

## Wytyczne projektu - Windows C# Application Store

### Cel ćwiczenia:

W trakcie realizacji zadania student opracuje aplikację w języku C# typu [Windows Universal Platform](#) w Microsoft Visual Studio 2017, której zadaniem będzie pobieranie i odpowiednie zarządzanie oraz wyświetlanie danych kursów walut obcych z serwisu NBP w ramach jednego z możliwych sposobów:

- [http://www.nbp.pl/home.aspx?f=/kursy/instrukcja\\_pobierania\\_kursow\\_walut.html](http://www.nbp.pl/home.aspx?f=/kursy/instrukcja_pobierania_kursow_walut.html).
- <http://api.nbp.pl/>

### Polecenie ćwiczeniowe:

Do zaliczenia zadania wymagane jest napisanie programu, który posiada następujące cechy:

1. Dwie strony (Page). Na głównej stronie „Średnie kursy walut” należy zaprojektować następujące panele:
  - a. Panel wyświetlający w czytelnie sformatowanej tabelce nazwę waluty, jej skrót oraz jej kurs na bieżący dzień;  
Dla uatrakcyjnienia interfejsu obok każdej waluty można wyświetlić flagę państwa (zobacz: <https://countryflags.io/> oraz konwersja skrótu waluty do skrótu nazwy państwa <https://restcountries.eu/>);  
Po kliknięciu na wybraną walutę z tabelki sterowanie programem powinno zostać przekazane do następnej strony zatytułowanej „Historia kursu ???/PLN”, gdzie ??? oznacza symbol wybranej waluty – szczegóły patrz punkt 2 niniejszej specyfikacji;
  - b. Panel (np. obok po lewej) wyświetlający kontrolkę wyboru daty bądź listę dat opublikowanych kursów z innych dni (od 2 stycznia 2002); Po wybraniu dowolnej daty zawartość panelu z punktu (a) powinna zostać zmodyfikowana o dane odczytane z tabeli z NBP dla wybranej daty;  
Należy jednocześnie pamiętać o zabezpieczeniu aplikacji przed wyborem daty, kiedy nie ma w tym dniu publikacji kursu; W tym celu należy albo uniemożliwić wybór takiej daty bądź odpowiednio użytkownika poinformować,
  - c. Pasek aplikacji na górze zawierający przynajmniej ikonę do wyjścia z programu; Mile widziane inne opcje;
2. Na drugiej stronie „Historia waluty ???/PLN” należy zaprojektować następujące elementy interfejsu użytkownika oraz ich funkcjonalności:
  - a. Kontrolki umożliwiające szybkie i łatwe ustalenie daty początkowej oraz końcowej wyświetlania historii kursu danej waluty  
uwaga:
    - i. należy zabronić ustawienia daty starszej niż tej, dla której opublikowano dane dot. walut w serwisie NBP;
    - ii. nie powinno się ograniczać wyświetlania historii tylko do jednego roku;
  - b. Element graficzny rysujący wykres przebiegu kursu danej waluty (w tym np. etykiety osi, legenda, siatka itp.) w przedziale czasowym określonym przez w/w kontrolki;
  - c. Pasek postępu wczytywania nowych danych z NBP oraz (opcjonalnie) zapisywanie lokalnie już raz pobranych danych,
  - d. Możliwość zapisania wykresu do pliku np. JPG z opcją wskazania katalogu docelowego (zaimplementowane np. w postaci dolnego paska aplikacji i odpowiedniego przycisku);
  - e. Mile widziane jest zaimplementowanie gestu powrotu do poprzedniej strony poprzez przesunięcie palcem/myszka w kierunku prawym;
  - f. Pasek aplikacji zawierający przynajmniej przyciski powrotu do poprzedniej strony oraz wyjścia z aplikacji;
3. Aplikacja powinna mieć zaimplementowany mechanizm przywracania stanu aplikacji i nawigacji obiektów Page oraz przechowywać dane użytkownika tj.:
  - a. Ostatnio otwartą stronę aplikacji;
  - b. W przypadku strony pierwszej - datę, dla której ostatnio były wyświetlane kursy walut w tabelce (punkty 1a,b) (chyba, że data nie istnieje już w archiwum NBP, i wtedy wyświetlić tabelę dla daty starszej);
  - c. Dla strony historii waluty ostatnio przeglądanej walutę wraz z ustawionymi datami początkową i końcową; Dodatkowo, jeżeli aplikacja została zamknięta w trakcie wyświetlania historii kursu, to ta strona powinna zostać przywrócona po ponownym uruchomieniu aplikacji.
4. Aplikacja powinna mieć zaprojektowany interfejs graficzny odpowiednio reagujący i skalujący się w przypadku zmiany wielkości ekranu urządzenia oraz jego orientacji (poziomy, pionowy).
5. Do poprawnego wykonania zadania na pewno przyda się:
  - a. klasa [HttpClient](#) i inne klasy, które posłużą do pobrania dokumentu XML i TXT ze strony NBP;  
Przydatne opracowania:
    - <http://www.asp.net/web-api/overview/advanced/calling-a-web-api-from-a-net-client>
    - <http://www.dreamincode.net/forums/topic/370811-wait-for-task-to-complete/>

- Pobieranie jednego dokumentu (NBP API) i raportowanie progresu lub tutaj
- <https://msdn.microsoft.com/en-us/library/jj155756.aspx>
- b. Do rysowania można użyć biblioteki [WinRT XAML Toolkit \(przykład użycia\)](#) lub [SyncFusion](#)
- c. Parser plików XML
  - <http://www.geekchamp.com/articles/how-to-read-xml-files-in-windows-8---winrt>
  - <http://stackoverflow.com/questions/566167/query-an-xdocument-for-elements-by-name-at-any-depth>
  - <http://stackoverflow.com/questions/11496570/hi-how-to-parse-xml-with-descendants-within-descendants-c-sharp>
  - <http://stackoverflow.com/questions/14297947/xml-parsing-in-winrt>
  - <https://chodounsky.net/2013/10/23/descendants-vs-elements-in-linq-to-xml/>
- d. Czytanie obiektów JSON w UWP
  - <https://docs.microsoft.com/en-us/uwp/api/windows.data.json.jsonobject>
- e. Ustawienie responsywności GUI
  - <https://docs.microsoft.com/en-us/windows/uwp/design/layout/responsive-design>
  - <https://docs.microsoft.com/en-us/windows/uwp/design/layout/alignment-margin-padding>
  - <https://docs.microsoft.com/en-us/windows/uwp/design/layout/layout-panels>
- f. Można wykorzystać inne źródła – wedle uznania.

#### Kryteria oceny:

Ocenie podlegać będą:

1. Zgodność aplikacji z wytycznymi,
  2. W pełni wykorzystanie mechanizmów Data Binding, Windows.Storage, app lifecycle and state,
  3. Przyjazny i intuicyjny interfejs użytkownika wolny od błędów niepoprawnego użytkownika,
  4. Awaryjność, Czytelność kodu,
  5. Na ocenę 3 wystarczy prawidłowa implementacja funkcjonalności tylko pierwszej strony „Średnie kursy walut”.
- Uwaga! Na ocenę 3 należy wykonać również punkty 3 i 4.

## Wytyczne projektu - WinJS Application Store

#### Cel ćwiczenia:

W ramach zaliczenia należy zaprojektować oraz zaimplementować aplikację zgodnie z wytycznymi z poprzedniego zadania korzystając z języka WinJS typu [Windows Universal Platform](#) w [Microsoft Visual Studio 2017](#). Z tym, że w tym przypadku pomocne będą następujące źródła:

- a. Klasa [HttpClient](#) i inne klasy, które posłużą do pobrania dokumentu XML i TXT ze strony NBP;
  - <http://msdn.microsoft.com/en-us/library/windows/apps/dn469430.aspx>
  - <http://msdn.microsoft.com/en-us/library/windows/apps/hh868282.aspx>
- b. Element *canvas* pomocny w rysowaniu
  - <http://msdn.microsoft.com/en-us/library/windows/apps/hh465064.aspx>
- c. Parser plików XML
  - <http://msdn.microsoft.com/en-us/library/windows/apps/hh452745.aspx>
- d. Czytanie obiektów JSON w WinJS
  - <https://msdn.microsoft.com/en-us/magazine/dn198241.aspx>
- e. Można wykorzystać inne źródła – wedle uznania.

#### Kryteria oceny:

Te same, co wypisane w ramach poprzedniego zadania.