

DOKUMENTACJA BIBLIOTEKI UŻYTEJ PODCZAS WYKONYWANIA PROJEKTU ((System.Collections.Generic)

Filip Rybiński, Mateusz Rojek, Rafał Rudnicki

Biblioteka reprezentuje silnie wpisaną listę obiektów, do których można uzyskać dostęp za pomocą indeksu. Dostarcza metody do wyszukiwania, sortowania i manipulowania listami.

Została użyta w projekcie do przechowywania listy klientów, samolotów, lotnisk, lotów, tras, rezerwacji.

Konstruktor listy obiektów: List<T>() - Inicjuje nowe wystąpienie List<T>() klasy, która jest pusta i ma domyślnie pojemność początkową.

Właściwości listy: Count- Pobiera liczbę elementów zawartych w słowniku List<T>.

Metody:

Equals(object)- Określa, czy dany obiekt jest taki sam, jak bieżący obiekt.

(Odziedziczone po Object)

Foreach(Action<T>)- Wykonuje określoną akcję dla każdego elementu List<T>.

ToString()- Zwraca ciąg reprezentujący bieżący obiekt. (Odziedziczone po Object)

GetType()- Pobiera Type bieżące wystąpienie. (Odziedziczone po Object)

List.Remove(Object)- Usuwa pierwsze wystąpienie określonego obiektu z List<T>.

List.Add(Object)- Dodaje element do elementu List.

```
private List<Samolot> samoloty = new List<Samolot>();
private List<Lot> loty = new List<Lot>();
private List<Klient> klienci = new List<Klient>();
private List<Trasa> trasy = new List<Trasa>();
private List<Lotnisko> lotniska = new List<Lotnisko>();
```

```
for (int i = 0; i < klienci.Count; i++)
{
    Klient K = klienci[i];
    if (K.getId().Equals(id)) klienci.Remove(K);
}
```

```
public void dodajSamolot(Samolot samolot)
{
    samoloty.Add(samolot);
}
```

```
public bool equals_(Object obj)
{
    if (!(obj is Lotnisko)) return false;
    Lotnisko L = (Lotnisko)obj;
    if (Miasto.Equals(L.Miasto) && Id.Equals(L.Id) && Nazwa.Equals(L.Nazwa)) return true;
    return false;
}
```

```
Console.WriteLine(klient1.GetType());
Console.WriteLine(klient2.GetType());
```

```
foreach (Lot L in liniaLotnicza.getLoty())
{
    Console.WriteLine(L.ToString());
}
```