

## Pizzeria

1(2pkt). Stwórz klasę Skladnik.

a) dodaj w niej prywatnej pola nazwaSkladnika oraz cenaSkladnika (typ double).

b) dodaj konstruktor parametryczny ustawiający jednocześnie nazwę i cenę składnika (kolejność dowolna, ale dalej trzeba być konsekwentnym).

c) przesłoń metodę ToString(), aby zwracała informację o składniku np.  
Nazwa: Szynka, cena: 2,50.

d) dodaj metodę zwracającą cenę składnika

2(2pkt). Do klasy Skladnik podepnij interfejs IComparable<T> lub IComparable oraz zaimplementuj metodę CompareTo tak, aby sortowała składniki od najtańszego do najdroższego. Przy równości ma być sortowanie po nazwie alfabetycznie (A-Z).

3(2pkt). W klasie Program i instrukcji Main wykonaj następujące czynności:

a) stwórz listę o nazwie pizza przechowującą obiekty typu Skladnik

b) za pomocą konstruktora parametrycznego dodaj na listę co najmniej 5 różnych elementów

c) wypisz na konsoli posortowane elementy, następnie odwróć kolejność na liście

d) oblicz sumę cen składników z listy i wypisz wartość sumy na konsoli

4(2pkt). Dodaj abstrakcyjną klasę Zamowienie.

a) dodaj w niej pole czasDostawy typu string z modyfikatorem protected (przechowujemy tu słowny opis daty, np. dziś po południu, na jutro itp.)

b) dodaj w tej klasie wirtualną metodę PoprawnyCzas bez parametru, zwracającą typ bool. Ma ona zwracać true w sytuacji kiedy czasDostawy jest niepusty (różny od null) i różny od stringa „wczoraj”. False ma być zwracane w przeciwnym wypadku.

c) dodaj zwykłą metodę typu void UstawCzasDostawy z parametrem typu string która ustawia parametr jako pole czasDostawy.

5(2pkt). Dodaj klasy potomne dziedziczące z klasy Zamowienie:

a) klasę NaMiejscu (można w środku zostawić ją pustą)

b) klasę NaWynos, w niej przesłoń metodę PoprawnyCzas tak, aby true było zwracane w sytuacji jeśli czasDostawy jest równy stringowi „dziś”, false w przeciwnym wypadku.

6(2pkt) W klasie Program i instrukcji Main wykonaj czynności:

a) oddziel linijką komentarza kod z polecenie nr 3

b) stwórz generyczną kolejkę o nazwie zamówienia przechowującą obiekty typu Zamowienie

c) dodaj na kolejkę co najmniej 5 obiektów z klas potomnych do klasy Zamowienie, w dowolny sposób ustaw dla nich jakiś czas

d) dla wszystkich elementów z kolejki wywołaj metodę PoprawnyCzas(), wartości zwracane wyświetl na konsoli

e) za pomocą metody dostępnej tylko dla kolejek usuń wszystkie elementy z kolejki

7(2pkt) W całym projekcie wykonaj czynności:

a) wyrzucić wyjątek przy próbie stworzenie obiektu typu Składnik z ceną będą wartością liczbą ujemną

b) obsłużyć wyjątek za pomocą bloku try...catch, tak aby w przypadku wyjątku wyświetlał komunikat na konsoli

c) dodaj ograniczenie górne na cenę (ma być mniejsza niż 7)

8(2pkt) Stwórz klasę Sos. W tej klasie dodaj następujące czynności:

a) dodaj w niej właściwości NazwaSosu (string) i CenaSosu(double)

b) dodaj konstruktor parametryczny z parametrami typu string, double, pobrane parametry mają być ustawione jako odpowiednia pola w tej klasie

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

a) oddziel linijką komentarza wcześniejszy kod

b) stwórz kolekcję ArrayList o nazwie pizza2

c) na kolekcję pizza2 dodaj co najmniej po 3 obiekty typu Skladnik i 3 obiekty typu Sos

d) wypisz na konsoli informacja o elementach kolekcji pizza2 np.

Składnik: szynka, cena:2

Sos: pomidorowy, cena 3

e) oblicz sumę cen elementów z kolekcji pizza2 i wypisz ją na konsoli

10(2pkt) Wykonaj następujące czynności:

a) do klasy Składnik podepnij interfejs ICloneable i zaimplementuj mechanizm zwany płytką kopią

c) z kolekcji pizza2 skopiuj jeden obiekt typu Składnik a następnie dodaj go na koniec tej kolekcji

d) usuń wszystkie elementy z kolekcji pizza2

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%.