Programowanie obiektowe w C# - zadania:

Zajęcia laboratoryjne przeprowadzane są na komputerach z systemem operacyjnym Windows 7 z wykorzystaniem oprogramowania Visual Studio 2010 w wersji Ultimate. Poniżej omówiono kilka często wykorzystywanych skrótów:
VS – Visual Studio
LPM – Lewy przycisk myszy
PPM – Prawy przycisk myszy

Zadanie 1a: Stwórz klasę `Licz` z:

- publicznym polem `wartosc` przechowującym wartość liczbową.
- metodą `Dodaj` przyjmującą jeden parametr i dodającą przekazaną wartość do wartości trzymanej w polu `wartosc`.
- analogiczną operację `odejmij`

W `Main` utwórz kilka obiektów klasy `Licz` i wykonaj różne operacje.

Zadanie 1b: Do klasy `Licz` dodaj konstruktor z jednym parametrem - który inicjuje pole `wartosc` na liczbę przekazaną w parametrze.

Zadanie 1c: Zmień widoczność pola na `private` i dodaj funkcję wypisującą stan obiektu (pole `wartosc`)

Zadanie 2a: Stwórz klasę `Sumator` z:

- publicznym polem `Liczby` będącym tablicą liczb
- metodą `Suma` zwracającą sumę liczb z pola `Liczby`
- metodę `SumaPodziel3` zwracającą sumę liczb z tablicy, które są podzielne przez 3

Zadanie 2b: Zmień widoczność pola `Liczby` na `private` oraz dodaj konstruktor.

Zadanie 2c: Dodaj metodę: `int IleElementów ()` zwracającej liczbę elementów na w tablicy

Zadanie 2d: Dodaj metodę wypisującą wszystkie elementy tablicy

Zadanie 2e: Dodaj metodę przyjmującą dwa parametry: `lowIndex` oraz `highIndex`, która wypisze elementy o indeksach `>= lowIndex` oraz `<= highIndex`. Metoda powinna zadziałać poprawnie, gdy `lowIndex` lub `highIndex` wykraczają poza zakres tablicy (pominąć te elementy).

Zadanie 3: Stwórz klasę o nazwie Statyczna z:
a) polem int `I`
b) statycznym polem `Liczba`
c) statyczną metodą `Zwieksz()` - zwiekszajaca pole `Liczba` o 1
d) konstruktorem ustawiającym pole I oraz wykonujacym metodę `Zwieksz`

Czy w `Main` można zwiększać `Liczba` wykonując `Statyczna.Zwieksz()`? Jeżeli tak to jaki będzie efekt?

Czy z metody `Statyczna.Zwieksz` można odwołać się do pola `I`?
Uzasadnij.

Zadanie 4a: Zdefiniuj klasę opisującą datę. Zastanów się nad wyborem wewnętrznej reprezentacji dat. Zdefiniuj metody pozwalające na odczytywanie bieżącej daty i przestawianie jej o jeden tydzień w przód i w tył. Zadbaj o dobranie odpowiednich modyfikatorów dostępu do składowych.

Zadanie 4b: Rozwiń klasę z poprzedniego zadania aby jako parametr podawać ciąg znaków decydujący o formacie wyświetlanej daty.

Zadanie 5: Zdefiniuj klasę Liczba, która przechowuje w tablicy cyfry liczby dziesiętnej. Zdefiniuj operacje wypisywania liczby, nadawania jej wartości (w postaci parametru konstruktora będącego napisem) oraz mnożenia przez liczbę typu int. W przypadku gdy w czasie mnożenia okaże się, że tablica jest za mała, procedura mnożąca powinna kopiować jej zawartość do większej. Zdefiniuj wreszcie metodę silnia, która policzy silnię zadanej jako parametr liczby typu int.

Zadanie 6a: Stwórz klasy:

- Osoba z polami: imie, nazwisko, wiek, konstruktorem inicjującym wszystkie pola oraz metodą Wypisz.
- Książka z polami: tytul, autor (typu Osoba), data wydania oraz metodą Wypisz

Utwórz różne obiekty stworzonych klas. Wykonaj metody Wypisz.

Zadanie 6b: Stwórz klasę Czytelnik, dziedziczącą z klasy Osoba. Dodatkowo klasa Czytelnik powinna posiadać pole – listę / tablicę obiektów typu Książka - listę książek przeczytanych przez danego czytelnika oraz metodę WypiszKsiążki - wypisujące tytuły książek, które czytelnik przeczytał.
Stwórz 3-5 książek, 2-4 czytelników, przypisz książki do tablic / list przeczytanych książek czytelników, wykonaj metody WypiszKsiążki.

Zadanie 6c: Dodaj do czytelnika metodę Wypisz, która oprócz wypisania danych czytelnika (tych samych które wypisuje Osoba.Wypisz ()) wypisze także listę książek przez danego czytelnika przeczytanych (skorzystaj z już istniejącej metody WypiszKsiążki)

Zadanie 6d: Metody Wypisz() w klasach Osoba i Czytelnik poprzedź odpowiednimi słowami kluczowymi, aby wykonanie kodu:

```
Osoba o = new Czytelnik (...);
o.Wypisz();
```

spowodowało wyoknanie metody Wypisz () z klasy Czytelnik.

Zadanie 6e: Zmień widoczność pól w klasie Osoba na pola prywatne. Jeżeli trzeba popraw metodę Czytelnik.Wypisz (), aby jej rezultat nie zmienił się. Np. wykorzystaj właściwości.

Zadanie 6f: Utwórz klasę Recenzent dziedziczącą z klasy Czytelnik. Wypisz () recenzenta powinno wypisać listę książek, które przeczytał, a obok każdej pozycji losową ocenę (różną dla każdego wykonania metody Wypisz ()).
Czy do stworzenia takiej funkcjonalności konieczne jest aby lista książek w klasie Czytelnik była

protected? Czy też może posiadać widoczność private?

Utwórz 2 recenzentów, przypisz im książki i wykonaj stworzoną metodę.

Zadanie 6g: W Main stwórz listę obiektów klasy Osoba (List<Osoba>) dodaj do niej zarówno Czytelników jak i Recenzentów. W pętli wykonaj metodę Wypisz na wszystkich obiektach z listy.

Zadanie 6h: Zmień widoczność pól w klasie Książka na protected. W razie konieczności dodaj konstruktor i popraw istniejące klasy, aby program dalej działał poprawnie.

Zadanie 6i: Stwórz klasy KsiążkaPrzygodowa oraz KsiążkaDokumentalna. Do każdej z nich dodaj dodatkowe pole(pola).

Zadanie 6j: Metodę Wypisz () klasy Książka zmień na virtual. Zaimplemetuj metodę Wypisz w klasach z 6i aby wypisywały dodane pola.
Do list książek czytelników dodaj obiekty nowych klas i wykonaj WypiszKsiążki. Dla książek przygodowych oraz dokumentalnych powinny pojawić się opisy rozszerzone (o pola dodane w 6i).

Zadanie 7: Napisz program składający się z dwu klas, umieszczonych w dwu plikach (ale w jednym pakiecie). Jeden plik ma zawierać definicję klasy Osoba (z imieniem i nazwiskiem oraz metodami pozwalającymi na odczytywanie i zapisywanie tych atrybutów oraz konstruktorem), drugi ma wczytać od użytkownika dane 10 osób, zapamiętać je w tablicy osób, a następnie wypisać w odwrotnej kolejności.

Wersja trudniejsza: program czyta dane, aż użytkownik poda puste nazwisko. Tablica początkowo ma np. 10 elementów, ale w trakcie działania programu należy ją wymieniać w miarę potrzeby na większą (za każdym razem dwukrotnie zwiększając jej pojemność). Oczywiście nie można przy tym zgubić dotychczasowej zawartości.

Zadanie 8: Zamodeluj obiektowo, fragment rzeczywistości jakim jest rodzina (członkowie rodziny, zawody, samochód, hobby, edukacja itd.). Następnie zaimplementuj klasę filtr, która pozwoli wypisywać odpowiednie dane ze zbioru wszystkich zdefiniowanych klas.

Zadanie 9: Stworzyć program konsolowy zawierający trzy klasy:

A
B : A
C : B

Klasa A niech posiada dwie metody, jedną virtual, drugą nie.
Natomiast każda z klas pochodnych niech dostarcza ich własne implementacje.
Utworzyć po 2-3 obiekty każdej klasy.
Utworzyć trzy listy:

List<A>
List<B>
List<C>

a) Które obiekty mozna umieścić na których listach?
Umieścić obiekty na wszystkich listach na których się da.

b) Przeiterować przy pomocy for lub foreach po każdej liście (trzy pętle)
i wykonać obie metody (czy można zawsze obie wykonać?)
Jakie są wyniki? Dlaczego takie?

c) dodać do ciała metody z klasy C:
base.Metoda()

Jaki ma to wpływ na wyniki?

Zadanie 10: Utworzyć trzy klasy A, B, C, nie dziedziczące miedzy sobą, z metodami o takiej samej nazwie. Utworzyć ich obiekty. Czy można przypisywać obiekty klasy A do zmiennych typu B i C? Czy inne kombinacje są możliwe? Jeśli nie to jakie zależności muszą istnieć pomiędzy typami by było to możliwe?

Zadanie 11: Utworzyć klasę A z polem X.

```
A a = new A ();
B b = a;
b.X = 3;
```

Czy zmiana wartości pola X będzie widoczna przez odwołanie a.X? Dlaczego?
A czy jeżeli zmodyfikujemy a.X to b.X będzie odzwierciedlać tą zmianę? Dlaczego?

Zadanie 12: Utworzyć klasę Koło. Utworzyć klasę Samochód, która posiada pola typu Koło (np. leweprzednie, prawietylnie, itd). Utworzyć obiekty klasy Samochód. Zobaczyć w debuggerze strukturę obiektów.