Eslam Mohamed Abdelfadel S9

namespace C\_\_Practices\_2

{

static internal class CollectionExtensions

{

public static IEnumerable<T> FilterV2<T>(this IEnumerable<T> CheckList, Predicate<T> predicate)

{

foreach (T item in CheckList)

{

if (predicate(item))

{

yield return item;

}

}

}

public static IEnumerable<TResult> MySelect<T, TResult>(this IEnumerable<T> List, Func<T, TResult> Selector)

{

foreach(T item in List)

{

yield return Selector(item);

}

}

public static IEnumerable<T> ToMyList<T>(this IEnumerable<T> List)

{

List<T> MyList = new List<T>();

foreach (var item in List)

{

MyList.Add(item);

}

return MyList;

}

public static T MyFirst<T>(this IEnumerable<T> List)

{

foreach(var item in List)

{

return item;

}

throw new InvalidOperationException("Sequence doesn't contains any element");

}

public static T MyFirst<T>(this IEnumerable<T> List, Func<T, bool> predicate)

{

foreach(var item in List)

{

if(predicate(item))

return item;

}

throw new InvalidOperationException("Sequence doesn't contains any element");

}

public static T MyLast<T>(this IEnumerable<T> List)

{

bool flage = false;

T ITEM = default;

foreach (var item in List)

{

ITEM = item;

flage = true;

}

if (flage)

return ITEM;

throw new InvalidOperationException("Sequence doesn't contains any element");

}

public static T MyLast<T>(this IEnumerable<T> List, Func<T, bool> predicate)

{

bool flage = false;

T ITEM = default;

foreach(var item in List)

{

if(predicate(item))

{

ITEM = item;

flage = true;

}

}

if (flage)

return ITEM;

throw new InvalidOperationException("Sequence doesn't contains any element");

}

public static T MyFirstOrDefault<T>(this IEnumerable<T> List)

{

foreach(var item in List)

{

return item;

}

return default;

}

public static T MyFirstOrDefault<T>(this IEnumerable<T> List, Func<T, bool> predicate)

{

foreach(var item in List)

{

if(predicate(item))

return item;

}

return default;

}

public static T MyLastOrDefault<T>(this IEnumerable<T> List)

{

T ITEM = default;

foreach(var item in List)

{

ITEM = item;

}

return ITEM;

}

public static T MyLastOrDefault<T>(this IEnumerable<T> List, Func<T, bool> predicate)

{

T ITEM = default;

foreach (var item in List)

{

if(predicate(item))

{

ITEM = item;

}

}

return ITEM;

}

}

}