

Celem ćwiczenia jest praktyczne wykorzystanie umiejętności definiowania klas oraz tworzenia obiektów w języku Java z uwzględnieniem techniki dziedziczenia. Zadanie zostanie wykonane na przykładzie aplikacji do gromadzenia i wyświetlania danych na temat tras pojazdów komunikacji miejskiej.

Opis modelowanego systemu i działania aplikacji

Dla każdego pojazdu komunikacji identyfikowanego za pomocą numeru zostaje zdefiniowana trasa składająca się z kolejnych przystanków. Każdy przystanek posiada nazwę, a także nazwę, odległość i czas dojazdu do kolejnego. Jeżeli na danym przystanku można przesiąść się na inne kierunki jazdy, są również gromadzone informacje o tych kierunkach. Aplikacja powinna uruchamiać się w oknie konsoli tekstowej. W ramach zadania należy zdefiniować dynamiczną tablicę obiektów zawierających wyszczególnione dane i kolejno wyprowadzać je na ekran np. po naciśnięciu spacji. Dane powinny być przypisywane w kodzie aplikacji, bez udziału użytkownika. W ramach zadania rozszerzonego należy udostępnić użytkownikowi programu możliwość wprowadzania informacji na temat liczby przystanków oraz danych każdego z nich.

Zadania do wykonania

□ Uruchomić środowisko programowania, zamknąć ewentualne otwarte projekty a następnie utworzyć nowy projekt aplikacji konsolowej o nazwie PROJ02.

Dodać do projektu klasę `CPrzystanek` z następującymi składnikami:

- `Nazwa`, `Nastepny` - składniki chronione, referencje do obiektów klasy `String`, domyślna wartość: `null`,
- `metrowDo`, `minDo` – składniki chronione typu `int`, domyślna wartość: zero,
- konstruktory: pierwszy z argumentami (`String`, `String`, `int`, `int`) przypisujący kolejno wartości do pól oraz drugi z argumentem (`String`) przypisujący tylko nazwę,
- nadpisana metoda `toString()` wyświetlająca informację o przystanku. Wskazówka: można użyć operatora warunkowego `?:` (patrz przykładowy kod poniżej).

□ Dodać do projektu klasę `CPrzystanekPrzes`, która dziedziczy z `CPrzystanek` oraz posiada następujące składniki:

- chronioną tablicę obiektów klasy `String` o nazwie `Przesiadki`,
- prywatną metodę `PrzypiszPrzesiadki` nie zwracającą wartości, posiadającą argument typu `varargs` klasy `String`. W metodzie należy sprawdzić, czy argument zawiera obiekty, jeżeli tak, to zarezerwować pamięć na tablicę `Przesiadki` i przepisać kolejno dane. Wskazówka: zaleca się wykorzystanie składni `foreach`,
- konstruktor domniemany – wywołujący konstruktor klasy bazowej z wartościami ("`-`", `null`, `0`, `0`) oraz przypisujący polu `Przesiadki` wartość `null`,
- dwa kolejne konstruktory z argumentami: (`String`, `String`, `int`, `int`, `String...`) oraz (`String`, `String...`), które wywołują konstruktor klasy bazowej a następnie metodę `PrzypiszPrzesiadki`. Uwaga 1: metoda `PrzypiszPrzesiadki` jest prywatna, stąd nie musi być deklarowana z modyfikatorem `final`. Uwaga 2: w drugim przypadku konstruktor powinien przypisać nazwę przystanku i listę przesiadek.
- nadpisana metoda `toString()` wyświetlająca informację o przystanku. Wskazówka: można wykorzystać metodę z klasy bazowej (patrz przykładowy kod poniżej), a następnie użyć składni `foreach` do wypisania nazw kierunków przesiadek.

□ Dodać do projektu klasę `CPojazd` z następującymi składnikami:

- Statyczny chroniony składnik `Przedsiębiorstwo` klasy `String` o wartości "MPK Krakow"
- Chroniony całkowity składnik `Numer` o wartości przypisywanej w konstruktorze
- Chroniona referencja do abstrakcyjnej klasy generycznej kontenera `List` o nazwie `Trasa` do przechowywania obiektów klasy `CPrzystanek` (patrz przykładowy kod poniżej)
- Publiczny konstruktor z argumentem `int`, przypisanie wartości polu `Numer` i inicjalizacja obiektu `Trasa` przy wykorzystaniu klasy generycznej `ArrayList`
- Czterokrotnie przeciążona publiczna metoda `dodajPrzystanek` nie zwracająca wartości, wywołująca odpowiedni konstruktor klasy `CPrzystanek` lub `CPrzystanekzPrzes` oraz dodająca referencję utworzonego obiektu do kontenera `Trasa`: - `CPrzystanek` z istniejącym kolejnym: konstruktor (`String`, `String`, `int`, `int`) - `CPrzystanek` bez kolejnego (ostatni): konstruktor (`String`) - `CPrzystanekzPrzes` z istniejącym kolejnym: konstruktor (`String`, `String`, `int`, `int`, `String`...) - `CPrzystanekzPrzes` bez kolejnego (ostatni): konstruktor (`String`, `String`...) Wskazówka: metoda dodająca element do kontenera `ArrayList` ma nazwę `add`
- nadpisana metoda `toString()` wyświetlająca informację o trasie pojazdu. Wskazówka: należy wypisać informacje z pól `Numer` i `Przedsiębiorstwo`, a następnie użyć składni `foreach` do wypisania danych kolejnych przystanków z kontenera `Trasa`.

Zadanie podstawowe

- zdefiniować klasy wg podanych wymagań i wskazówek;
- wewnątrz funkcji `main()` zdefiniować dwie referencje do obiektu klasy `CPojazd` o nazwach odpowiednio `p1` i `p2`, a następnie za pomocą operatora `new` utworzyć obiekty przypisując numery pojazdów 4 i 129.
- Wykorzystując odpowiednio różne wersje metody `dodajPrzystanek` należy zdefiniować trasy obu pojazdów w taki sposób, aby uzyskać efekt pokazany w punkcie 5.

Fragmenty kodu:

- Propozycja fragmentu kodu metody `toString()` klasy `CPrzystanek`:

```
return "Przystanek: "+Nazwa+" następny: "+((Nastepny==null)?"-brak-":Nastepny);
```

- Propozycja kodu metody `toString()` klasy `CPrzystanekzPrzes`:

```
String res = super.toString();
res += "\n przesiadki: ";
if(Przesiadki==null)res += "brak"; else for(String pp: Przesiadki)res += pp+", ";
return res;
```

- Definicja referencji do obiektu `Trasa` klasy generycznej `List` wewnątrz klasy `CPojazd`:

```
protected List<CPrzystanek>Trasa;
```

- Utworzenie obiektu `Trasa`: klasy `ArrayList` (potomka `List`):

```
Trasa = new ArrayList<>();
```

- Przypisanie listy przesiadek (fragment prywatnej metody `PrzypiszPrzesiadki(String...p)`)

```
Przesiadki = new String[p.length];
int i=0;
for(String pp: p)Przesiadki[i++]=pp;
```

```
Output - Lab02 (run) - Editor
Output - Lab02 (run) %
run:
NR: 4, właściciel: MPK Krakow, trasa:
  Przystanek: Dworzec
    następny: LOT, odległość: 300m, czas: 2min
    przesiadki: Bieżanów, Prokocim, Kurdwanów,
  Przystanek: LOT
    następny: Bagatela, odległość: 500m, czas: 3min
    przesiadki: Krowodrza Górka,
  Przystanek: Bagatela
    następny: Plac Invalidów, odległość: 800m, czas: 4min
  Przystanek: Plac Invalidów
    następny: Petla, odległość: 500m, czas: 3min
    przesiadki: Kleparz, Cichy kącik,
  Przystanek: Petla
    następny: -brak-

NR: 129, właściciel: MPK Krakow, trasa:
  Przystanek: Akademiki PK
    następny: Komorowskiego, odległość: 500m, czas: 3min
  Przystanek: Komorowskiego
    następny: Krokus, odległość: 600m, czas: 4min
    przesiadki: Mistrzejowice, Nowa Huta, Działek Czerwony,
  Przystanek: Zajezdnia Kleparz
    następny: -brak-
    przesiadki: Krowodrza, Bieżanów, Prokocim, Balice,

BUILD SUCCESSFUL (total time: 0 seconds)
```