Laboratorium nr 7:

Wykorzystując informacje przedstawione na wykładzie opracuj kod realizujący następujące zadania:

1. **Dziedziczenie: Tworzenie Zwierząt** Zaimplementuj klasę bazową Zwierzę z metodą dzwiek(). Następnie stwórz klasy dziedziczące, takie jak Pies, Kot i Ptak, które nadpisują tę metodę, wytwarzając różne dźwięki.
2. **Polimorfizm: Operacje Matematyczne** Utwórz klasę Kalkulator z metodami dodaj(), odejmij(), pomnóż() i podziel(). Napisz program, który wykorzystuje polimorfizm, aby obsługiwać różne typy operacji matematycznych na różnych danych.
3. **Enkapsulacja: Zarządzanie Książką Adresową** Zdefiniuj klasę Osoba z prywatnymi atrybutami \_\_imie i \_\_nazwisko. Umożliw użytkownikowi dostęp do tych informacji tylko za pomocą publicznych metod pobierz\_imie() i pobierz\_nazwisko().
4. **Agregacja: Szkoła i Kursy** Stwórz klasę Kurs i klasę Szkola, gdzie szkoła posiada listę kursów. Wykorzystaj paradygmat agregacji, aby przechowywać kursy w ramach szkoły.
5. **Kompozycja: Samochód i Silnik** Zaprojektuj klasę Samochód, która zawiera obiekt klasy Silnik. Wykorzystaj kompozycję, aby modelować relację między samochodem a jego silnikiem.
6. **Polimorfizm: Figury Geometryczne** Utwórz klasę bazową Figura z metodą oblicz\_pole(). Następnie stwórz klasy dziedziczące: Kwadrat, Koło i Trójkąt, które nadpisują tę metodę.
7. **Dziedziczenie: Sprzęt Elektroniczny** Stwórz klasę bazową UrządzenieElektroniczne z metodami wlacz() i wylacz(). Utwórz klasy dziedziczące: Telewizor, Laptop i Smartfon, które dostosowują te metody.
8. **Abstrakcja: System Bankowy** Zdefiniuj klasę abstrakcyjną KontoBankowe z metodami wplac() i wyplac(). Utwórz klasy dziedziczące, takie jak KontoOszczędnościowe i KontoKredytowe, które implementują te metody.
9. **Przeciążanie Operatorów: Klasa Wektor** Stwórz klasę Wektor z metodami do dodawania i mnożenia wektorów. Wykorzystaj przeciążanie operatorów, aby umożliwić operacje na wektorach zapisane w formie matematycznych operatorów.
10. **Zdarzenia: System Powiadomień** Zaprojektuj klasę Powiadomienie z metodą wyslij(). Stwórz klasę SystemPowiadomien, która zarządza subskrybentami powiadomień i wysyła powiadomienia przy wystąpieniu zdarzenia.

Programowanie obiektowe w C++ (dla osób chętnych):

1. **Klasa Samochód z Danymi Prywatnymi:** Zdefiniuj klasę Samochod, która ma prywatne atrybuty takie jak marka, model i numerRejestracyjny. Użyj specyfikatora dostępu private do zabezpieczenia danych. Dostęp do tych danych powinien być możliwy tylko za pomocą publicznych metod UstawMarke(), PobierzMarke(), itp.
2. **Dziedziczenie z Dostępem Protected:** Utwórz klasę bazową Figura z chronionym atrybutem kolor. Użyj specyfikatora dostępu protected. Następnie stwórz klasę dziedziczącą, np. Kwadrat, która ma dostęp do atrybutu kolor i posiada publiczną metodę PobierzKolor().
3. **Klasa Konta Bankowego z Metodami Private:** Zaimplementuj klasę KontoBankowe, która ma prywatne atrybuty saldo i numerKonta. Użyj specyfikatorów dostępu private do zabezpieczenia danych. Umożliw użytkownikowi dostęp do salda tylko za pomocą publicznej metody PobierzSaldo(), ale dostęp do numeru konta powinien być ograniczony.
4. **Zmienne Protected w Dziedziczeniu:** Stwórz klasę bazową Osoba z chronionymi atrybutami imie i nazwisko. Użyj specyfikatora dostępu protected. Następnie utwórz klasę dziedziczącą, np. Student, która ma dostęp do tych zmiennych i publiczną metodę WyswietlDane().
5. **Klasa Zdarzeń z Metodami Protected:** Zaprojektuj klasę Zdarzenie z chronionymi metodami Rozpocznij() i Zakoncz(). Użyj specyfikatora dostępu protected. Utwórz klasę dziedziczącą, np. Spotkanie, która ma dostęp do tych metod i publiczną metodę PlanujSpotkanie().