1. Stwórz klasę Osoba.
2. Dodaj niej prywatne pola: imie, nazwisko, rok urodzenia, dodaj domyślny konstruktor i parametryczny, metodą WypiszInfo() wyświetlającą wartości obiektu.
3. Stwórz klasę potomną Student z dodatkowymi polami rok, numerGrupy, numerAlbumu, stwórz konstruktury i metodę WypiszInfo().
4. Stwórz obiekty z klasy bazowej i potomnej. Za pomocą debuggera sprawdź jakie wartości na prywatnych polach mają obiekty.
5. Ustaw modyfikatory dostępu w klasie "rodzica" na protected.
6. W klasie Osoba dodaj metodę ObliczWiek(). Następnie stwórz obiekt z klasy Student i spróbuj na nim wywołać metodę ObliczkWiek().
7. W klasie Osoba dodaj pole miejsceZamieszkania z modyfikatorem private. Spróbuj je wywołać dla obiektu z klasy Student. Popraw kod aby wszystko działało.
8. Stwórz klasę potomną StudentPierwszegoRoku dziedziczoną z klasy Student i Osoba. Czy to możliwe? Co w C# możemy jedynie zrobić?
9. Stwórz obiekt typu Osoba z klasy Student (np. Osoba student2 = new Student()). Wywołaj dla niego metodę WypiszInfo().
10. Stwórz nowy obiekt student3 typu Student. Następnie stwórz obiekt osoba3 typu Osoba i podstaw za niego student3. Następnie stwórz obiekt student4 typu Student i podstaw za niego osoba3.
    Student student3 = new Student();
    Osoba osoba3 = student3;
    Student student4 = osoba3;
    Dla obiektu osoba3 wywołaj metodę WypiszInfo(). Gdzie jest błąd?
11. Stwórz program z klasą Konto. Dodaj w niej samodzielnie wybrane pola, metody, konstruktory, opisujące możliwe działania na koncie (saldo początkowe, końcowe, przelew, wpłata/wypłata w kasie, wypłata w bankomacie, płatność kartą, itp).
    a. Stwórz kilka obiektów, wyciągi dla kont wyświetl na ekranie.
    b. Stwórz klasy potomne KontoPrywatne (rozbuduj o przelew wynagrodzenia, otrzymanie 500+, itp) i KontoFirmowe (przelewy do ZUS, US).

Dodatkowe:

Zaprojektuj i zaimplementuj podstawy gry RPG. W grze może istnieć kilka typów bohaterów, lecz na początku będą tylko dwa rodzaje – wojownik i mag. W przyszłości planowana jest rozbudowa.

a. Wiadomo że wszystkie postacie będą opisane imieniem, poziomem życia, oraz posiadać będą operację która pozwoli obliczyć moc ataku.

b. Zaimplementuj wojownika i maga zgodnie ze schematem:

Mag: imię, żywotność (w %), zręczność (liczba całkowita), punkty taktyki (liczba całkowita)
Wojownik: imię, żywotność(w %), siła (liczba całkowita), punkty taktyki (liczba całkowita)

c. Zadbaj o hermetyzację w klasach.

d. Konstruktor domyślny powinien implementować bohaterów zgodnie ze schematem:
Mag: Imię=”ElfA”, żywotność=100%, zręczność=15, PT = 3;
Wojownik: Imię=”OrkA”, żywotność=100%, siła=15,PT=1;
e. Pozostałe metody:
i. zmiana pkt życia (nie mniej niż 0% i nie więcej niż 100%),

ii. moc ataku (zręczność/siła * PT * żywotność) UWAGA: w przypadku wojownika gdy żywotność spada poniżej 20% wpada w szał i mnożnik żywotności zmienia się na stałe 150%

iii. przeciąż metodę toString() tak aby wyświetlała informacje o bohaterze
f. Stwórz program w którym stworzysz 2 osobową drużynę bohaterów: Legolas i Aragorn. Pobaw się ich żywotnością i sprawdź jak zmienia się wartość ataku.

Pytania: Czy pomyślałeś o klasie rodzica np.bohater? Czy mag i wojownik dziedziczą po bohaterze? Czy ciężko byłoby dodać kolejnych bohaterów? Czy możliwe byłoby stworzenie klasy DrużynaBohaterów i czy miałoby to sens?