**DAL**:

**public class Card** : IEquatable<Card>

* Конструктори

public **Card**()

public **Card**(string word, string description, string transcription = "", string imagePath = "", Difficulty difficulty = Difficulty.None)

public **Card**(Card card)

* Властивості

public **int** Id { get; set; }

public **string** Word { get; set; }

public **string** Description { get; set; }

public **string** Transcription { get; set; }

public **Difficulty** Difficulty { get; set; }

public **string** ImagePath { get; set; }

* Методи

public override **string** ToString()

public override **int** GetHashCode()

public override **bool** Equals(object obj)

public **bool** Equals(Card card)

**public class Deck** : IEquatable<Deck>, IEnumerable<Card>, ICollection<Card>

* Конструктори

public **Deck**()

public **Deck**(Deck deck)

* Властивості

public **int** Id { get; set; }

public **List<Card>** Cards { get; }

public **string** DeckName { get; set; }

public **int** Count { get => Cards.Count; }

public **string** TagName { get; set; }

public **bool** IsReadOnly => false;

* Індексер

public **Card** this[int index]

* Методи

public **void** Add(Card card)

public **void** AddRange(IEnumerable<Card> cards)

public **void** InsertCard(int index, Card card)

public **void** InsertRange(int index, IEnumerable<Card> cards)

public **bool** Remove(Card card)

public **void** RemoveAt(int index)

public **void** MoveCard(int oldIndex, int newIndex)

public **int** IndexOf(Card card)

public **void** Clear()

public **bool** Contains(Card card)

public **void** CopyTo(Card[] array, int arrayIndex)

public override **string** ToString()

public override **int** GetHashCode()

public override **bool** Equals(object obj)

public **bool** Equals(Deck deck)

public **IEnumerator<Card>** GetEnumerator()

IEnumerator IEnumerable.GetEnumerator()

**public static class Repository**

* Методи

public static **Deck** FetchDeck(string deckName)

public static **Deck** FetchDeck(int id)

public static **Card** FetchCard(int id)

public static **IEnumerable<Deck>** FetchAllDecks()

public static **IEnumerable<Card>** FetchAllCards(string deckName = "")

public static **int** AddCard(Card card)

public static **void** RemoveCard(int id)

public static **void** UpdateCard(int id, Card card)

public static **void** AddTagToCard(int cardId, string tagName)

public static **void** RemoveTagFromCard(int cardId, string tagName)

public static **int** AddDeck(Deck deck)

public static **void** RemoveDeck(int id)

public static **void** UpdateDeck(int id, Deck deck)

public static **void** AddCardToDeck(int deckId, int cardId)

public static **void** RemoveCardFromDeck(int deckId, int cardId)

**BLL**:

**public class AppHandler**

* Конструктори

public **AppHandler**(Deck deck)

public **AppHandler**(int deckId)

public **AppHandler**(string deckName)

* Властивості

public **Deck** Deck { get; }

* Методи

public **void** Start()

public **void** Stop()

public **void** RadndomizeDeck()

* Підписники

public **void** FlipCard(object sender, AppHandlerFlipEventArgs e)

public **void** NextCard(object sender, AppHandlerNextCardEventArgs e)

public **void** MoveCardIntoDeck(object sender, AppHandlerMoveCardEventArgs e)

**public class DeckEditor**

* Конструктори

public **DeckEditor**()

public **DeckEditor**(Deck deck)

public **DeckEditor**(string deckName)

public **DeckEditor**(int deckId)

* Властивості

public Deck Deck { get; private set; }

* Підписники

public void AddCard(object sender, DeckEditorAddCardEventArgs e)

public void RemoveCard(object sender, DeckEditorRemoveCardEventArgs e)

public void ChangeDeck(object sender, DeckEditorDeckEventArgs e)

public void SaveChanges(object sender, DeckEditorDeckEventArgs e)

public void ExitEditor(object sender, ExitDeckEditorEventArgs e)

**public class** Statistics

* Конструктори

public **Statistics**()

* Властивості

public **TimeSpan** TimeSpentToday { get; set; }

public **TimeSpan** AvarageTimePerDay { get; set; }

public **List<Card>** CardsLearnedToday { get; set; }

* Підписники

public **void** AddSpentTimeToday(object sender, StatAddSpentTimeEventArgs e)

public **void** AddCardLearned(object sender, StatCardLearnedEventArgs e)

public **void** ManageStatistics()