Laboratorium nr 4:

Wykorzystując informacje przedstawione na wykładzie opracuj kod realizujący następujące zadania:

1. **Enkapsulacja: Konto Bankowe** Utwórz klasę KontoBankowe z prywatnym atrybutem \_\_stan i metodami wpłata oraz wypłata. Zapewnij, że stan konta nie może być bezpośrednio zmieniany spoza klasy.
2. **Dziedziczenie: Pojazdy** Stwórz klasę bazową Pojazd z atrybutem predkosc i metodą poruszaj\_sie(). Następnie utwórz dwie klasy dziedziczące: Samochod i Rower, które nadpisują metodę poruszaj\_sie().
3. **Polimorfizm: Zwierzęta** Zdefiniuj klasę bazową Zwierze z metodą wydaj\_dzwiek(). Następnie utwórz klasy dziedziczące: Pies, Kot i Krowa, które nadpisują tę metodę, wydając różne dźwięki.
4. **Enkapsulacja: Książka** Stwórz klasę Książka z prywatnym atrybutem \_\_tytuł. Umożliw użytkownikowi uzyskanie dostępu do tytułu za pomocą publicznej metody pobierz\_tytul().
5. **Dziedziczenie: Figury Geometryczne** Zaimplementuj klasę bazową Figura z metodą oblicz\_powierzchnię(). Utwórz klasy dziedziczące: Kwadrat i Koło, które nadpisują tę metodę.
6. **Polimorfizm: Operacje Matematyczne** Utwórz klasę OperacjaMatematyczna z metodą wykonaj\_operację(). Następnie utwórz klasy dziedziczące: Dodawanie, Odejmowanie, Mnożenie i Dzielenie, które implementują różne operacje matematyczne.
7. **Enkapsulacja: Pracownik** Zdefiniuj klasę Pracownik z prywatnym atrybutem \_\_pensja. Umożliw dostęp do pensji tylko do odczytu za pomocą publicznej metody pobierz\_pensję().
8. **Dziedziczenie: Pojazdy Lądowe** Utwórz klasę bazową PojazdLądowy z atrybutem ilość\_kół i metodą jeźdź(). Następnie utwórz klasy dziedziczące: Samochód i Motocykl, które nadpisują metodę jeźdź().
9. **Polimorfizm: Kalkulator** Stwórz klasę Kalkulator z metodami dodaj(), odejmij(), pomnóż() i podziel(). Umożliw użytkownikowi wykonywanie operacji matematycznych na różnych typach danych.
10. **Enkapsulacja: Student** Zaimplementuj klasę Student z prywatnym atrybutem \_\_oceny. Umożliw dodawanie ocen tylko za pomocą metody dodaj\_ocenę().
11. **Dziedziczenie: Gry Komputerowe** Stwórz klasę bazową Gra z atrybutem nazwa i metodą rozpocznij\_grę(). Następnie utwórz klasy dziedziczące: GraAkcji, GraRPG i SymulatorSamolotu.
12. **Polimorfizm: Zwierzęta** Zdefiniuj klasę Zwierzę z metodą jedz(). Utwórz klasy dziedziczące: Lew, Żyrafa, Słoń i Wąż, które implementują różne sposoby jedzenia.
13. **Enkapsulacja: Zegar** Utwórz klasę Zegar z prywatnymi atrybutami \_\_godzina i \_\_minuta. Zapewnij dostęp do aktualnego czasu za pomocą publicznej metody pobierz\_czas().
14. **Dziedziczenie: Kształty** Stwórz klasę bazową Kształt z metodą oblicz\_pole(). Utwórz klasy dziedziczące: Kwadrat, Koło i Trójkąt, które nadpisują tę metodę.
15. **Polimorfizm: Gry Planszowe** Zaimplementuj klasę GraPlanszowa z metodą rozpocznij\_grę(). Utwórz klasy dziedziczące: Szachy, Monopoly i Chińczyk, które implementują różne mechaniki gry.
16. **Enkapsulacja: Baza Danych** Utwórz klasę BazaDanych z prywatnym atrybutem \_\_dane. Umożliw dostęp do danych tylko za pomocą publicznych metod dodaj\_wpis() i pobierz\_wpis().
17. **Dziedziczenie: Zwierzęta Morskie** Zdefiniuj klasę bazową ZwierzęMorskie z metodą plyń(). Następnie utwórz klasy dziedziczące: Ryba, Delfin i Rekin.
18. **Polimorfizm: Zadania** Napisz klasę Zadanie z metodą wykonaj(). Utwórz klasy dziedziczące: ZadanieMatematyczne, ZadanieProgramistyczne i ZadanieArtystyczne, które implementują różne rodzaje zadań.
19. **Enkapsulacja: Kawiarnia** Stwórz klasę Kawiarnia z prywatnym atrybutem \_\_menu. Umożliw dostęp do menu tylko za pomocą publicznej metody pobierz\_menu().
20. **Dziedziczenie: Instrumenty Muzyczne** Utwórz klasę bazową Instrument z atrybutem nazwa i metodą graj(). Następnie utwórz klasy dziedziczące: Gitara, Saksofon i Perkusja.