt, bool darmowaDostawa = false)  
159     {  
160         lb_KosztTransportu.Text = $"{KosztTransportu} PLN";  
161         lb_Waga.Text = $"{LacznaWaga} kg";  
162         if (gabaryt) lb_Waga.Text += " - gabaryt";  
163         if (darmowaDostawa) lb_KosztTransportu.Text += " - darmowa dostawa";  
164     }  
165
```

```
bool darmowaDostawa = false;  
if(WartoscZamowienia > 1000)  
{  
    KosztTransportu = 0;  
    darmowaDostawa = true;  
}
```

```
WyświetlDane(KosztTransportu, LacznaWaga, gabaryt, darmowaDostawa);
```

Sprawdź działanie programu

18. Wnioski

Opisz swoje spostrzeżenia na temat algorytmu. Prosto było go zaimplementować?

Coś można było inaczej zrobić ?

Opisz doświadczenie ze środowiskiem Visual Studio – łatwo jest modyfikować programy?
