

Zadanie 1. Sprawdzanie poprawności wieku

Opis sytuacji:

W programie rejestracyjnym użytkownik podaje swój wiek. Musisz sprawdzić, czy wprowadzona wartość jest poprawna.

Polecenie:

Napisz program, który:

1. Pobiera od użytkownika wiek.
2. Sprawdza, czy wiek jest większy lub równy 0.
3. Jeśli nie – rzuca wyjątek `IllegalArgumentException` z komunikatem: "Wiek nie może być ujemny!".
4. Jeśli wiek jest poprawny, wypisuje komunikat: "Wiek poprawny".
5. Obsłuż wyjątek w bloku `try / catch`, aby program nie zakończył się błędem.

Wymagania techniczne:

- Użyj słowa kluczowego `throw`.
 - Zastosuj `try` i `catch`.
-

Zadanie 2. Kalkulator dzielenia

Opis sytuacji:

Tworzysz prosty kalkulator, który wykonuje dzielenie dwóch liczb wprowadzonych przez użytkownika.

Polecenie:

1. Wczytaj dwie liczby całkowite od użytkownika.
2. Wykonaj dzielenie `a / b`.
3. Obsłuż wyjątek `ArithmeticException`, jeśli druga liczba (`b`) jest równa 0.
4. W bloku `finally` wypisz komunikat "Zakończono obliczenia".

Wymagania techniczne:

- Blok `try` z operacją dzielenia.
- Blok `catch` do obsługi wyjątku dzielenia przez zero.
- Blok `finally`, który zawsze się wykona.

Uwaga:

Spróbuj też dodać komunikat, jeśli dzielenie zakończyło się sukcesem.

Zadanie 3. Konwersja tekstu na liczbę

Opis sytuacji:

Program ma zamienić wpisany tekst na liczbę całkowitą, ale użytkownik może wpisać niepoprawny tekst.

Polecenie:

1. Pobierz od użytkownika dowolny napis.
2. Spróbuj zamienić go na liczbę przy pomocy `Integer.parseInt()`.
3. Jeśli wystąpi `NumberFormatException`, wypisz komunikat "Niepoprawny format liczby!".
4. W bloku `finally` wypisz "Koniec programu".

Wymagania techniczne:

- Użyj `try` / `catch` / `finally`.
 - Obsłuż wyjątek `NumberFormatException`.
-

Zadanie 4. Operacje na tablicy

Opis sytuacji:

Masz program, który pracuje na tablicy z danymi liczbowymi. Użytkownik może przez pomyłkę podać błędny indeks.

Polecenie:

1. Utwórz tablicę `int[]` o długości 5.
2. Poproś użytkownika o podanie indeksu i wartości.
3. Spróbuj zapisać wartość w tym miejscu w tablicy.
4. Obsłuż `ArrayIndexOutOfBoundsException`, jeśli indeks jest spoza zakresu.
5. W bloku `finally` wypisz "Operacja zakończona."

Wymagania techniczne:

- `try` z operacją na tablicy.
 - `catch` do przechwycenia błędu indeksu.
 - `finally` z komunikatem końcowym.
-

Zadanie 5. Program z wieloma wyjątkami

Opis sytuacji:

Tworzysz program, który pobiera dane od użytkownika i wykonuje dzielenie. Użytkownik może jednak popełnić różne błędy.

Polecenie:

1. Wczytaj dwie liczby od użytkownika.
2. Wykonaj dzielenie `a / b`.
3. Obsłuż trzy różne wyjątki:
 - `InputMismatchException` – gdy użytkownik poda tekst zamiast liczby,
 - `ArithmeticException` – gdy `b = 0`,
 - `Exception` – wszystkie inne błędy.
4. Każdy wyjątek powinien wypisywać inny komunikat.
5. W bloku `finally` wypisz "Zamykanie programu..."

Wymagania techniczne:

- Kolejność **catch** od najbardziej szczegółowego do najbardziej ogólnego.
- **finally** zawsze wykonywany.