
OPIS SYTUACJI

Tworzysz program dla sklepu internetowego, który obsługuje różne rodzaje zamówień. Każde zamówienie:

- ma cenę podstawową,
- może mieć dodatkowe koszty lub rabaty,
- jest rozliczane w inny sposób.

Program główny nie może sprawdzać typu zamówienia za pomocą `if` lub `switch`. Ma korzystać z polimorfizmu, aby każde zamówienie samo wiedziało, jak obliczyć swoją cenę końcową.

CZĘŚĆ 1 – Klasa bazowa `Order`

1. Utwórz klasę `Order`.
2. Klasa powinna posiadać pola:
 - `double basePrice` – cena podstawowa zamówienia,
 - `String customerName` – imię klienta.
3. Dodaj konstruktor, który ustawia oba pola.

Dodaj metodę:

`double calculateFinalPrice()`

4. która domyślnie zwraca `basePrice`.

Dodaj metodę:

`void printSummary()`

5. która:
 - wypisuje imię klienta,
 - wypisuje nazwę klasy zamówienia,
 - wypisuje cenę końcową zamówienia (korzystając z `calculateFinalPrice()`).

👉 Metoda `printSummary()` nie może być nadpisywana w klasach potomnych.

CZĘŚĆ 2 – Zamówienie z dostawą do domu

1. Utwórz klasę `HomeDeliveryOrder`, która dziedziczy po `Order`.

2. Dodaj pole:
 - `double deliveryCost` – koszt dostawy.
3. Utwórz konstruktor, który ustawia:
 - cenę podstawową,
 - imię klienta,
 - koszt dostawy.

Nadpisz metodę `calculateFinalPrice()` tak, aby:

cena końcowa = `basePrice + deliveryCost`

4.

CZĘŚĆ 3 – Zamówienie z rabatem

1. Utwórz klasę `DiscountOrder`, która dziedziczy po `Order`.
2. Dodaj pole:
 - `double discountPercent` – rabat w procentach.
3. Utwórz konstruktor ustawiający wszystkie pola.

Nadpisz metodę `calculateFinalPrice()` tak, aby:

cena końcowa = `basePrice - (basePrice * discountPercent / 100)`

4.

CZĘŚĆ 4 – Zamówienie premium (dziedziczenie wielopoziomowe)

1. Utwórz klasę `PremiumOrder`, która dziedziczy po `HomeDeliveryOrder`.
2. Dodaj pole:
 - `double premiumFee` – dodatkowa opłata premium.
3. Utwórz konstruktor ustawiający wszystkie pola.

Nadpisz metodę `calculateFinalPrice()` tak, aby:

cena końcowa = `basePrice + deliveryCost + premiumFee`

4.

CZĘŚĆ 5 – Program główny i polimorfizm

1. Utwórz klasę `Main`.

2. W metodzie `main`:
 - utwórz co najmniej trzy różne zamówienia,
 - każde przypisz do zmiennej typu `Order`,
 - dla każdego wywołaj metodę `printSummary()`.

👉 W tej części nie wolno używać:

- `if`
 - `switch`
 - `instanceof`
-

PRZYKŁAD DZIAŁANIA PROGRAMU

Klient: Anna

Typ zamówienia: `HomeDeliveryOrder`

Cena końcowa: 120.0 zł

Klient: Bartek

Typ zamówienia: `DiscountOrder`

Cena końcowa: 90.0 zł

Klient: Celina

Typ zamówienia: `PremiumOrder`

Cena końcowa: 150.0 zł
