

Poziom 1 – podstawowe wyjątki

1. Zadanie 1 — Kalkulator reszty z dzielenia

Napisz program, który pobiera dwie liczby całkowite od użytkownika i oblicza resztę z dzielenia.

Obsłuż sytuację, gdy użytkownik poda 0 jako dzielnik.

➤ Użyj: `try/catch`

2. Zadanie 2 — Odczyt znaku z napisu

Napisz program, który pobiera od użytkownika tekst i liczbę (indeks znaku), a następnie wypisuje znak o tym indeksie.

Obsłuż przypadek, gdy użytkownik poda liczbę spoza długości napisu.

➤ Użyj: `try/catch`

3. Zadanie 3 — Sumowanie tablicy

Użytkownik podaje rozmiar tablicy, a następnie elementy.

Program sumuje wartości i wypisuje wynik.

Obsłuż wyjątek, gdy użytkownik wprowadzi dane nie będące liczbą.

➤ Użyj: `try/catch/finally`

Poziom 2 – `throw` i `throws`

4. Zadanie 4 — Sprawdzanie hasła użytkownika

Napisz metodę `sprawdzHaslo(String haslo)`, która wyrzuca wyjątek, jeśli hasło nie zawiera cyfry.

Obsłuż ten wyjątek w metodzie `main`.

➤ Użyj: `throw` (np. `IllegalArgumentException`)

5. Zadanie 5 — Odczyt liczby rzeczywistej z pliku

Napisz metodę `odczytajLiczbeDouble(String plik)`, która zwraca liczbę typu `double` z pliku tekstowego.

Jeśli plik nie istnieje lub nie można odczytać liczby, metoda powinna zgłaszać wyjątek.

➤ Użyj: `throws IOException, try/catch w main`

6. Zadanie 6 — Sprawdzanie temperatury

Napisz metodę `sprawdzTemperature(int temp)`, która wyrzuca wyjątek, jeśli temperatura jest poniżej zera.

W `main` obsłuż wyjątek i wypisz komunikat, że „Temperatura za niska”.

➤ Użyj: `throw i try/catch`

Poziom 3 – praca z plikami

7. Zadanie 7 — Liczenie linii w pliku

Napisz program, który otwiera plik `tekst.txt` i wypisuje liczbę jego linii.

Jeśli plik nie istnieje — wypisz komunikat o błędzie.

➤ Użyj: `try/catch, FileNotFoundException`

8. Zadanie 8 — Zapis danych do pliku

Napisz program, który pobiera z konsoli kilka linii tekstu i zapisuje je do pliku `zapis.txt`.

Obsłuż wyjątki wejścia/wyjścia i pamiętaj o zamknięciu pliku w bloku `finally`.

➤ Użyj: `try/catch/finally, IOException`

Poziom 4 – własne wyjątki i zagnieżdżone bloki

9. Zadanie 9 — Własny wyjątek „ZłaCenaException”

Utwórz klasę wyjątku `ZlaCenaException`, a następnie napisz program, który wczytuje cenę produktu.

Jeśli cena jest mniejsza lub równa 0, rzuć ten wyjątek.

Obsłuż wyjątek w `main` i wypisz komunikat o błędnej wartości.

➤ Użyj: `throw, throws, własny wyjątek`

10. Zadanie 10 — Zagnieżdżony try/catch w operacjach plikowych

Napisz program, który wczytuje liczby z pliku `dane.txt`, a następnie dzieli 100 przez każdą z nich.

Obsłuż:

- brak pliku,
- błędne dane (tekst zamiast liczby),
- oraz dzielenie przez zero.
➤ Użyj: zagnieżdżonych `try/catch` bloków lub wielu `catchów`,
`FileNotFoundException`, `NumberFormatException`,
`ArithmetricException`