

Platforma .NET – Laboratorium  
Informatyka – stopień pierwszy – tytuł inżyniera  
Studia stacjonarne oraz niestacjonarne  
**Zbiór zadań**

\*\*\*lista zadań w trakcie aktualizacji\*\*\*

Zbiór zadań oraz problemów zawiera listę zadań do wykonania w ramach laboratorium przedmiotu Platforma .NET. Zadania zostały podzielone na sześć głównych tematów.

## 1 Aplikacje w trybie konsoli

Podstawowe aplikacje tworzone w języku C# oparte o konsolę tekstową<sup>1</sup> oraz kompilacja projektu w środowisku Visual Studio:

1. przygotować kompilację przykładu „Matching Game”, kod źródłowy jest dostępny pod adresem: <https://github.com/dotnet/samples/tree/main/windowsforms/matching-game/net45/cs>,
2. utworzyć podstawowy program typu Hello World!, w ramach języka C# jako zwykłą aplikację .NET (zwrócić uwagę na typ projektu),
3. utworzyć program obsługujący parametry wejściowe do programu oraz odczytujący zmienne otoczenia (zmienne środowiskowe),
4. opracować program będący odpowiednikiem polecenia ls, dir z konsoli (bez stosowania rekurencji).
5. napisać program wykorzystujący systemowe pojęcie strumieni oraz potoków do zamieniania podanego słowa innym słowem, przykład wywołania programu byłby następujący:

```
progZ4 słowo1 słowo2 <fileIn.txt >fileOut.txt
```

6. opracować przykład prezentujący oddzielną bibliotekę DLL i program w konsoli (korzystanie z DLL może być zrealizowane w wersji statycznej lub opcjonalnie dynamicznej).

---

<sup>1</sup>Warto wyszukać w sieci Internet hasło: „101 Samples for Visual Studio”. Niestety, na dzień 03.03.2024, strona z przykładami już nie jest dostępna. Jednakże pod adresem: <https://github.com/microsoftarchive/msdn-code-gallery-microsoft/tree/master/Visual%20Studio%20Product%20Team> znajduje się archiwum z wieloma przykładami podobnymi przykładów z podanego hasła. Nowy zestaw przykładów można odszukać pod adresem: <https://github.com/dotnet/samples>.

## 2 Proste aplikacje GUI

Nieskomplikowane aplikacje GUI tworzone w języku C#:

1. napisać program zegar (plus funkcja alarmu) wyświetlający aktualny czas (dla uproszczenia w postaci cyfrowej) i datę,
2. opracować program, który wykonuje cztery podstawowe operacje arytmetyczne, wartości argumentów mają być wprowadzane za pomocą pól tekstowych,
3. podać program, który zaprezentuje możliwości klasy MessageBox (a dokładniej metody Show) z przestrzeni System.Windows,
4. opracować odpowiednik programu kalkulator, który jest dostępny w systemie Windows,
5. zaprezentować komponent Listbox, tworząc program prezentujący podstawowe możliwości tego komponentu,
6. wykorzystać komponent ListView, tworząc prostą listę teleadresową, wprowadzić możliwość wyszukiwania danych na liście oraz sortowanie np. po nazwisku, czy dacie urodzin,
7. przygotować program pokazujący listę dostępnych „koderów<sup>2</sup>” dla różnych formatów plików graficznych 2D, wraz z prezentacją podstawowych informacji o danym „kodeku”.

## 3 Biblioteka klas bazowych

Programy powinny wykorzystywać typy i klasy zdefiniowane w BCL:

1. opracować program z interfejsem użytkownika pokazujący różnicę w dniach, godzinach, sekundach pomiędzy dwoma podanymi datami,
2. podać program dla konsoli do weryfikacji adresu IP w formacie:

`XXX.XXX.XXX.XXX,`

gdzie *X* to cyfra oraz e-mail za pomocą wyrażenia regularnego.

3. program demonstrujący funkcje klasy BigInteger na przykładzie szyfru asymetrycznego RSA z podstawowym interfejsem graficznym,
4. opracować prostą bazę danych do przechowywania informacji o osobach jak np.: adres, numer telefonu wykorzystując dostępne klasy kolekcji w .NET, w zadaniu nie wykorzystujemy systemów dostępu do danych wbudowanych w platformę .NET,
5. napisać program do wyszukiwania wielkości największej bądź najmniejszej za pomocą klas oraz możliwości dostępnych w przestrzeni System.Threading,
6. opracować aplikację do wysyłania listu e-mail (również z załącznikami),
7. podać definicję własnej klasy zgodnej z pętlą typu foreach.

---

<sup>2</sup>W rozwiązaniu zadania przydatna będzie klasa ImageCodecInfo: <https://docs.microsoft.com/en-gb/dotnet/api/system.drawing.imaging.imagecodecinfo?view=netframework-4.8>.

## 4 Zaawansowane aplikacje .NET

### 4.1 Komponenty .NET

Tworzenie własnych komponentów w ramach platformy .NET

1. zaprojektować prosty komponent wyświetlający jedno, dwu bądź trójkolorowe pola podobne do sygnalizatora świetlnego, komponent powinien posiadać własności do zmiany swojego stanu,
2. zaprojektować komponent będący zegarem o postaci tradycyjnej oraz cyfrowej,
3. zaprojektować komponent/usługę niewizualną udostępniający nową usługę<sup>3</sup> w systemie Windows (można wykorzystać projekt typu Windows Service (.NET Framework)).

### 4.2 Urządzenia wejścia

Dostęp do urządzeń wejścia

1. opracować program prezentujący stan urządzenia wejścia manipulator do gier tj. joystick, zadanie można zrealizować za pomocą standardowego GUI, np. Windows Forms, i obiektu Canvas/Graphics/System.Drawing.Drawing2D, bądź środowiska Unity,
2. opracować program prezentujący stan urządzenia wejścia manipulator do gier tj. kierownica, zadanie można zrealizować za pomocą standardowego GUI, np. Windows Forms, i obiektu Canvas/Graphics/System.Drawing.Drawing2D, bądź środowiska Unity.

### 4.3 Pliki XML

Zagadnienia dotyczące przetwarzania plików XML:

1. dla klasy osoba, podać przykłady dotyczące zagadnienia serializacji danych obiektów typu osoba,
2. zrealizować prostą osobową bazę danych, zapis danych ma być wykonywany do pliku w formacie XML, przygotować odpowiedni opis DTD lub XML Schema,
3. opracować prostą aplikację do weryfikacji poprawności pliku XML w oparciu o DTD lub XML Schema,
4. napisać program dla Windows.Forms wczytujący plik XML zawierający dane osobowe i wyświetlający je w postaci drzewa (kontrolka TreeView).
5. wykorzystać przykład z komponentem ListView i przygotować aplikację drukującą wskazane lub wszystkie elementy z listy. Aplikacja ma także umożliwiać skorzystanie z podglądu wydruku,
6. przygotować aplikację z interfejsem MDI prezentującą pliki graficzne np.. formacie jpg lub png, czy też bmp.

---

<sup>3</sup>Podstawowe informacje dot. implementacji usług w C#: <https://learn.microsoft.com/pl-pl/dotnet/framework/windows-services/>

## 5 Język F#

Zadania dotyczące podstaw języka funkcyjnego F#:

1. proste rekurencje, słowo kluczowe `match` – podać przykłady implementacji kilku typowych funkcji rekurencyjnych w języku F#,
2. klasa reprezentująca ułamki zwykłe oraz zestaw przeciążonych operatorów,
3. klasa reprezentująca liczby zespolone oraz zestaw przeciążonych operatorów,
4. podać klasę w F# oraz zestaw przeciążonych operatorów do realizacji kwaternionów,
5. zbudować prosty odbierający zgłaszane zadania i dokonujący ich przetwarzania wykorzystujący klasę F# `MailboxProcessor`,
6. zbudować analizator leksykalny i gramatyczny podstawowych wyrażeń arytmetycznych.

## 6 Aplikacje ADO.NET/Entity Framework

Utworzone bazy danych nie muszą współpracować z serwerem danych Microsoft SQL Server (MSS), można wybrać dowolny inny serwer bądź system bazodanowy (np. SQLite, pakiet ten również bardzo dobrze współpracuje z Entity Framework), przy czym wsparcie dla MSS w przypadku Visual Studio jest najwygodniejsze i cechuje się dość wysokim zautomatyzowaniem:

1. zaprojektować podstawowe elementy bazy danych typu książka telefoniczna, dopisanie nowej osoby, wyszukiwanie osób, w bazie ma być też obecne relacje, tj. jeden do jeden, czy wiele-do-wielu,
2. zaprezentować działanie systemu migracji na przykładzie rozbudowany poprzedniego zadania,
3. podać przykłady dla osobowej bazy danych, które wykorzystują zapytania LINQ dla wyszukiwania danych,
4. zbudować prostą bazę danych do korespondencji seryjnej w oparciu o adres e-mail, wysyłającej np.: życzenia urodzinowe bądź imieninowe,

## 7 Aplikacje ASP.NET i WebServices

Zaprojektować następujące aplikacje WEB oraz ASP.NET (można wykorzystać podstawową wersję ASP.NET a także odmianę MVC, bądź aplikacje .NET Blazor). W przypadku usług sieciowych można wykorzystać technologię WCF bądź gRPC:

1. usługa WEB generująca liczby losowe i klient odczytujący wygenerowane wielkości,
2. usługa WEB, która dla podanego miasta, będzie zwracać informacje o aktualnej prognozie pogody,
3. podstawowa strona WWW w ASP.NET, strona tytułowa oraz trzy podstrony,

4. zbudować stronę ASP.NET, wykorzystującą klasy do weryfikacji poprawności danych,
5. jeszcze raz kalkulator ale w wersji dla ASP.NET,
6. prosty system wiadomości z ostatnich chwil, z funkcją administratora dodającego nowe wiadomości,
7. strona pozwalające na wgrywanie na serwer plików wskazanych przez użytkownika/klienta strony,
8. opracować API, w ramach ASP.NET Core udostępniające wzorem FastAPI zestaw punktów końcowych np. do obsługi listy zadań, można opracować dwa rozwiązania oparte o kontroler (bardziej zalecane, jest też dokumentacja w formacie SWAGGER) lub Minimal.API prostsze w implementacji ale mniej elastyczne.