

## Wzorce Projektowe

Przeanalizuj i zaprojektuj z wykorzystaniem diagramu klas rozwiązania poniższych zadań.

**Wprowadzenie.** Dana jest aplikacja typu PIM (ang Personal Information Manager). Aktualna wersja aplikacji pozwala na zarządzanie informacjami życia codziennego: adresami, spotkaniami, kalendarzem, etc. Aplikacja jest kombinacją książki adresowej, planu/harmonogramu zajęć oraz listy kontaktów wykorzystującej dane teleadresowe. Firma rozwijająca aplikację przymierza się do wydania kolejnej wersji programu.

**Zadanie 1.** Nowa wersja aplikacji powinna posiadać funkcję wielopoziomowego Cofnij/Ponów. Lista czynności do cofnięcia/ponowienia powinna być wspólna dla całej aplikacji. PIM może posiadać tylko jedną „instancję” historii czynności. Operacje, które można wykonać na liście czynności, to *void addCommand(String command)*, *Object undoCommand()* oraz *String toString()*.

**Zadanie 2.** Nowa wersja aplikacji powinna zostać rozbudowana o książkę adresową. Każdy kontakt w książce posiada przypisany adres (ulica, miasto, region, kod pocztowy) oraz numer telefonu (numer telefonu, numer kierunkowy, prefiks kraju). Dodatkowo, chcemy, aby w książce adresowej znalazła się informacja o nazwie kraju naszego kontaktu, a opcjonalnie o domyślnym separatorze. Wiemy, że obecne osoby, które są zapisane w obecnych kontaktach pochodzą ze USA (nazwa: "UNITED STATES", separator: „, ”, kod kraju ‘01’, długość numeru telefonu to 10 znaków) oraz z Francji (nazwa: "FRANCE", kod kraju ‘33’, długość numeru telefonu to 9 znaków). W najbliższej przyszłości rozważane są także Holandia oraz Dania. Operacje, które muszą być zaimplementowane to *getStreet()*, *getCity()*, *getPostalCode()*, *getRegion()*, *getCountry()*, *getFullAddress()*, *setStreet()*, *setCity()*, *setPostalCode()*, *setRegion()*.

**Zadanie 3.** Nowa wersja aplikacji powinna posiadać funkcję terminarza. Użytkownik powinien mieć możliwość zdefiniowania spotkania definiowanego przez datę początku i końca, opis spotkania, miejsce spotkania oraz listę uczestników spotkania. Należy zwrócić uwagę, że spotkania mogą mieć różny charakter i wymagają różnych informacji (wyjście do kina, konferencja, wyjście na wystawę). Operacje, które muszą być zaimplementowane to *getStartDate()*, *getEndDate()*, *getDescription()*, *getAttendees()*, *getLocation()*, *setStartDate(Date newStartDate)*, *setEndDate(Date newEndDate)*, *setDescription(String newDescription)*, *setAttendees(ArrayList newAttendees)*, *setLocation(Location newLocation)*, *addAttendee(Contact newAttendee)*, *removeAttendee(Contact oldAttendee)*. Należy przewidzieć możliwość definiowania nowych spotkań posiadających dodatkowe ograniczenia (np. konieczność zdefiniowania daty zakończenia).

**Zadanie 4.** Nowa wersja aplikacji powinna posiadać możliwość dostosowania jej do potrzeb użytkownika, to jest zmiany kroju pisma, jego koloru, rozmiaru, zestawu dostępnych usług, domyślnej waluty, czy formatu daty. Kreator powinien udostępniać metody *setNation(String name)*, *Nation getNation()*, *char getCurrencySymbol()*, *NumberFormat getNumberFormat()*,

*String getPhonePrefix()*. Klasa waluty udostępnia metody: *setCurrencySymbol(char newCS)*, *setNumberFormat(NumberFormat newNF)*, *char getCurrencySymbol()*, *NumberFormat getNumberFormat()*. Klasa telefonu powinna udostępniać metody: *String getInternationalPrefix()*, *String getAreaNumber()*, *String getNetNumber()*, *setInternationalPrefix(String newIP)*, *setAreaNumber(String newAN)*, *setNetNumber(String newNN)*.

**Zadanie 5.** Nowa wersja aplikacji powinna posiadać proste funkcje zarządzania przedsięwzięciami. Elementami każdego przedsięwzięcia są zadania i produkty. Należy przewidzieć możliwość pojawienia się elementów takich jak podprojekty czy zadania zależne. Dla elementu projektu powinna istnieć możliwość ustalenia szacowanego kosztu realizacji. Metoda ta powinna nosić nazwę *double getCostEstimate()* i wykorzystywać metody *double getTimeRequired()* oraz *double getMaterialCost()*.