

Programowanie Graficznego Interfejsu Użytkownika

LISTA 2 [ważna do 15.05.2025]

INTERFEJSY I TABLICE NIEREGULARNE w C#

1. Utwórz klasę `Lista`, w skład której będą wchodziły:
 - a) lista liczb całkowitych,
 - b) konstruktor inicjujący klasę losowym ciągiem liczb całkowitych z przedziału 1...100 o długości zadanej argumentem konstruktora (może być zero – lista pusta),
 - c) konstruktor bezargumentowy inicjujący klasę losowym ciągiem liczb całkowitych z przedziału 1...100 o długości będącej losową liczbą z przedziału 1...5,
 - d) przeciążona metoda `ToString()` wypisująca zawartość danej listy.

Utwórz następnie dwie klasy potomne: `Lista1` i `Lista2`, które będą różniły się jedynie sposobem porównywania elementów – zaimplementuj interfejs `IComparable<T>` w każdej z tych klas, żeby można było porównywać ze sobą obiekty w ramach utworzonych klas.

W każdej klasie inne jest kryterium porównania. I tak w klasie `Lista1` ta lista jest większa, której pierwszy element jest większy, w razie równości pierwszych, decyduje drugi, itp. W przypadku porównania list o różnych długościach brak elementu jest traktowany jako element mniejszy od wszystkich możliwych – i tak np.:

$$\{\} < \{1\} < \{1,4,1,5,8,5,9\} < \{2\} < \{2,1\} < \{2,4,10,3\} < \{2,5,8\} < \{2,6\}.$$

Z kolei w klasie `Lista2` pierwszym kryterium porównawczym jest długość listy – im lista dłuższa, tym większa. Dla list o tej samej długości porównujemy kolejno elementy (tzn. podobnie, jak dla klasy `Lista1`) – wówczas np.:

$$\{\} < \{1\} < \{2\} < \{2,1\} < \{2,6\} < \{2,5,8\} < \{2,4,10,3\} < \{1,4,1,5,8,5,9\}.$$

Następnie wygeneruj po jednej kolekcji (np. tablicy albo liście) obiektów utworzonych klas i każdą z nich posortuj (wbudowaną metodą `Sort()` kolekcji). Dla każdej klasy wypisz przygotowaną kolekcję przed i po posortowaniu.

2. Wymyśl przykład klasy używającej tablic nieregularnych i przetestuj jej działanie (inicjowanie i dostęp). Przykład nie powinien być „na siłę” – chodzi o znalezienie naturalnego zastosowania takich tablic.

Grzegorz Kondrat