Lista zadań - Wyjątki

Zad 1. Zabezpiecz poniższy kod tak, aby przy podaniu błędnych danych wyrzucał wyjątek i wyświetlał komunikat o błędzie.

Zad 2.1. W przestrzeni nazw MathExceptions, stwórz dwie klasy wyjątków z wirtualna metoda what(): DivideByZeroException oraz BothNumbersAreZeroException. Obydwie klasy powinny dziedziczyć z klasy std::exception. Pierwszy wyjątek byłby przeznaczony dla dzielenia przez 0, z kolei drugi, gdy zarówno dzielna i dzielnik są równe 0.

Zad 2.2. Uzupełnij funkcje Division(), by w odpowiednich sytuacjach wyrzucały odpowiednie, wcześniej utworzone wyjątki.

Zad 2.3. W funkcji main zaimplementuj instrukcje try/catch, tak aby była możliwość wykrycia stworzonych wyjątków.

Zad 3.1. Napisać funkcję ThrowSomething, która na podstawie przekazanego do niej typu zwróci wyjątek tego typu wraz z jakaś wartością. Ma ona wykrywać pierwsze 4 typy z enum Type, natomiast w każdym innym wypadku ma wyrzucać tekst "Nieznany typ!".

Zad 3.2. Napisać odpowiednio zagnieżdżone instrukcje try/catch w funkcji ExceptionHandler, tak by najpierw łapała znane typy, a dopiero w ostateczności nieznane typy. W każdym bloku catch ma zostać wypisana nazwa typu oraz wartość rzuconego wyjątku.

Zad 4. Użyć Google do wyszukania, jakich wyjątków należy spodziewać się po nieprawidłowym użyciu funkcji zastosowanych w programie. Zabezpieczyć funkcje main tak, aby program kończył się kodem 0 (prawidłowo). Każde błędne użycie funkcji powinno być zakomunikowane na standardowym wyjściu (cout) za pomocą funkcji what() wyrzucanego wyjątku.