

# WBUDOWANE TYPY DANYCH, OPERATORY, WBUDOWANE FUNKCJE I METODY

## *ĆWICZENIA DO PREZENTACJI* MODUŁ 3

AltKom Akademia S.A., materiały własne

# 1 PODSTAWOWE WBUDOWANE TYPY DANYCH

## ĆWICZENIE 1.1:

### Lata przestępne

#### UMIEJĘTNOŚCI:

- Wykonanie ćwiczenia pozwoli na zapoznanie się z różnymi typami operatorów

#### CELE I ZADANIA:

- Napisz program, określający, czy podany rok jest przestępny
- Wprowadź testowany rok z klawiatury
- Wypisz wynik na ekranie

#### ALGORYTM WYKONANIA:

- Napisz program, określający, czy podany rok jest przestępny
- Wprowadź z klawiatury rok
- Reguła przestępności:

*Rok jest przestępny, jeśli jest wielokrotnością 4, z wykluczeniem lat będących wielokrotnościami 100, chyba, że są wielokrotnościami 400 (te są przestępne)*

- Zastanów się, jakie dwa niezależne warunki muszą być spełnione, aby rok był przestępny?
- Do zapisu warunków logicznych użyj operatora równości `==` oraz różności `!=`
- Zapisz powyższą regułę w postaci pojedynczego wyrażenia logicznego
- Oblicz wyrażenie, a wynik przedstaw na ekranie
- Przetestuj działanie programu dla różnych wartości lat, np.:

rok 2000 – przestępny (podzielny przez 400)

rok 1900 – nieprzestępny (podzielny przez 100 i niepodzielny przez 400)

rok 2020 – przestępny (podzielny przez 4 i niepodzielny przez 100)

rok 2022 – nieprzestępny (niepodzielny przez 4)

## ĆWICZENIE 1.2:

### Stan lokaty

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - korzystania z wybranych typów danych i operatorów
  - interakcji programu z użytkownikiem (wczytywania danych z klawiatury i wypisywania wyników na ekranie)

#### CELE I ZADANIA:

- Napisz program, który wyliczy stan lokaty bankowej przy stałym oprocentowaniu po upływie zadanej ilości lat
- Wczytaj wszystkie niezbędne dane wejściowe z klawiatury
- Wypisz na ekranie końcowy stan na koncie

#### ALGORYTM WYKONANIA:

- Napisz program, który wyliczy stan lokaty bankowej przy stałym oprocentowaniu w skali roku, po upływie zadanej ilości lat (zakładamy, że po upływie każdego pełnego roku odsetki są dopisywane do kwoty wyjściowej)
- Zastanów się, jakie dane wejściowe będą potrzebne?
- Jakich typów będą te dane?
- Wprowadź niezbędne dane z klawiatury
- Pamiętaj o dokonaniu konwersji wczytanych danych na odpowiednie typy
- Napisz wyrażenie wyliczające stan lokaty
- Zastanów się, jak wynik końcowy zaokrąglić do dwóch miejsc w części ułamkowej
- Wypisz wynik na ekranie

**ĆWICZENIE 1.3:****Losowanie lotto****UMIEJĘTNOŚCI:**

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - pracy z dokumentacją
  - wyszukiwania potrzebnych informacji
  - korzystania z wbudowanych bibliotek

**CELE I ZADANIA:**

- Napisz program, który wyliczy szansę wylosowania  $k$  liczb spośród  $n$  różnych liczb (jak w lotto)

**ALGORYTM WYKONANIA:**

- Wczytaj z klawiatury dane wejściowe:
  - $k$  – ilość skreślanych liczb (np. 6)
  - $n$  – całkowita ilość liczb spośród których skreślamy (np. 49)
- Liczbę możliwych kombinacji wyboru  $k$  spośród  $n$  różnych liczb opisuje wzór:

$$\frac{n!}{k! * (n - k)!}$$

gdzie:  $m!$  oznacza silnię liczby  $m$

- Odszukaj dokumentację modułu matematycznego, a następnie funkcję wyliczającą silnię (możesz też posłużyć się prezentacją)
- Podobnie, jak to było w ćwiczeniu 4.1 z modułu 2, dodaj w skrypcie instrukcję importu modułu, a nazwę funkcji poprzedź nazwą modułu i kropką
- Korzystając z funkcji zastosuj powyższy wzór, a wynik przedstaw na ekranie
- Zastanów się, jak to zadanie można rozwiązać prościej korzystając z aktualnych możliwości biblioteki matematycznej

## ĆWICZENIE 1.4:

### BMI – Body Mass Index

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - korzystania z instrukcji warunkowych
  - prezentacji danych na ekranie

#### CELE I ZADANIA:

- Napisz program, który wyliczy wartość indeksu masy ciała (*BMI – Body Mass Index*)
- Wczytaj wzrost (w metrach) i wagę (w kg)
- Wylicz indeks i wypisz diagnozę
- Dodatkowo określ przedział prawidłowej wagi

#### ALGORYTM WYKONANIA:

- Wczytaj z klawiatury:
  - wzrost (w metrach)
  - wagę (w kg)
- Wartość indeksu opisuje wzór:

$$BMI = \frac{waga}{wzrost^2}$$

- W oparciu o wyliczoną wartość postaw diagnozę (wykorzystaj do tego celu instrukcję warunkową):
  - $< 16.00$  – wygłodzenie
  - $16.00$ – $16.99$  – wychudzenie
  - $17.00$ – $18.49$  – niedowaga
  - $18.50$ – $24.99$  – wartość prawidłowa
  - $25.00$ – $29.99$  – nadwaga
  - $30.00$ – $34.99$  – I stopień otyłości
  - $35.00$ – $39.99$  – II stopień otyłości (otyłość kliniczna)
  - $\geq 40.00$  – III stopień otyłości (otyłość skrajna)
- Na ekranie wypisz wartość wyliczonego indeksu oraz diagnozę
- Odszukaj sposób zaokrąglenia wartości indeksu do 2 cyfr w części ułamkowej
- Dodatkowo podaj przedział prawidłowej wagi dla określonego wzrostu

**ĆWICZENIE 1.5:****Wieczny kalendarz****UMIEJĘTNOŚCI:**

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - korzystania z wybranych typów danych i operatorów
  - stosowania instrukcji warunkowych

**CELE I ZADANIA:**

- Napisz program, który pełni rolę wiecznego kalendarza – na podstawie podanej daty (dnia, miesiąca i roku) wyznacza nazwę dnia tygodnia
- Wykorzystaj algorytm opracowany przez Mike'a Keith'a
- Elementy daty wczytaj z klawiatury

**ALGORYTM WYKONANIA:**

- Napisz program, który dla podanej daty zwróci nazwę dnia tygodnia
- Wczytaj z klawiatury: dzień, miesiąc i rok
- Do wyliczenia dnia tygodnia wykorzystaj algorytm opracowany przez Mike'a Keith'a:

Jeśli przyjmiemy, że:

- $d$  oznacza dzień miesiąca (1..31)
- $m$  oznacza miesiąc (1..12)
- $y$  oznacza rok
- $\text{calk}(x)$  oznacza część całkowitą liczby  $x$
- $\text{mod}$  oznacza operator modulo (resztę z dzielenia całkowitego)

to należy obliczyć:

- rok z poprawką:  $z = \text{jeśli } m < 3 \text{ to: } y - 1, \text{ w przeciwnym razie: } y$
- korektę:  $c = \text{jeśli } m < 3 \text{ to: } 0, \text{ w przeciwnym razie: } 2$
- wyrażenie:  
$$(\text{calk}(23 * m / 9) + d + 4 + y + \text{calk}(z / 4) - \text{calk}(z / 100) + \text{calk}(z / 400) - c) \bmod 7$$

Interpretacja wyniku wyrażenia:

- $0$  – niedziela,  $1$  – poniedziałek, ...,  $6$  – sobota

- Na ekranie wypisz datę i nazwę wyliczonego dnia tygodnia
- Do wypisania nazwy dnia tygodnia – podobnie jak w poprzednim ćwiczeniu – użyj instrukcji warunkowej
- Przetestuj działanie programu dla kilku wybranych dat

## 2 ZŁOŻONE TYPY DANYCH

### ĆWICZENIE 2.1:

#### Skrót

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - wykorzystania pętli
  - korzystania z operacji na tekstach

#### CELE I ZADANIA:

- Napisz program, który w oparciu o podaną pełną nazwę utworzy jej skrót, np. zamieni *United Nations Educational, Scientific and Cultural Organization* na *UNESCO*

#### ALGORYTM WYKONANIA:

- Z podanej pełnej nazwy wyodrębnij słowa – wyszukaj odpowiednią funkcję
- Z każdego słowa weź pierwszy znak i upewnij się, że jest dużą literą
  - uwaga: jeśli zmienna *s* reprezentuje tekst, to pierwszy znak tego tekstu można otrzymać za pomocą konstrukcji *s[0]*
- Połącz ze sobą ciąg tych liter tworząc skrót
- Wypisz na ekranie oryginalną pełną nazwę i utworzony skrót

## ĆWICZENIE 2.2:

### Wieczny kalendarz

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - korzystania z wybranych typów danych i operatorów
  - stosowania krotek

#### CELE I ZADANIA:

- Napisz program, który pełni rolę wiecznego kalendarza – na podstawie podanej daty (dnia, miesiąca i roku) wyznacza nazwę dnia tygodnia
- Wykorzystaj algorytm opracowany przez Mike’a Keith’a
- Elementy daty wczytaj z klawiatury

#### ALGORYTM WYKONANIA:

- Zmodyfikuj rozwiązanie ćwiczenia 1.5 bieżącego modułu
- Utwórz krotkę z nazwami dni tygodnia
- Uprość kod usuwając instrukcję warunkową
- Wykorzystaj krotkę do powiązania indeksu krotki z numerem dnia tygodnia wyliczonym przez algorytm
- Zastanów się, jak będzie najprościej – który dzień tygodnia powinien być pierwszym elementem krotki?
- Na ekranie wypisz datę i nazwę wyliczonego dnia tygodnia
- Przetestuj działanie programu dla kilku wybranych dat



## ĆWICZENIE 2.3:

### Liczba “pechowych” dni

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - korzystania z wybranych typów danych i operatorów
  - stosowania pętli

#### CELE I ZADANIA:

- Napisz program, który dla podanego roku wyliczy ile w nim jest piątków 13-tego
- Wykorzystaj algorytm z poprzedniego ćwiczenia
- Testowany rok wczytaj z klawiatury

#### ALGORYTM WYKONANIA:

- Napisz program, który dla podanego roku sprawdzi ile w nim jest piątków 13-tego
- Wczytaj z klawiatury: rok
- Utwórz licznik i wstępnie go zainicjalizuj (jaką wartością?)
- W pętli sprawdź, jakie dni tygodnia wypadały 13-tego dnia kolejnego miesiąca – jeśli był to piątek, to zinkrementuj licznik
- Do sprawdzenia dnia tygodnia wykorzystaj algorytm zaimplementowany w poprzednim ćwiczeniu
- Na koniec przedstaw wynik na ekranie

**ĆWICZENIE 2.4:****Losowanie lotto****UMIEJĘTNOŚCI:**

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - wykorzystania pętli
  - tworzenia zakresów

**CELE I ZADANIA:**

- Napisz program, który wyliczy szansę wylosowania  $k$  liczb spośród  $n$  różnych liczb (jak w lotto)
- W rozwiązaniu nie posługuj się zewnętrznymi modułami
- Wykorzystaj pętle i zakresy

**ALGORYTM WYKONANIA:**

- Podobnie jak w ćwiczeniu 1.3 wczytaj z klawiatury dane wejściowe:
  - $k$  – ilość skreślanych liczb (np. 6)
  - $n$  – całkowitą ilość liczb (np. 49)
- Liczbę możliwych kombinacji wyboru  $k$  spośród  $n$  różnych liczb opisuje wzór:

$$\frac{n!}{k!(n-k)!}$$

gdzie:  $m!$  oznacza silnię liczby  $m$

- Zauważ, że:

$$k! = 1 * 2 * \dots * k$$

zaś  $n!$  dzieli się bez reszty przez  $(n-k)!$  i da się wyliczyć jako iloczyn liczb

$$\frac{n!}{(n-k)!} = (n-k+1) * (n-k+2) * \dots * n$$

- Powyższe iloczyny oblicz wykorzystując pętle i zakresy
- Podziel otrzymane wartości przez siebie i sprawdź, czy otrzymałeś identyczny wynik jak w ćwiczeniu 1.3

## ĆWICZENIE 2.5:

### Robot

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - korzystania z wybranych typów danych i ich funkcji
  - stosowania pętli

#### CELE I ZADANIA:

- Napisz program, który będzie symulował ruch robota w 4 kierunkach (N/E/S/W)
- Program powinien działać w pętli i kontrolować położenie robota
- Po zakończeniu pracy program powinien obliczyć odległość robota w linii prostej od punktu startowego

#### ALGORYTM WYKONANIA:

- Napisz program, który w nieskończonej pętli będzie przyjmował z klawiatury polecenia ruchu dla robota
- Polecenia powinny mieć postać:

*< kierunek > < ile\_krokov >*

np.:

- *N 4* oznacza 4 kroki na północ
  - *W 1* oznacza 1 krok na zachód
  - *S 2* oznacza 2 kroki na południe
  - *E 3* oznacza 3 kroki na wschód, itd.
- Po każdym poleceniu program powinien wypisywać aktualne współrzędne położenia robota
- Wciśnięcie klawisza <Enter> powinno być sygnałem do zakończenia programu (i opuszczenia nieskończonej pętli)
- Na koniec program powinien obliczyć w jakiej odległości od punktu startowego znalazł się robot (możesz wykorzystać odpowiednią funkcję z biblioteki matematycznej)

## ĆWICZENIE 2.6:

### Skrót

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - korzystania z operacji na tekstach
  - budowy wyrażeń listowych

#### CELE I ZADANIA:

- Napisz program, który w oparciu o podaną pełną nazwę utworzy jej skrót, np. zamieni *United Nations Educational, Scientific and Cultural Organization* na *UNESCO*

#### ALGORYTM WYKONANIA:

- Zmodyfikuj rozwiązanie ćwiczenia 2.1
- Wykorzystując ten sam algorytm uprość rozwiązanie stosując “podejście pythonowe”
- Usuń ze skryptu pętlę i zastąp ją wyrażeniem listowym
- Sprawdź poprawność rozwiązania
- Wypisz na ekranie oryginalną pełną nazwę i utworzony skrót

## ĆWICZENIE 2.7:

### Palindrom

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - wykorzystania wycinków sekwencji
  - korzystania z operacji na tekstach

#### CELE I ZADANIA:

- Napisz program, który zweryfikuje, czy podany tekst jest palindromem
- Palindrom brzmi tak samo, niezależnie od tego, czy jest czytany od przodu, czy wspak
- Wielkość liter, znaki separatorów i znaki interpunkcyjne nie mają znaczenia

#### ALGORYTM WYKONANIA:

- Zapisz w zmiennej treść tekstu z potencjalnym palindromem
- Palindrom brzmi tak samo, niezależnie od tego, czy jest czytany od przodu, czy wspak
- Do testów możesz użyć następującego tekstu: *Co mi dał duch? Cud, ład i moc.*
- Na podstawie powyższego tekstu utwórz nowy tekst zawierający jedynie litery (wszystkie inne znaki usuwamy)
- W tekście należy ujednolicić wielkość liter – zamień wszystkie litery w tekście na litery małe (lub na duże)
- Sprawdź, czy ten tekst i tekst o odwrotnej kolejności liter są identyczne
- Uwaga: do utworzenia tekstu o odwrotnej kolejności znaków możesz użyć wycinków sekwencji
- Możesz “ulepszyć” rozwiązanie testując, czy pierwsza połowa tekstu jest identyczna z drugą połową napisaną wspak
- Wypisz na ekranie wynik porównania

## ĆWICZENIE 2.8:

### Początki kwartałów

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - użycia list
  - wykorzystania pętli
  - wykorzystania wyrażeń listowych

#### CELE I ZADANIA:

- Napisz program, który z podanej krotki nazw miesięcy wybierze te, które są początkami kwartałów
- Wykorzystaj wyrażenie listowe

#### ALGORYTM WYKONANIA:

- Utwórz krotkę zawierającą nazwy miesięcy
- Napisz wyrażenie listowe, które zwróci nową listę z nazwami pierwszych miesięcy w każdym kwartale
- Wypisz zawartość tej listy na ekranie
- Zastanów się, jak wypisać elementy listy, separując je przecinkami, bez użycia pętli
- Jak inaczej można rozwiązać to zadanie bez użycia wyrażeń listowych?

## ĆWICZENIE 2.9:

### Kwartały

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - użycia list
  - wykorzystania pętli
  - wykorzystania wyrażeń listowych

#### CELE I ZADANIA:

- Napisz program, który z podanej krotki nazw miesięcy utworzy nową listę, w której miesiące będą pogrupowane w kwartały
- Wykorzystaj wyrażenie listowe

#### ALGORYTM WYKONANIA:

- Utwórz krotkę zawierającą nazwy miesięcy
- Napisz wyrażenie listowe, które zwróci nową listę, zawierającą 4 listy z nazwami miesięcy kolejnych kwartałów (macierz 4x3)
- Wypisz zawartość tej listy na ekranie
  
- Zwróć uwagę, że jeśli skrypt napiszesz uniwersalnie nie kodując w nim informacji o tym, że mamy 4 kwartały, a każdy z nich zawiera 3 miesiące, to można go będzie wykorzystać do prezentacji dowolnych danych w formie tabeli
- Opcjonalnie, spróbuj ulepszyć w ten sposób swój skrypt i wykorzystaj go do pogrupowania miesięcy w półrocza (sprawdź także, czy działa z kwartałami)

## ĆWICZENIE 2.10:

### Liczby rzymskie

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - definiowania i wykorzystania słowników
  - użycia instrukcji sterujących

#### CELE I ZADANIA:

- Napisz program, który podaną liczbę zapisze za pomocą cyfr rzymskich

#### ALGORYTM WYKONANIA:

- Zdefiniuj słownik, którego kluczami będą liczby arabskie, a wartościami ich odpowiedniki rzymskie (zastanów się, które?)
- Dla przypomnienia: I - 1, V - 5, X - 10, L - 50, C - 100, D - 500, M - 1000
- Klucze powinny być uporządkowane malejąco
- Wczytaj z klawiatury liczbę całkowitą z przedziału między 1 a 3999
- Jeśli wczytana liczba nie mieści się w tym zakresie, to program powinien o tym poinformować
- Korzystając z pętli pomniejszaj liczbę o kolejne wartości zdefiniowane jako klucze słownika (od największych do coraz mniejszych) mieszczące się w liczbie i zapisuj ich odpowiedniki rzymskie
- Uwaga: Algorytm postępowania przypomina problem wydania reszty z użyciem jak najmniejszej ilości nominałów



## ĆWICZENIE 2.11:

### “Baza” osób

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - symulowania kolekcji danych za pomocą słowników oraz list (namiastka bazy danych)
  - wydobywania potrzebnych danych z utworzonej struktury

#### CELE I ZADANIA:

- Za pomocą słowników utwórz dane opisujące kilka osób
- Utwórz z nich listę
- Napisz kilka konstrukcji za pomocą których utworzysz podzbiór osób spełniających zadane kryteria
- Zaproponuj kilka przykładów w których użyjesz funkcji agregujących dane (np. suma, średnia arytmetyczna, minimum, maksimum, liczba elementów)

#### ALGORYTM WYKONANIA:

- Zaprojektuj słownik opisujący osobę
- W tym celu zaproponuj atrybuty charakteryzujące osobę – w słowniku będą one pełniły rolę kluczy (np. imię, nazwisko, adres, płeć, wiek, itp.)
- Zastanów się, jakie typy danych będą powiązane z tymi kluczami (najlepiej, gdyby były różnorodne)
- Za pomocą zaprojektowanych słowników opisz kilka osób
- Zgrupuj te osoby, tworząc na ich podstawie listę
- W ten sposób utworzona została struktura danych przypominająca kolekcję obiektów lub trywialną bazę danych
- W oparciu o tak utworzone dane wejściowe zrealizuj kilka zapytań, np.:
  - utwórz listę mężczyzn mieszkających w podanym mieście
  - utwórz listę osób z danego przedziału wiekowego, np. 20-latków
  - oblicz średnią wieku kobiet o imionach rozpoczynających się literą A
  - itp.
- Spróbuj wymyśleć kilka przykładów podobnych zapytań i je zrealizować
- Uwaga: w rozwiązaniach mogą być przydatne wyrażenia listowe