

Rząd 2. Dział Płac /Firma

Pamiętaj: podstawą do rozpoczęcia sprawdzania jest poprawna kompilacja programu i przesłanie go na Githuba.

0. Stwórz projekt aplikacja konsolowa, zapisz go w folderze na Pulpicie, nazwij go Poprawa+swój numer indeksu np. poprawa130000.

1. Stwórz abstrakcyjną klasę *Osoba*. W tej klasie wykonaj czynności:

- a) dodaj pola imie (string), nazwisko (string), podstawa (int) – mod. protected
- b) dodaj wirtualną metodę bez parametru *ObliczPensje()* o typie zwracanym int i zwracającą wartość pola podstawa
- c) dodaj konstruktor z parametrami (string, string, int), który ustawi parametry jako odpowiednie pola w klasie

2. Stwórz klasę *Pracownik* dziedziczącą z klasy *Osoba*. W nowoutworzonej klasie wykonaj czynności:

- a) dodaj prywatne pole staz typu int
- b) przesłoń metodę *ObliczPensje()* następująco: do wartości zwróconej poprzez metodę *ObliczPensje()* z klasy bazowej dodaj wartość $staz * 100$,
- c) dodaj konstruktor z parametrami (string, string, int, int), pobrane parametry należy ustawić jako odpowiednie pola w klasie
- d) przesłoń metodę *ToString()* aby zwracała informacje o pracowniku np.

Pracownik: Jan Kowalski, podstawa: 3000, staż: 4.

3. Stwórz klasę *Kierownik* dziedziczącą z klasy *Osoba*. W nowoutworzonej klasie wykonaj czynności:

- a) dodaj prywatne pole *iloscPodwladnych* typu int
- b) przesłoń metodę *ObliczPensje()* następująco: do wartości zwróconej poprzez metodę *ObliczPensje()* z klasy bazowej dodaj wartość $iloscPodwladnych * 7$,
- c) dodaj konstruktor z parametrami (string, string, int, int), pobrane parametry należy ustawić jako odpowiednie pola w klasie
- d) przesłoń metodę *ToString()* aby zwracała informacje o kierowniku np.

Kierownik: Jan Kowalski, podstawa: 4000, ilość podwładnych: 24.

4. W klasie Program i metodzie Main wykonaj czynności:

a) stwórz listę pracownicy na obiekty typu Osoba

b) dodaj na listę 10 różnych obiektów

c) dla obiektów o indeksie nieparzystym wywołaj metodę ObliczCene(),
zwrócone wartości wypisz na konsoli

d) usuń z listy trzeci i siódmy elementy z listy (numerację liczymy od zera)

e) wypisz na konsoli aktualną zawartość listy

5. Stwórz interfejs INadgodziny i dodaj w nim deklarację metody
DoliczNadgodziny z parametrem typu int zwracającą typ int.

6. Podepnij interfejs INadgodziny do klasy Pracownik i Kierownik tak, aby
metoda DoliczNadgodziny(int parametr) zwracała następujące wartości:

- w klasie Pracownik do wartość z metody ObliczPensje() trzeba dodać
parametr*15;

- w klasie Kierownik do wartość z metody ObliczPensje() trzeba dodać
parametr*25;

7. Dla elementów na liście pracownicy z punktu 4 wywołaj metodę
DoliczNadgodziny. Parametr ma pokrywać się z indeksem elementu na liście.
Zwrócone wartości wypisz na konsoli.

8. Do klasy Osoba podepnij interfejs IComparable<Osoba> tak, aby sortowanie
odbywało się alfabetycznie po nazwisku (A->Z).

9. Posortuj listę pracownicy i po sortowaniu wypisz listę na konsoli.

10. Do klasy Pracownik dodaj mechanizm płytkiej kopii i skopiuj jeden z
obiektów typu Pracownik z listy pracownicy.

Punktacja

Polecenia 1-4 – łącznie 4 punktów

Polecenia 5-7 – łącznie 3 punkty

Polecenia 8-9 – łącznie 2 punkt

Polecenie 10 – 1 punkt

Aby otrzymać ocenę dostateczną na koniec, trzeba uzyskać min. 6 punktów.