

Laboratorium Programowania Obiektowego w Javie
2014/2015

Głównym założeniem projektu jest zastosowanie poznanych mechanizmów programowania obiektowego. Program powinien być napisany *samodzielnie*.

Należy uwzględnić jak najwięcej technik:

1. dziedziczenie, polimorfizm
2. klasy abstrakcyjne, interfejsy
3. kolekcje
4. wyjątki
5. hermetyzacja (ukrycie tych elementów, dla których nie ma uzasadnienia, by były upublicznione), modyfikatory static, final itp.
6. przeciążanie i przesłanianie
7. różne konstruktory
8. metody powinny przyjmować odpowiednie parametry i zwracać odpowiednie typy danych
9. co najmniej 6 klas w sensownej hierarchii

Dla chętnych dodatkowo:

1. strumienie: wejście-wyjście, wczytywanie z pliku/zapisywanie do pliku
2. wątki
3. interfejs graficzny
4. inne pomysły

Przykładowa realizacja projektu: Sklep z owocami

- Klasy: Sklep, Koszyk, Cennik, Kasa, Torba oraz kilka klas reprezentujących produkty (np. Jabłka, Gruszki, Pomarańcze, Truskawki).
- W klasie testowej zakupy mogą być z góry ustalone (tzn. nie trzeba pobierać ich od użytkownika, z pliku itp.). Przykładowy wynik działania programu w konsoli:

Kowalski bierze koszyk nr 1

Zawartość "koszyk nr 1": Jabłka 2.0 kg, Gruszki 0.5 kg, Pomarańcze 1.5 kg

Kasa - rachunek za koszyk nr 1:

Jabłka $2.0 * 3.0 \text{ zł/kg} = 6.0 \text{ zł}$

Gruszki $0.5 * 5 \text{ zł/kg} = 2.5 \text{ zł}$

Pomarańcze $1.5 * 4 \text{ zł/kg} = 6.0 \text{ zł}$

Razem: 14.5 zł

Zawartość torby (właściciel: Kowalski): Jabłka 2.0 kg, Gruszki 0.5 kg, Pomarańcze 1.5 kg

- Dziedziczenie (np. klasa Jabłka dziedziczy z klasy Owoc), polimorfizm, klasy abstrakcyjne (np. klasa Owoce (zamiast posiadania pola nazwa_owocu) będzie klasą abstrakcyjną z abstrakcyjną metodą zwracającą rodzaj owoców; metodę należy potem przededefiniować w poszczególnych klasach owoców), interfejsy (np. klasa Owoce implementuje interfejs Produkt)
- Klasa Cennik zaprojektowana w taki sposób, aby istniał co najwyżej jeden obiekt tej klasy (bo cennik jest tylko jeden). Taka klasa to singleton.
- Zastosowanie kolekcji:
 - ◆ Przy ustalaniu cen można skorzystać z klasy Hashmap, która przechowuje pary (klucz, wartość) i umożliwia łatwe odszukanie informacji według kluczy.
 - ◆ Dodawanie do koszyka i torby można zrealizować, wykorzystując klasę ArrayList.
- Przechwycenie wyjątku (np. klient chce kupić produkt, którego nie ma w cenniku).
- Dodatkowo: pobieranie nazwisk klientów i ich zakupów od użytkownika, wczytywanie zakupów z pliku, klienci jako wątki itp.

TEMATY DO WYBORU:

1. Automat do kawy

W automacie powinny być dostępne różne rodzaje kaw z dodatkowymi opcjami (np. bez cukru, mleko sojowe). Każdy automat ma numer seryjny, zawiera informacje o dostępności kaw (niektóre mogą się skończyć), liczbie wydanych kaw itp. Automat powinien obsługiwać zapłatę w gotówce oraz kartami kredytowymi, wydawać resztę lub połykać pieniądze, itp.

2. Winda

Implementacja typowej windy: zamawianie windy, wybór piętra, zamykanie drzwi, dzwonek itp. Winda porusza się w pionie (np. 10 pięter i parter), ma ograniczoną liczbę miejsc, informację o ostatniej konserwacji itp.

3. System rezerwacji miejsc w samolotach

System dysponuje informacjami o dostępnych lotach (data, godzina, miasto startu i lądowania, liczba miejsc, cena lotu, typ samolotu, posiłek na pokładzie itp.) oraz umożliwia wyszukanie lotu z wolnymi miejscami (według miasta startu, daty itp.), dokonanie i odwołanie rezerwacji. W przypadku rezerwacji pasażer jest wpisywany na listę pasażerów; przy odwołaniu – usuwany z listy. System powinien posiadać opcję wypisania pasażerów danego lotu itp.

4. Podobnie można zaimplementować:

- **System obsługi kina** (rezerwacje, sprzedaż biletów, wyświetlanie aktualnego repertuaru po tytule, dacie, sali...)
- **System obsługi biblioteki** (rezerwacje, wypożyczenia, zwroty, dodawanie i edycja zbiorów, naliczanie kar...)
- **System obsługi komunikacji miejskiej** (dla pasażerów: wyszukiwanie połączeń po nr linii, przystankach, godzinie odjazdu; dla kierowców: odpytywanie o aktualne opóźnienie....)
- **Obsługa bankomatu** (różne karty, sprawdzenie PIN, wypłaty, wpłaty, sprawdzenie stanu konta, zakup kodów w telefonii pre-paid...)
-

5. Własne propozycje

TERMIN: do 13 czerwca 2015 , godz. 23:59.

Pliki należy umieścić w module *Foldery Zadań* na edux.pjwstk.edu.pl