F#: Napisz funkcję, która na podstawie określonej liczby rzutów trzema kostkami obliczy prawdopodobieństwo wyrzucenia kombinacji 1,2,3. Liczbę powtórzeń podaj jako parametr funkcji

F#: Zdefiniuj nowy typ Lista reprezentujący listę łączoną mogącą przechowywać wartości dowolnego typu. Następnie napisz funkcję, która będzie przyjmowała dwie takie listy liczb całkowitych i obliczała sumę odpowiadających sobie liczb. Przykładowo dla list (1,2,3,4); (2,3,4,5) funkcja powinna zwrócić listę (3,5,7,9). Napisz również kod, który demonstruje działanie przygotowanej funkcji.

C#: Zdefiniuj w C# klasę reprezentującą drzewo binarne. Następnie napisz metodę, która policzy sumę wartości przechowywanych w węzłach, spełniających warunek określony w funkcji przekazanej jako parametr. Zademonstruj działanie swojego kodu.

C#: Stwórz klasy na podstawie przykładów:







F#: Napisz funkcję, która sprawdzi czy dany punkt znajduje się wewnątrz prostokąta. Określ odpowiednie typy do reprezentowania punktu oraz prostokąta.

F#: Napisz funkcję, która będzie obliczała pole i obwód określonego prostokąta. Do zamodelowania parametru i wyniku tej funkcji wykorzystaj najlepszy wg siebie typ danych.

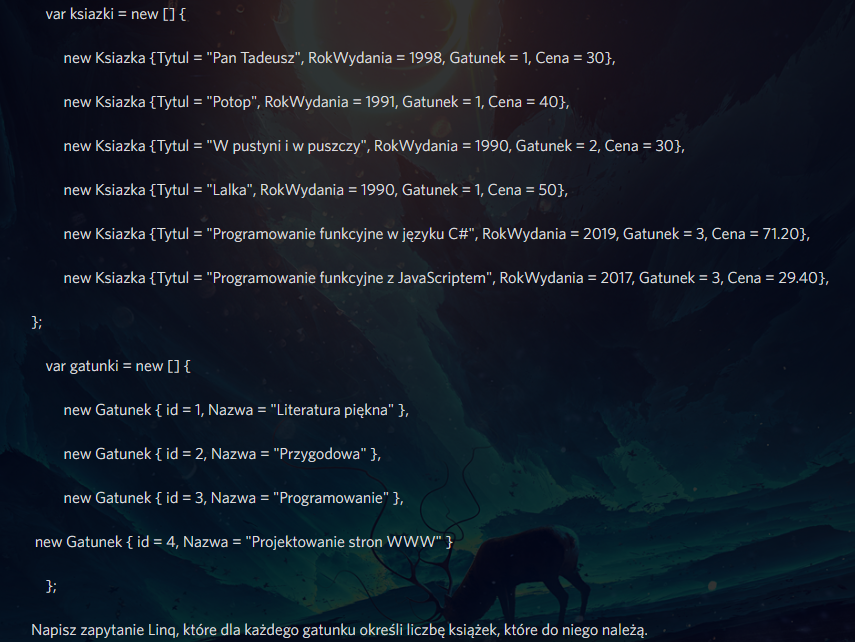
F#: Zdefiniuj nowy typ Lista reprezentujący listę łączoną mogącą przechowywać wartości dowolnego typu. Następnie napisz funkcję, zwracającą te elementy, które nie spełniają określonego warunku. Zademonstruj jej działanie.

F#: Napisz funkcję, która na podstawie określonej liczby rzutów trzema kostkami obliczy prawdopodobieństwo wyrzucenia kombinacji 1,2,3. Liczbę powtórzeń podaj jako parametr funkcji

F#: Zdefiniuj nowy typ danych reprezentujący drzewo binarne. Następnie napisz program, który wyświetli elementy tego drzewa w kolejności postorder. Zademonstruj jego działanie.

F#: Napisz funkcję, która przyjmie parę liczb i określi, która z nich jest większa po podniesieniu do kwadratu np. wywołanie wieksza (-5, 4) powinno zwrócić -5. Zastosuj dopasowanie do wzorca i wzorce warunkowe

C#: Stwórz klasy na podstawie przykładów:



F#: Zdefiniuj nowy typ Lista reprezentujący listę łączoną mogącą przechowywać wartości dowolnego typu. Następnie napisz funkcję, zwracającą te elementy, które nie spełniają określonego warunku. Zademonstruj jej działanie.

C#: Stwórz klasy na podstawie przykładów:



F#: Napisz program, który utworzy listę 100 dwuwymiarowych punktów losowych z przedziału od -20 do 20. Następnie napisz program, który wybierze te punkty, które znajdują się w promieniu r=5 od środka układu współrzędnych. Wykorzystaj funkcje modułu List

F#: Zdefiniuj nowy typ Lista reprezentujący listę łączoną mogącą przechowywać wartości dowolnego typu. Następnie napisz funkcję, zwracającą te elementy, które nie spełniają określonego warunku. Zademonstruj jej działanie.

F#: Zdefiniuj nowy typ Lista reprezentujący listę łączoną mogącą przechowywać wartości dowolnego typu. Następnie napisz funkcję, zwracającą te elementy, które nie spełniają określonego warunku. Zademonstruj jej działanie.

C#: Napisz zapytanie LINQ, które dla dowolnej policzalnej kolekcji punktów dwuwymiarowych określi ile z tych punktów znajduje się powyżej funkcji sinus. Punkty generuj jako wartości zmiennoprzecinkowe. Zakres wartości dla współrzędnej X to -10.0 : 10.0. Zakres wartości dla współrzędnej Y określ samodzielnie. Zademonstruj działanie tego zapytania. Punkty mogą być reprezentowane w dowolny sposób.

C#: Napisz funkcję generującą leniwą kolekcję punktów w przestrzeni trójwymiarowej. Każda składowa powinna być losową liczbą całkowitą z przedziału -20 do 20. Punkty możesz reprezentować w dowolny sposób np. jako tablicę lub obiekty klasy Tuple. Następnie napisz funkcję, która pozwoli przerwać generowanie tych wartości, gdy pojawi się punkt, którego trzecia składowa będzie ujemna oraz funkcję zwracającą punkt, którego poszczególne składowe będą sumą składowych wygenerowanych punktów. Zademonstruj ich działanie.

C#: Napisz funkcję generującą leniwą kolekcję punktów w przestrzeni trójwymiarowej. Każda składowa powinna być losową liczbą całkowitą z przedziału -20 do 20. Punkty możesz reprezentować w dowolny sposób np. jako tablicę lub obiekty klasy Tuple. Następnie napisz funkcję, która pozwoli przerwać generowanie tych wartości, gdy pojawi się punkt, którego pierwsza składowa będzie dodatnia oraz funkcję, która określi ile takich punktów było. Zademonstruj ich działanie.

F#: Napisz funkcję, która będzie obliczała pole i obwód określonego prostokąta. Do zamodelowania parametru i wyniku tej funkcji wykorzystaj najlepszy wg siebie typ danych.

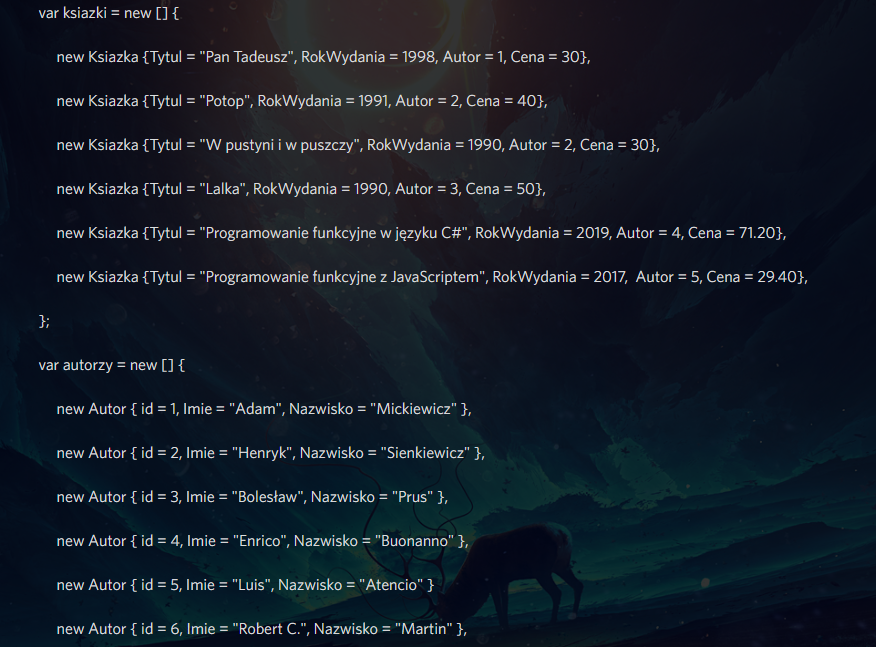
1. F#: Napisz funkcję rekurencyjną, która określi ile razy w danym łańcuchu znaków wystąpiła każda z cyfr. Jeżeli jakaś cyfra nie występuje w tekście jej wartość w wynikach powinna zawierać 0

2. F#: Napisz program, który utworzy listę 100 dwuwymiarowych punktów losowych z przedziału od -20 do 20. Następnie określ, odległość euklidesową każdego z tych punktów od początku układu współrzędnych. Wykorzystaj funkcje modułu List.

1. C#: Napisz funkcję obliczającą wartość silni w taki sposób, aby możliwe było jej wywołanie w następujący sposób: 100.Silnia()

2. C#: Stwórz generyczną klasę Lista implementującą listę łączoną i przechowującą wartość dowolnego typu. Stwórz ją w taki sposób, aby każdy węzeł był elementem tylko do odczytu. Następnie napisz w tej klasie metodę, która policzy ile wartości dodatnich jest na tej liście. Napisz również kod, który demonstruje jej działanie.

C#: Stwórz klasy na podstawie przykładów:



Napisz zapytanie Linq, które dla każdego autora określi łączną cenę książek, które napisał.

F#: Zdefiniuj nowy typ Lista reprezentujący listę łączoną mogącą przechowywać wartości dowolnego typu. Następnie napisz funkcję, która obliczy ile wyrazów na tej liście zaczyna się na literę 'a'. Napisz również kod, który demonstruje działanie przygotowanej funkcji.

C#: Stwórz generyczną klasę Lista implementującą listę łączoną i przechowującą wartość dowolnego typu. Stwórz ją w taki sposób, aby każdy węzeł był elementem tylko do odczytu. Następnie napisz metodę, która będzie poszukiwała elementu najmniejszego (dla dowolnego typu). Zademonstruj działanie klasy

C#: Stwórz klasę Samochod zawierającą pola Marka, Model, PojemnoscSilnika. Zaimplementuj w niej interfejs IComparable, który pozwoli sortować instancje tej klasy wg. właściwości PojemnosciSilnika w kolejności malejącej.

C#: Napisz funkcję obliczającą wartość silni w taki sposób, aby możliwe było jej wywołanie w następujący sposób: 100.Silnia()

Zadania z F# Pytanie 1

F#: Napisz program, który wczytując od użytkownika liczby z klawiatury zapamięta liczby parzyste. (Wprowadzanie zakończ, jeżeli użytkownik poda 0). Po zakończeniu wprowadzania danych program powinien wyświetlić je w odwrotnej kolejności (od ostatniej wprowadzonej do pierwszej).

Pytanie 2 F#: Napisz program, który utworzy listę 100 dwuwymiarowych punktów losowych z przedziału od -20 do 20. Następnie napisz program, który wybierze 10 punktów, które są najbliżej początku układu współrzędnych. Wykorzystaj funkcje modułu List.

Pytanie 3 C#: Napisz zapytanie LINQ, które dla dowolnej policzalnej kolekcji punktów dwuwymiarowych określi ile punktów znajduje się w poszczególnych ćwiartkach układu współrzędnych. Wyniki wyświetl w kolejności określonej numerem ćwiartki malejąco. Punkty mogą być reprezentowane w dowolny sposób.

Pytanie 4 C#: Napisz zapytanie LINQ, które dla tablicy dowolnych wyrazów, określi ile było wyrazów zaczynających się na każdą z liter.

C#: Napisz zapytanie LINQ, które dla dowolnej policzalnej kolekcji punktów dwuwymiarowych określi ile punktów znajduje się w poszczególnych ćwiartkach układu współrzędnych. Wyniki wyświetl w kolejności określonej numerem ćwiartki malejąco. Punkty mogą być reprezentowane w dowolny sposób.

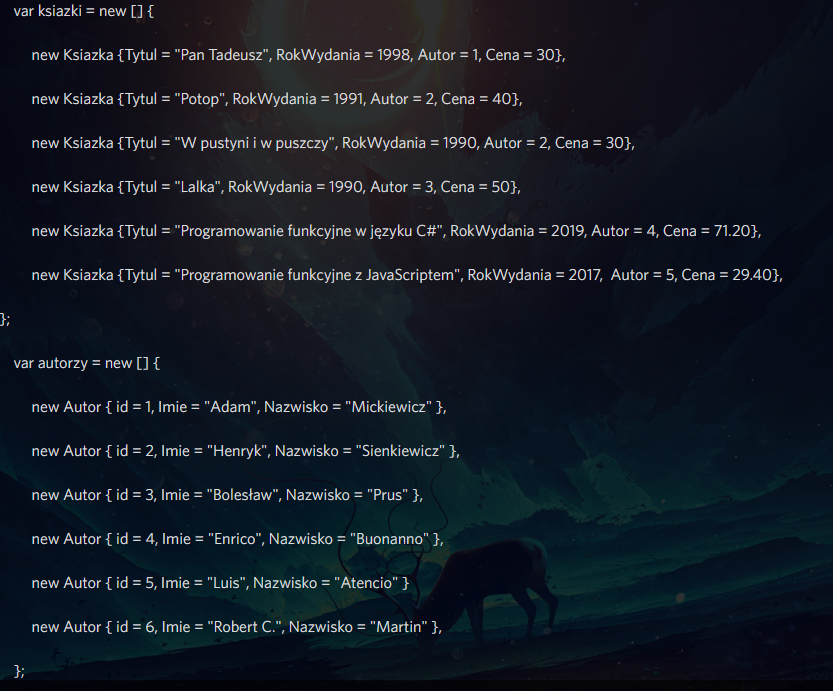
F# 1. Napisz program, który wczytując od użytkownika liczby z klawiatury, określi ile z nich było parzystych. (Wprowadzanie zakończ, jeżeli użytkownik poda 0)

2. Zdefiniuj nowy typ Lista reprezentujący listę łączoną mogącą przechowywać wartości dowolnego typu. Następnie napisz funkcję zwracającą wartość logiczną, która określi, czy wszystkie elementy listy spełniają warunek określony jako parametr. Zademonstruj jej działanie.

C# 1. Napisz zapytanie LINQ, które dla dowolnej policzalnej kolekcji punktów dwuwymiarowych określi ile punktów znajduje się w poszczególnych ćwiartkach układu współrzędnych. Wyniki wyświetl w kolejności określonej numerem ćwiartki malejąco. Punkty mogą być reprezentowane w dowolny sposób.

2. Napisz funkcję generującą leniwą kolekcję punktów w przestrzeni trójwymiarowej. Każda składowa powinna być losową liczbą całkowitą z przedziału -20 do 20. Punkty możesz reprezentować w dowolny sposób np. jako tablicę lub obiekty klasy Tuple. Następnie napisz funkcję, która pozwoli przerwać generowanie tych wartości, gdy pojawi się punkt, którego trzecia składowa będzie ujemna oraz funkcję zwracającą punkt, którego poszczególne składowe będą sumą składowych wygenerowanych punktów. Zademonstruj ich działanie.

C#: Stwórz klasy na podstawie przykładów:



Napisz zapytanie Linq, które dla autora określi cenę najtańszej książki, którą napisał. Jeżeli autor nie napisał książek zapytanie powinno zwrócić 0.