

Część I. Aplikacja internetowa

Zadanie: aplikacja webowa (lista TODO)

W ramach zadania należy zaimplementować aplikację webową, która będzie umożliwiała dodawanie nowych zadań, które będą wyświetlane w formie listy. Wpisy listy składają się z dwóch elementów: kwadracik z ptaszkiem (checkbox) oraz tekst. Nad listą znajduje się panel, który umożliwia wprowadzenie tekstu (zadanie) oraz przycisku Dodaj. Powyżej znajduje się napis 'TODO List:'. Dodatkową funkcjonalnością jest obsługa zaznaczenia checkboxa, który jest wyświetlany jako część wpisu na liście. Po zmianie stanu checkboxa (na zaznaczony), należy zmienić styl tekstu tak, aby został on przekreślony. Po kolejnej zmianie stanu checkboxa (na odznaczony), należy zmienić styl tekstu tak, aby on nie był przekreślony. Aplikację należy zaimplementować w jednym z dostępnych frameworków ReactJS lub Angular.

W trakcie implementacji należy wykorzystać style z biblioteki 'Bootstrap'. Biblioteka jest już dołączona do projektu, wystarczy ją tylko zaimportować.

ReactJS	<code>import 'bootstrap/dist/css/bootstrap.css'</code>
Angular	Plik: angular.json Pole: <code>styles</code> w <code>build</code> Dodaj do tablicy: <code>"./node_modules/bootstrap/dist/css/bootstrap.css"</code>

Podgląd aplikacji webowej po starcie:

TODO List:

Dodaj

Podgląd aplikacji webowej po wprowadzeniu kilku zadań:

TODO List:

Dodaj

☐Zadanie 1

☒Zadanie 2

☐Zadanie 3

☒Sprawdzić

☐Test

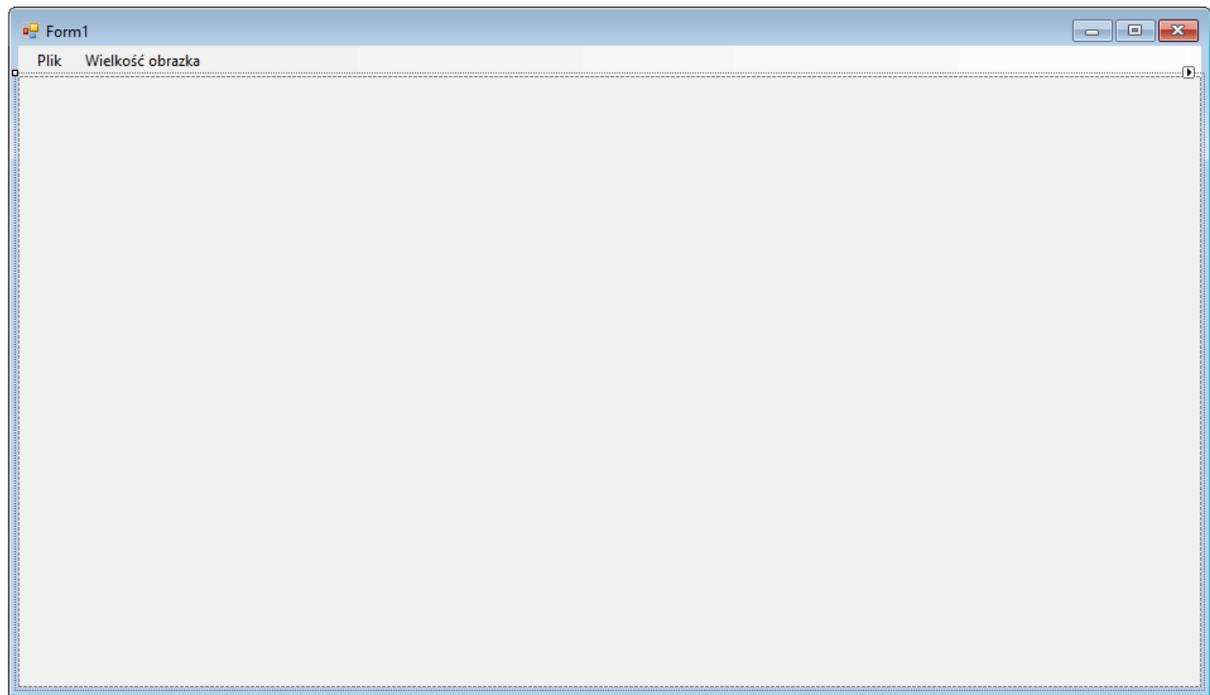
Ogólne klasy z biblioteki bootstrap:

- btn
- btn-secondary
- container
- h2
- text-center

- form-control

Część II. Aplikacja okienkowa – przeglądarka zdjęć

Napisz program, który odtworzy następujący wygląd okna:

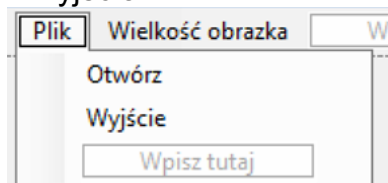


Opis wyglądu aplikacji. Elementy od góry:

Program zawiera Menu a w nim Plik oraz Wielkość obrazka:

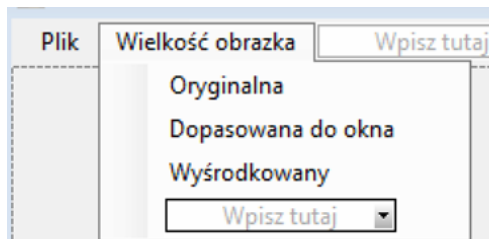
Menu plik to opcje:

- Otwórz
- Wyjście



Menu wielkość obrazka to opcje:

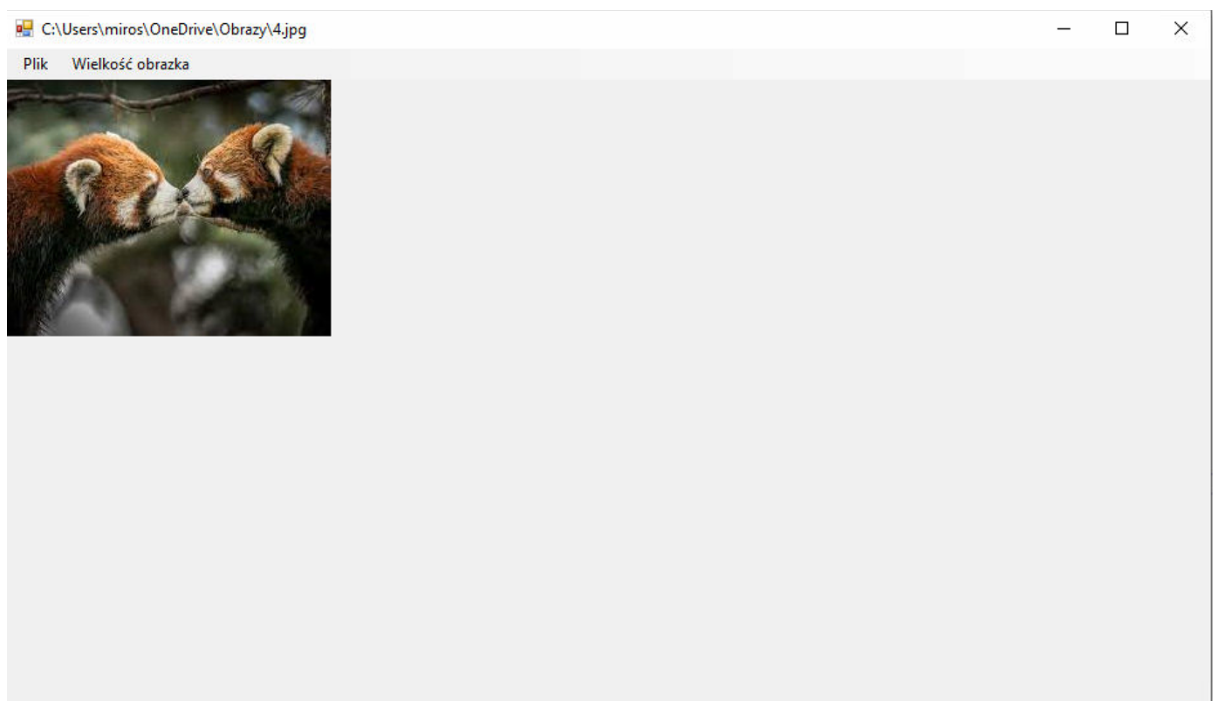
- Normalna
- Dopasowana do okna
- Wyśrodkowany



W oknie powinna być zadokowana kontrolka PictureBox dopasowująca swój rozmiar do rozmiaru okna aplikacji

Działanie aplikacji

1. Po wybraniu opcji „Otwórz” pojawia się okno dialogowe OpenFileDialog w którym użytkownik wybiera plik, który następnie jest odczytywany i jego zawartość pojawia się w kontrolce PictureBox. Nazwa odczytanego pliku pojawia się na górnej belce okna np.:



Po wybraniu z menu opcji „Wyjście”, program powinien zakończyć swoje działanie.

W kontrolce OpenFileDialog i ustaw właściwość Filter na:
 JPEG Files (*.jpg)|*.jpg|PNG Files (*.png)|*.png|BMP Files (*.bmp)|*.bmp|All files (*.*)|*.*

W kontrolce OpenFileDialog i ustaw właściwość Title na: **Wybierz plik**

WindowsForms przydatne kontrolki, metody i stałe:

MenuStrip

OpenFileDialog

PictureBox

```
openFileDialog1.ShowDialog() == DialogResult.OK
```

```
pictureBox1.Load(openFileDialog1.FileName);
```

```
pictureBox1.SizeMode
```

```
PictureBoxSizeMode.Normal;
```

```
PictureBoxSizeMode.StretchImage;
```

```
PictureBoxSizeMode.CenterImage;
```

Założenia do programu

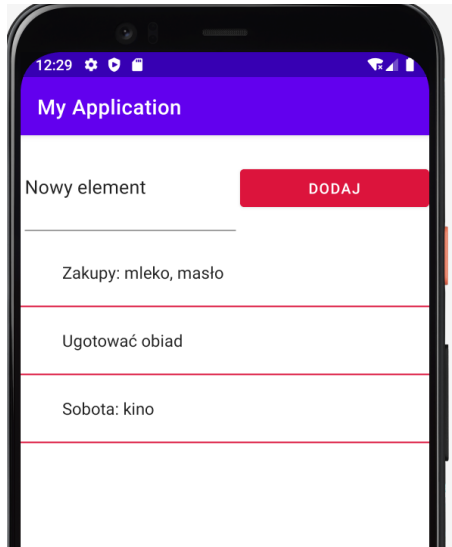
- Program wykonywany w oknie.
- Obiektowy język programowania zgodny z zainstalowanym na stanowisku egzaminacyjnym: C++ lub C# lub Java lub Python.
- Program powinien być zapisany czytelnie, z zasadami czystego formatowania kodu, należy stosować znaczące nazwy zmiennych, funkcji i kontrolek.

Podjmij próbę kompilacji i uruchomienia. Informacje dotyczące dokumentacji i zrzutów ekranowych umieszczono w części IV zadania.

Kod aplikacji przygotuj do nagrania. W podfolderze *okienkowa* powinno znaleźć się archiwum całego projektu o nazwie *okienkowa.zip*

Część III. Aplikacja mobilna

Wykonaj aplikację mobilną za pomocą środowiska programistycznego dostępnego na stanowisku egzaminacyjnym oraz uruchom ją w dostępnym emulatorze systemu mobilnego. Aplikacja umożliwia wpisywanie prostych notatek. Dane początkowe k.



Na obrazie przedstawiono stan po uruchomieniu aplikacji mobilnej. W stanie początkowym aplikacja wyświetla trzy notatki tak jak na obrazie powyżej. Notatki możesz wpisać ręcznie do kodu źródłowego. W zależności od zastosowanego środowiska programistycznego oraz emulowanego systemu wynik końcowy może nieznacznie się różnić od przedstawionego.

Elementy aplikacji:

- Pole edycyjne z podpowiedzą: „Nowy element”
- Przycisk o treści: „Dodaj”
- Lista (Element widoku listy)
- Rozmieszczenie elementów takie jak na powyższym obrazku

Działanie aplikacji:

- W stanie początkowym wyświetlone są trzy notatki jak na powyższym obrazku
- Po wpisaniu do pola edycyjnego tekstu i wybraniu przycisku, jest on zapisywany jako ostatni element widoku listy.

Założenia aplikacji:

- Interfejs użytkownika zapisany za pomocą języka znaczników wspieranego w danym środowisku (np. XAML, XML).
- Zastosowany typ rozkładu liniowy wertykalny (Linear / Stack lub inny o tej idei) z zagłębionym rozkładem liniowym horyzontalnym dla pola edycyjnego i przycisku.
- Cechy przycisku: Kolor tła #DC143C, biały kolor czcionki
- Cechy elementów listy: domyślny kolor czcionki, kolor separatora Crimson, separator jest widoczny (np. poprzez ustawienie wysokości separatora)
- Notatki są zapisywane w dowolnej kolekcji z elementami typu napisowego połączonej z widokiem listy jako jej źródło danych
- Wybranie przycisku powoduje dopisanie do kolekcji treści z pola edycyjnego, widok listy jest automatycznie odświeżany

– Aplikacja powinna być zapisana czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji.

Podjmij próbę kompilacji i emulacji. Informacje dotyczące dokumentacji i zrzutów ekranowych umieszczono w części IV zadania.

Kod aplikacji przygotuj do nagrania. W podfolderze *mobilna* powinno znaleźć się archiwum całego projektu o nazwie *mobilna.zip*

Typy, metody i pola wykorzystywane do odświeżenia widoku listy dla AndroidStudio:

`String[]`

`ArrayList<String>`

`ArrayAdapter<String>`

`Arrays.asList(tablicaNapisów)`

Konstruktor i metody klasy `ArrayAdapter`

`ArrayAdapter(Context context, int resource, List<T> objects)`

Np. `ArrayAdapter(this, Android.R.layout.simple_list_item_1,arrayList);`

`notifyDataSetChanged()` Notifies the attached observers that the underlying data has been changed and any View reflecting the data set should refresh itself.

Metody klasy `ListView`

`setAdapter(AdapterObj)`

Część IV. Dokumentacja aplikacji

Wykonaj dokumentację do aplikacji utworzonych na egzaminie. W kodzie źródłowym aplikacji konsolowej utwórz nagłówek funkcji przeszukującej, według wzoru zgodnie z listingiem 1. Nagłówek powinien znaleźć się w kodzie źródłowym nad funkcją. W miejscu znaków <> należy podać nazwę funkcji, nazwy argumentów wraz z krótkim opisem, typ zwracany wraz z krótkim opisem, krótki opis zawierający przynajmniej nazwę algorytmu. Gdy funkcja nie ma argumentów – zapisać „brak”. W miejscu autor należy podać swoje nazwisko i imię.

UWAGA: Dokumentację umieścić w komentarzu (wieloliniowym lub kilku jednoliniowych). Znajdujący się w listingu 1 wzór dokumentacji jest bez znaków początku i końca komentarza, gdyż te są różne dla różnych języków programowania

Listing 1. Wzór dokumentacji funkcji

```
*****
nazwa funkcji: <tu wstaw nazwę funkcji>
argumenty: <nazwa argumentu> - <co przechowuje>
           < nazwa argumentu> - <co przechowuje>
typ zwracany: <nazwa typu>, <co jest zwracane>
informacje: <opis>
autor: <Nazwisko i imię zdającego>
*****
```

Wykonaj zrzuty ekranu dokumentujące uruchomienie aplikacji utworzonych podczas egzaminu. Zrzuty powinny obejmować cały obszar ekranu monitora z widocznym paskiem zadań. Jeżeli aplikacja uruchamia się, na rzucie należy umieścić okno z wynikiem działania programu oraz otwarte środowisko programistyczne z projektem lub okno terminala z kompilacją projektu. Jeżeli aplikacja nie uruchamia się z powodu błędów kompilacji, należy na rzucie umieścić okno ze spisem błędów i widocznym otwartym środowiskiem programistycznym. Wykonać należy tyle zrzutów ile interakcji podejmuje aplikacja. Wymagane zrzuty ekranu:

- Aplikacja internetowa – dowolna liczba zrzutów nazwanych *internetowa1*, *internetowa2* ...
- Aplikacja okienkowa – dowolna liczba zrzutów nazwanych *window1*, *window2* ...
- Aplikacja mobilna – dowolna liczba zrzutów nazwanych *mobile1*, *mobile2* ...

W edytorze tekstu pakietu biurowego utwórz plik z dokumentacją i nazwij go *egzamin*. Dokument powinien zawierać podpisane zrzuty ekranu oraz zapisane informacje:

- o Nazwę systemu operacyjnego, na którym pracował zdający.
- o Nazwy środowisk programistycznych, z których zdający korzystał na egzaminie.
- o Nazwy języków programowania.
- o Nazwa emulatora systemu mobilnego.

Zrzuty ekranu i dokument umieścić w podfolderze *dokumentacja*.

Nagraj folder z rezultatami pracy. W folderze z Nazwiskiem i imieniem zdającego powinny się znajdować podfoldery: internetowa, okienkowa, mobilna, dokumentacja.

W folderze dokumentacja: pliki ze zrzutami oraz plik egzamin.

W folderze internetowa: spakowany cały projekt aplikacji internetowej, część serwerowa oraz klientowa

W folderze okienkowa: spakowany cały projekt aplikacji okienkowej, pliki z kodem źródłowym, opcjonalnie plik uruchomieniowy

W folderze mobilna: spakowany cały projekt aplikacji mobilnej, pliki ze źródłami interfejsu i logiki.

Upewnij się, że wszystko co stworzyłeś zostało prawidłowo nagrane