

Telefon – aplikacja do bilingu.

1(2pkt). Stwórz abstrakcyjną klasę Usługa.

- a) dodaj w niej pole czas typu string oraz cena typu double – oba z modyfikatorem protected;
- b) dodaj deklarację abstrakcyjnej metody ObliczCene() typu void, metoda ma być bez parametru,
- c) dodaj konstruktor parametryczny z jednym parametrem typu string, parametr ma być ustawiony jako pole czas

2(2pkt). Dodaj klasę Polaczenie dziedziczącą z klasy Usługa.

- a) w klasie potomnej dodaj prywatne pola numer typu string oraz i czasP typu int,
- b) w klasie Polaczenie dodaj implementację metody ObliczCene() tak, aby ustawiała za pole cena wartość obliczoną następującą:  $\text{czasP} * 0,25$
- c) w klasie Polaczenie dodaj konstruktor parametryczny z trzema parametrami (kolejno typy string, string, int), w konstruktorze pobrane parametry należy ustawić jako odpowiednia pola czas, numer, czasP; ponadto w konstruktorze należy wywołać metodę ObliczCena(),
- d) przesłoń metodę ToString(), aby zwracała informacje o połączeniu np.

Połączenie: numer 1234, data i godzina rozmowy: 9.01.2017 13:00, długość trwania:2, łączny koszt: 0,58.

3(2pkt). Stwórz klasę Sms dziedziczącą z klasy Usługa.

- a) w klasie potomnej dodaj prywatne pole numer typu string,
- b) w klasie Sms dodaj implementację metody ObliczCene() tak, aby podstawiła za pole cena wartość 0,12
- c) w klasie Sms dodaj konstruktor parametryczny z dwoma parametrami (kolejne typy string, string), pierwszy parametr ma być ustawione jako pole czas, drugi jako pole numer; w konstruktorze należy także wywołać metodę ObliczCene()
- d) przesłoń metodę ToString(), aby zwracała informacje o sms np.

Sms: numer 1234, data i godzina smsa: 9.01.2017 13:00, łączny koszt: 0,15.

4(2pkt). Stwórz klasę Internet dziedziczącą z klasy Uslua.

a) w klasie potomnej dodaj prywatne pole iloscMB typu int,

b) w klasie Internet dodaj implementacje metody ObliczCene() tak, aby podstawiała za pole cena wartość obliczoną następująco: wartość pola iloscMB dzielimy na 365 i wynik jest zaokrąglony do dwóch miejsc po przecinku,

c) w klasie Internet dodaj konstruktor parametryczny z dwoma parametrami (kolejne typy string, int), pierwszy parametr ma być ustawione jako pole czas, drugi jako pole iloscMB; w konstruktorze należy także wywołać metodę ObliczCene()

d) przesłoń metodę ToString(), aby zwracała informacje o internecie np.

Internet, data i godzina internetu: 9.01.2017 13:00, ilośćMB: 1512, łączny koszt: 2.

5(2pkt). W klasie Program i instrukcji Main wykonaj następujące czynności:

a) stwórz listę o nazwie biling na obiekty typu Usługa

b) za pomocą konstruktorów parametrycznych dodaj na listę po 3 elementy typu Polaczenie, Sms, Internet

c) wypisz na konsoli elementy listy biling o indeksie nieparzystym (1,3,5...)

d) oblicz sumę cen wszystkich elementów z listy biling i wyświetl sumę na konsoli

6(2pkt) W projekcie wykonaj czynności:

a) dodaj interfejs IRabat, dodaj w nim deklarację metody Rabat(), bez parametru, typu void

b) do klas Polaczenie, Sms, Internet podepnij interfejs IRabat i zaimplementuj następująco metody

- w klasie Połączenie metoda obniża cenę o 10% (za cena jest podstawiana jest wartość o 10% mniejsza)

- w klasie Sms metoda obniża cenę o 20% (za cena jest podstawiana jest wartość o 20% mniejsza)

- w klasie Internet metoda obniża cenę o 30% (za cena jest podstawiana jest wartość o 30% mniejsza)

c) dla wszystkich elementów z listy biling wywołaj metodę Rabat(), a potem wypisz elementy z listy na konsoli (używając metody ToString()).

7(2pkt). Stwórz klasę Abonent. W niej wykonaj czynności:

a) dodaj w klasie właściwości Imie, Nazwisko, Numer (odpowiednio typów string, string, int) – pamiętaj o hermetyzacji

b) dodaj konstruktor parametryczny ustawiający wszystkie pola w klasie

c) przesłoń metodę ToString() np.

Abonent, imię i nazwisko: Jan Kowalski, numer 12346556

8(4pkt). Do klasy abonent podepnij dwa interfejsy

a) IComparable<T> i dodaj implementację metody CompareTo tak, aby sortowała alfabetycznie po nazwisku, potem po imieniu.

b) ICloneable i zaimplementuj mechanizm płytkiej kopii

9(2pkt). W klasie Program i instrukcji Main wykonaj czynności:

a) oddziel linijką komentarza powstały kod

b) stwórz generyczny listę abonenci i dodaj na nią co najmniej 5 obiektów typu Abonent

c) w dowolny sposób zrób tak, aby każdy abonent miał ustawione pole imie, nazwisko, numer

d) posortuj elementy na liście abonenci i wypisz je na konsoli

e) skopiuj trzeci element na liście za pomocą płytkiej kopii i dodaj go na koniec listy

Max – 20 pkt.

bardzo dobry (5,0) - 91-100%;

dobry plus (4,5) - 86-90%;

dobry (4,0) - 75-85%;

dostateczny plus (3,5) - 70-74%;

dostateczny (3,0) - 50-69%;

niedostateczny (2,0) - 0-49%.