

# Projekt na zaliczenie przedmiotu „Programowanie obiektowe C#”

## Dokumentacja

Link do repozytorium - <https://github.com/aureliuszek/pasjans>

Współautorzy:

**Aureliusz Wala nr indeksu 11861**

**Paweł Wójsik nr indeksu 11881**

**Karol Niemiec nr indeksu 12035**

Oświadczenie współautorów o procentowym udziale w realizacji projektu:

**Paweł Wójsik – 40%**

**Zrealizowane zadania: Konstrukcja programu, logika programu, Program, Extensions, Stack**

**Aureliusz Wala - 30%**

**Zrealizowane zadania: Logika programu, MainForm, Results, dokumentacja**

**Karol Niemiec – 30%**

**Zrealizowane zadania: Logika programu, Card, Images, Szata graficzna**

## Class MainForm

### Funkcje

- Opis

Zwraca pusty rząd

```
public Stack GetFreeStack();
```

### Parametry

brak

### Zwraca

Pusty obiekt klasy Stack

- Opis

Aktualizuje wynik

```
public void UpdateScore ();
```

Parametry

brak

Zwraca

Brak

- Opis

Wyświetla komunikat o wyniku zapisuje wynik i zatrzymuje czas

```
public void FinishGame ();
```

Parametry

brak

Zwraca

brak

- Opis

Resetuje wszystkie pola czas wynik i liczbę ruchów i rozstawia na nowo karty

```
public void StarNewGame ();
```

Parametry

brak

Zwraca

brak

- Opis

Usuwa wszystkie obiekty child z Panelów

```
public void ClearAllControls();
```

## Parametry

brak

## Zwraca

brak

## Class Extension

### Funkcje

- Opis

Zmienia BackgroundImage elementów z listy na podany w parametrze img

```
public static void ChangeBackgroundImage(this List<PictureBox> ls, Bitmap img);
```

## Parametry

ls – Lista PictureBox'ów

img - Zdjęcie

## Zwraca

brak

- Opis

Pobiera następną kartę z podanego stosu kart

```
public static int GetNextCardPriorityInStack(this Stack s, Card c);
```

## Parametry

s – obiekt typu Stack

c – obiekt typu Card

## Zwraca

Wartość następnej karty

- Opis

Pobiera poprzednią kartę z podanego stosu kart

```
public static int GetPrevCardPriorityInStack (this Stack s, Card c);
```

#### Parametry

s – obiekt typu Stack

c – obiekt typu Card

#### Zwraca

Wartość poprzedniej karty

- Opis

Zmienia szerokość wszystkich elementów z podanej listy na wartość parametru width

```
public static void ChangeWidth(this List<PictureBox> ls, int width);
```

#### Parametry

ls – lista pictureBox'ów

width – wartość zmienianej szerokości

#### Zwraca

Brak

- Opis

Zmienia wysokość wszystkich elementów z podanej listy na wartość parametru height

```
public static void ChangeHeight(this List<PictureBox> ls, int height);
```

#### Parametry

ls – lista pictureBox'ów

height – wartość zmienianej wysokości

#### Zwraca

brak

- Opis

Zwraca listę typu Card z liczbą elementów – 104 o podanej wartości cardType

```
public static List<Card> GenerateCard(CardType cardType);
```

#### Parametry

cardType – zmienna typu CardType

#### Zwraca

listę typu Card

- Opis

Zapisuje wartość PictureBox'a do zmiennych klasy Extension

```
public static void SavePB(this PictureBox pb,Card c,Stack stack);
```

#### Parametry

pb – zmienna typu PictureBox

c – zmienna typu Card

stack – zmienna typu Stack

#### Zwraca

Brak

- Opis

Zapisuje wartość PictureBox'a do zmiennych klasy Extension

```
public static void SavePB(this PictureBox pb,Card c,Stack stack);
```

#### Parametry

pb – zmienna typu PictureBox

c – zmienna typu Card

stack – zmienna typu Stack

#### Zwraca

brak

## Class Images

### Zmienne

`public static Bitmap Back;` - Zdjęcie tylnej części karty

`public static Bitmap Frame;` - Zdjęcie obramowania

`public static int Width;` - Szerokość obramowania

`public static int Height;` - Wysokość obramowania

`public static bool f;` – flaga kontrolna

`public static bool f2;` – flaga kontrolna2

`public static Stack stack ;` - zmienna klasy Stack

## Class Card

- Opis

Kopiuje i zwraca obiekt typu Card;

`public Card Copy();`

### Parametry

brak

### Zwraca

Obiekt typu Card

## Class Stack

### Funkcje

- Opis

Dodaje kartę zależnie od parametrów do Staku

`public void AddCardToStack(Card card, bool is6 = false, bool n = false)`

### Parametry

`card` – zmienna klasy Card

`bool is6` – przekazane zostanie 6 kart

`bool n` – sprawdza czy karta jest brana z kart dodatkowych i odkrywa ją

Zwraca

brak

- Opis

Sprawdza czy jest talia została ułożona w prawidłowej kolejności

```
private void CheckIfFullCardSort(Stack stack)
```

Parametry

stack – zmienna klasy Stack

Zwraca

Brak

- Opis

Dodaje do staku zmienną typu PictureBox

```
private void AddToBase(PictureBox p)
```

Parametry

p – zmienna typu PictureBox

Zwraca

brak

- Opis

Sprawdza czy można przenieść kartę

```
private bool CheckIfCanMove(List<Card> list, Card p)
```

Parametry

list – zmienna typu List

p – zmienna typu Card

Zwraca

czy można przenieść kartę

- Opis

Pobiera wszystkie karty po podanym indeksie

```
private List<PictureBox> ClaimAllCardsAfterIndex(PictureBox p)
```

#### Parametry

p – zmienna typu PictureBox

#### Zwraca

karty po podanym indeksie

- Opis

Miesza listę i dodaje karty do Stack

```
public void AddCardToStack(List<Card> list, int numbers)
```

#### Parametry

list – zmienna typu List

numbers – ilość kart

#### Zwraca

brak

- Opis

Czyści PictureBox'y oraz karty

```
public void Clear()
```

#### Parametry

brak

#### Zwraca

Brak



- Opis

Czyści PictureBox'y

```
public void Clear(PictureBox p)
```

Parametry

p – zmienna typu PictureBox

Zwraca

Brak