


<b>Informatyka Stosowana</b>		
Laboratorium 1	<i>Wprowadzenie do biblioteki pandas</i>	 <b>POLITECHNIKA BYDGOSKA</b> Wydział Telekomunikacji, Informatyki i Elektrotechniki
Przedmiot	Eksploracyjna analiza danych - laboratorium	
Prowadzący	mgr inż. Gracjan Kątek	

## 1. Wprowadzenie

Pandas to biblioteka do analizy danych w Pythonie, która ułatwia pracę z tabelarycznymi danymi, jak również z danymi szeregowymi. Jest szczególnie popularna wśród analityków danych i naukowców zajmujących się danymi, gdyż umożliwia łatwe i wydajne manipulowanie oraz analizowanie danych strukturalnych (np. tabelarycznych danych z plików CSV lub baz danych).

### Główne cechy Pandas:

- ☑ Obiekty DataFrame i Series: pandas wprowadza dwa główne obiekty do pracy z danymi – DataFrame (odpowiednik tabeli, z wierszami i kolumnami) oraz Series (jednowymiarowy odpowiednik listy lub kolumny z bazy danych). Te struktury umożliwiają łatwe operacje na danych, takie jak filtrowanie, agregacja, indeksowanie czy grupowanie.
- ☑ Efektywna manipulacja danymi: pandas zapewnia wysokowydajne narzędzia do manipulacji danymi, takie jak filtrowanie, sortowanie, wycinanie, indeksowanie, a także operacje arytmetyczne i logiczne.
- ☑ Broadcasting: NumPy obsługuje broadcasting, co oznacza, że może wykonywać operacje między tablicami różnych kształtów, automatycznie dostosowując ich rozmiar w sposób logiczny.
- ☑ Łatwy import i eksport danych: Pandas umożliwia łatwe wczytywanie danych z różnych formatów plików, takich jak CSV, Excel, SQL, JSON czy HDF5, oraz zapisywanie przetworzonych danych do tych formatów.
- ☑ Obsługa brakujących danych: pandas posiada wbudowane narzędzia do identyfikowania i obsługi brakujących wartości, co jest bardzo przydatne podczas pracy z rzeczywistymi zestawami danych, które często nie są kompletne.
- ☑ Grupowanie i agregacja danych: Dzięki funkcji `groupby()`, pandas pozwala na grupowanie danych według wartości w jednej lub kilku kolumnach, a następnie wykonywanie różnych operacji agregujących (np. sumowanie, średnia, liczba wystąpień itp.).



- ☑ Integracja z innymi bibliotekami: pandas integruje się z innymi popularnymi bibliotekami, takimi jak NumPy, matplotlib, scikit-learn czy seaborn, co umożliwia łatwe wykonywanie bardziej zaawansowanych analiz i wizualizacji danych.

Pandas jest niezwykle przydatne w analizie danych, obróbce surowych danych przed ich użyciem w uczeniu maszynowym oraz w codziennej pracy analityków danych, inżynierów i naukowców.

## 2. Zadania do wykonania

### Zadanie 1. Tworzenie obiektów DataFrame i Series

Twoim zadaniem jest utworzenie jednowymiarowego obiektu Series, który będzie zawierał liczby od 1 do 10. Wykorzystaj w tym celu funkcję `pd.Series()` dostępną w bibliotece pandas. Następnie, stwórz obiekt DataFrame, który będzie reprezentował dane o trzech produktach. Każdy produkt powinien mieć przypisane trzy właściwości: nazwę, cenę oraz ilość dostępnych sztuk. Do utworzenia DataFrame skorzystaj z funkcji `pd.DataFrame()`, przekazując do niej dane w formie słownika.

### Zadanie 2. Import danych z pliku CSV

Zaimportuj dane z pliku CSV do obiektu DataFrame za pomocą funkcji `read_csv()`. Plik powinien zawierać dane tabelaryczne, takie jak np. dane o sprzedaży produktów. Po zaimportowaniu danych, wyświetl na ekranie pierwsze 5 wierszy, aby zweryfikować, że dane zostały prawidłowo wczytane. Upewnij się, że plik CSV znajduje się w odpowiedniej lokalizacji, aby Python mógł go odczytać.

### Zadanie 3. Podstawowe operacje na danych

Twoim kolejnym zadaniem jest wykonanie kilku podstawowych operacji na zaimportowanych danych. Na początek wyświetl pierwsze 5 wierszy danych za pomocą metody `head()`, a następnie ostatnie 5 wierszy przy użyciu `tail()`. Kolejnym krokiem jest analiza statystyk opisowych danych liczbowych w zestawie danych. Użyj w tym celu metody `describe()`, aby zobaczyć informacje takie jak średnia, odchylenie standardowe czy minimalne i maksymalne wartości dla poszczególnych kolumn.

### Zadanie 4. Selekcja kolumn i filtrowanie danych

Przygotuj się na selekcję konkretnych kolumn oraz filtrowanie danych. Najpierw wybierz tylko dwie kolumny z zestawu danych: kolumnę zawierającą informacje o cenach oraz kolumnę z

ilością produktów. Następnie, przefiltruj dane tak, aby wyświetlić tylko te wiersze, gdzie cena produktu jest większa niż 10. Wykorzystaj w tym celu operacje na obiekcie DataFrame, takie jak selekcja kolumn oraz filtrowanie przy użyciu warunków logicznych.

#### Zadanie 5. Dodawanie nowych kolumn

W tym zadaniu musisz dodać nową kolumnę do istniejącego DataFrame. Kolumna ta powinna nazywać się „wartość” i będzie wynikiem mnożenia ceny produktu przez ilość dostępnych sztuk. Upewnij się, że obliczenia są poprawne i wyświetl nową kolumnę w połączeniu z istniejącymi danymi, aby zweryfikować jej wartości.

#### Zadanie 6. Grupowanie danych

Teraz musisz pogrupować dane według nazwy produktów, a następnie obliczyć sumaryczną wartość dla każdego z nich. W tym celu użyj funkcji `groupby()`, która pozwala na agregację danych według wybranej kolumny. Wynikiem będzie zestawienie, w którym dla każdego produktu podana będzie suma wartości (czyli iloczyn ceny i ilości).

#### Zadanie 7. Sortowanie danych

Kolejne zadanie polega na posortowaniu danych w DataFrame według kolumny „wartość”. Zastosuj sortowanie w porządku malejącym, tak aby na początku tabeli znajdowały się produkty o najwyższej wartości. W ten sposób łatwo zobaczysz, które produkty generują najwyższą wartość sprzedaży.

#### Zadanie 8. Wycinanie danych

W ostatnim zadaniu będziesz wycinać dane z DataFrame. Wybierz wiersze od 2 do 4 oraz tylko dwie kolumny: „nazwa” i „wartość”. Wykorzystaj w tym celu metody do wycinania danych w pandas, takie jak `loc[]`, aby uzyskać odpowiednie wycinki z tabeli

### 3. Sprawozdanie

Sprawozdanie powinno zawierać:

- ☒ Treść zadania
- ☒ Kod napisanego programu
- ☒ Wynik działania napisanego programu
- ☒ Opis działania programu
- ☒ Wnioski końcowe

Sprawozdanie musi być przesłane w formacie .pdf i zgodne z formatką. Sprawozdanie należy dostarczyć najpóźniej do północy dnia poprzedzającego dzień kolejnych laboratoriów. W przypadku spóźnienia przysługują 2 terminy poprawkowe wskazane przez prowadzącego.