

Laboratorium 8

1. Ćwiczenie 1 Elementy obiektowości 1

Zdefiniuj klasę **Auto** opisującą pojazd i jego właściciela zawierający takie dane jak:

- Pojazd
 - Typ pojazdu
 - Marka
 - Pojemność
 - Liczba miejsc
 - VIN
 - Nr rejestracyjny
 - Rok produkcji
 - Kolor
 - Polisa nr
 -
- Właściciel
 - Imię i nazwisko
 - Adres zamieszkania
 - PESEL
 - Nr prawa jazdy
 - Data uzyskania prawa jazdy
 - Rok zakupu
 - Liczba punktów karnych
 - ...

Powyższa klasa implementuje trzy interfejsy na potrzeby różnych instytucji:

- **ICepikData** - dostęp do pełnych danych o pojeździe i właścicielu (ale bez punktów karnych)
- **IStatData** – dostęp jedynie do danych potrzebnych do statystycznej analizy typów pojazdów, zwracających jedynie dane na temat pojazdu, bez danych właściciela
- **IPoliceData** – dostęp do pełnych danych.

Zdefiniuj trzy listy pozwalające na enumerację obiektów typu **ICepikData** lub **IStatData** lub **IPoliceData**.

W celu przetestowania utwórz kilka instancji klasy **Auto** do list i wypisz zawartość kolejnych.

2. Ćwiczenie 2 Przeciążanie operatorów I

Dla klasy **Point** (opracowanej w poprzednich laboratoriach) odpowiednio przeciąż operatory:

- Dwuargumentowy operator oddawania, tak aby operacja **Point** $p3 = p1 + p2$, na dwóch obiektach **Point** $p1$ i $p2$ tworzyła obiekt typu **Point** $p3$, zawierający sumę odpowiednich współrzędnych obu punktów.
- Jednoargumentowy operator **true** i **false**, zwracające:
 - **true** jeśli choć jedna współrzędna jest różna od 0
 - **false** w przeciwnym przypadku.
- Dwuargumentowy operator **==**, tak aby operacja $p1 == p2$, na dwóch obiektach **Point** $p1$ i $p2$ zwracała **true**, w przypadku równości odpowiednich współrzędnych obu punktów. Czy wystarczy zdefiniować tylko operator „==”?

- Dwuargumentowy operator `<` (`<=`), tak aby operacja `p1 < p2` (`p1 <= p2`), na dwóch obiektach **Point** `p1` i `p2` zwracała `true` w przypadku, gdy obie współrzędne w `p2` są większe (lub równe). Czy wystarczy zdefiniować tylko operator `<`?
- Jednoargumentowe operatory `++` i `--`, odpowiednio inkrementującą/dekrementującą każdą współrzędną w obiekcie typu **Point**.
- Zapewnij, aby poniższa operacja:
 - `Point p = 5;`
Spowodowała powstanie obiektu typu **Point**, zawierającego współrzędne (5,0).
- Zapewnij, aby poniższa operacja zwróciła sumę współrzędnych:
 - `Point p = new Point (2,5);`
 - `int suma = (int)p; //suma==3`
- Czy jest możliwe nadpisanie operatora `+=`?

3. Ćwiczenie 3 Delegaty

- Zdefiniuj delegata, który przyjmie funkcje typu `f(int, int)`.
- Zdefiniuj w klasie kilka funkcji, które te liczby sumują, odejmują i mnożą oraz dzielą dwie wprowadzone liczby i wypisują odpowiedni komunikat.
- Wprowadź je do delegata i uruchom go z odpowiednimi parametrami.