

Obsługa wyjątków

Wyjątek (ang. *exception*) — nietypowa sytuacja pojawiająca się podczas działania programu. Każdy wyjątek w Javie jest obiektem określonego typu, przekazywanym do metody, której działanie doprowadziło do pojawienia się błędu. Metoda może przechwycić wyjątek i wykonać instrukcje zapobiegające przerwaniu działania programu.

Konstrukcja obsługi wyjątków przedstawia się następująco:

```
try { instrukcja }
catch (typ_wyjatku_1 e) { instrukcja_obsługi_wyjatku_1 }
catch (typ_wyjatku_2 e) { instrukcja_obsługi_wyjatku_2 }
...
catch (typ_wyjatku_n e) { instrukcja_obsługi_wyjatku_n }
[finally { instrukcja }]
```

Opcjonalny blok `finally` jest zawsze wykonywany przed powrotem do instrukcji występującej za konstrukcją `try-catch`.

W kodzie programu możemy generować wyjątki, stosując instrukcję `throw`.

Zadanie 1

Podczas wprowadzania danych liczbowych użytkownik może świadomie lub przez pomyłkę wprowadzić ciąg znaków niebędący poprawnie zapisaną liczbą. Spowoduje to przerwanie pracy programu. Napisz program umożliwiający wczytywanie z klawiatury liczby zmiennoprzecinkowej typu `double` przy użyciu metody `nextDouble()` z klasy `Scanner` i reagujący poprawnie na popełniony błąd.



Wskazówka

Należy przechwycić i obsłużyć wyjątek `InputMismatchException` (pakiet `java.util`), a następnie ponowić proces wczytywania liczby.

Zadanie 2

Dane liczbowe ze standardowego wejścia możemy wczytywać w postaci łańcucha znaków, a następnie konwertować je na liczby odpowiedniego typu. Napisz program, który w ten sposób wczyta z klawiatury liczbę zmiennoprzecinkową i zareaguje poprawnie na popełnione błędy.



Wskazówka

Podczas konwersji łańcucha na liczbę może wystąpić wyjątek `NumberFormatException`. Należy przechwycić i obsłużyć ten wyjątek, a następnie ponowić proces wczytywania danych.

Zadanie 3

Napisz program, który odczyta i obliczy sumę pięciu liczb całkowitych, wprowadzonych ze standardowego wejścia. Pomiń liczby zmiennoprzecinkowe i łańcuchy znaków niebędące liczbami.



Wskazówka

Wykorzystaj wyjątki lub informację, że klasa `Scanner` oferuje metody sprawdzające, jaki kolejny token znajduje się w strumieniu. Na przykład metoda `hasNextInt()` sprawdza, czy kolejny token jest liczbą całkowitą typu `int`.

Zadanie 4

Napisz program obliczający odwrotność liczby całkowitej wprowadzonej z klawiatury.

- a) Przechwycić i obsłużyć wszystkie wyjątki, jakie mogą pojawić się podczas wczytywania danych i wykonywania obliczeń.
- b) Zrealizuj zadanie bez obsługi wyjątków.

Zadanie 5

Napisz program obliczający wynik dzielenia z resztą dwóch liczb całkowitych wprowadzonych z klawiatury. Przechwycić i obsłużyć wyjątki, jakie mogą pojawić się podczas wczytywania danych i wykonywania obliczeń.

Klasa opakowująca Angle — miara kąta i funkcje trygonometryczne

Zadanie 1

Utwórz klasę Angle zawierającą jedno pole prywatne `x` typu `double` przechowujące miarę kąta podaną w radianach. W klasie zbuduj konstruktor z jednym parametrem `double` (miara kąta w radianach) i sześć metod zwracających wartości funkcji trygonometrycznych kąta reprezentowanego przez obiekt. Napisz program demonstrujący działanie konstruktora i utworzonych metod.



Oprócz powszechnie znanych funkcji trygonometrycznych *sinus*, *cosinus* *tangens* i *cotangens* w klasie Angle uwzględnimy mniej popularne funkcje *secans* ($\sec x = \frac{1}{\cos x}$) i *cosecans* ($\csc x = \frac{1}{\sin x}$).

Zadanie 1

Dołącz do klasy Angle publiczne metody (`radian()` i `degree()`) zwracające liczbę typu `double`, miarę kąta reprezentowanego przez obiekt wyrażoną w *radianach* i *stopniach*. Napisz program demonstrujący działanie tych metod.

Zadanie 2

Dołącz do klasy Angle publiczną metodę `toString()`, która będzie zwracać łańcuch znaków (o postaci `45°30'15"`) przedstawiający miarę kąta reprezentowanego przez obiekt wyrażoną w stopniach, minutach i sekundach. Napisz program demonstrujący działanie tej metody.



Metoda `toString()` przesłoni metodę o tej samej nazwie, dziedziczoną z klasy `Object`. Metodę tę należy zdefiniować z adnotacją `Override` (`@Override`).

Zadanie 3

Dołącz do klasy `Angle` trzy konstruktory — z jednym, dwoma lub trzema parametrami typu `int` — umożliwiające zbudowanie obiektu reprezentującego kąt o podanej liczbie stopni, stopni i minut lub stopni, minut i sekund. Ogranicz liczbę stopni do zakresu od 0° do 360° , a liczbę minut i sekund do zakresu od 0 do 60. Napisz program demonstrujący działanie tych konstruktorów.

Zadanie 4

Ciąg znaków o postaci `105°30'15"` przedstawia miarę kąta w stopniach, minutach i sekundach katowych. Utwórz konstruktor w klasie `Angle`, który zbuduje obiekt reprezentujący kąt o podanej w ten sposób mierze. Napisz program demonstrujący działanie tego konstruktora.

Zadanie 5

Utwórz w klasie `Angle` sześć metod o nazwie `setOfXxx`, gdzie `Xxx` oznacza nazwę funkcji trygonometrycznej, ustawiających miarę kąta reprezentowanego przez obiekt na podstawie podanej wartości odpowiedniej funkcji (metody te odpowiadają funkcjom odwrotnym do funkcji trygonometrycznych). Metody zmieniają wartość obiektu, ale nie zwracają żadnej wartości. Napisz program demonstrujący działanie tych metod.

Zadanie 6

Na płaszczyźnie z układem współrzędnych często potrzebujemy wyznaczyć miarę kąta nachylenia prostej przechodzącej przez początek układu współrzędnych i punkt $P(x, y)$ do osi OX . Utwórz w klasie `Angle` metodę o nazwie `setOfPoint()`, ustawiającą miarę tego kąta jako wartość obiektu. Napisz program demonstrujący działanie metody `setOfPoint()`.

Zadanie 7

Utwórz w klasie `Angle` konstruktor z dwoma parametrami typu `double`, tworzący nowy obiekt reprezentujący miarę kąta nachylenia prostej przechodzącej przez początek układu współrzędnych i punkt $P(x, y)$ do osi OX . Napisz program demonstrujący działanie tego konstruktora.

Zadanie 8

Zbuduj w klasie `Angle` statyczne funkcje o nazwie `valueOf()`, o różnych parametrach (takich samych jak konstruktory zbudowane w zadaniach 21.1, 21.4, 21.5 i 21.8), zwracające obiekt klasy `Angle` odpowiadający kątowi o podanych parametrach. Napisz program demonstrujący działanie tych metod.

Zadanie 9

Na miarach kątów możemy wykonywać dodawanie i odejmowanie. Dodaj do klasy `Angle` metody `add()` (`sub()`), które będą zwracać nowy obiekt, będący sumą (różnicą) miary kąta reprezentowanego przez obiekt wywołujący metodę i miary kąta w obiekcie podanym jako parametr. Napisz program demonstrujący działanie tych metod.

Zadanie 10

Dołącz do klasy `Angle` statyczne metody `sum()` (`diff()`), które będą obliczać sumę (różnicę) dwóch obiektów podanych jako parametry (miar kątów reprezentowanych przez te obiekty). Wynik powinien być obiektem klasy `Angle`. Napisz program demonstrujący działanie tych metod.

Zadanie 11

Miary kątów można mnożyć (dzielić) przez liczbę. Napisz metodę `mult()` (`div()`), która będzie zwracać nowy obiekt, reprezentujący kąt o mierze będącej iloczynem (ilorazem) miary kąta reprezentowanego przez obiekt i liczby podanej jako parametr (całkowitej lub zmiennoprzecinkowej). Napisz program demonstrujący działanie tych metod.

Zadanie 12

Napisz metodę `prod()` (`quot()`), która będzie zwracać obiekt klasy `Angle`, reprezentujący kąt o mierze będącej iloczynem (ilorazem) miary kąta (obektu) podanego jako parametr i liczby podanej jako drugi parametr. Napisz program demonstrujący działanie tych metod.

Zadanie 13

Dołącz do klasy `Angle` statyczne pola przechowujące stałe wartości obiektów reprezentujących wybrane kąty: `RIGHT_ANGLE` (kąt prosty), `STRAIGHT_ANGLE` (kąt półpełny) i `FULL_ANGLE` (kąt pełny), `RADIAN` (1 radian), `DEGREE` (1 stopień), `ARCMINUTE` (1 minuta katowa) i `ARCSECOND` (1 sekunda katowa). Napisz program pokazujący wartości tych stałych.



Wskazówka

Stałe w Javie są właściwie zmiennymi deklarowanymi ze słowem kluczowym `final`. Wartości stałych w czasie działania programu nie możemy zmienić.

Zadanie 14

Dodaj do klasy `Angle` metodę `compl()` (ang. *complementary angle* — kąt dopełniający), zwracającą obiekt reprezentujący dopełnienie kąta (obektu) podanego jako parametr, i metodę `suppl()` (ang. *supplementary angle* — kąt przyległy), zwracającą obiekt reprezentujący miarę kąta przyległego do kąta (obektu) podanego jako parametr. Napisz program demonstrujący działanie tych metod.

Klasa `Angle` jest już dostatecznie rozbudowana. W kilku kolejnych zadaniach pokażemy przykłady wykorzystania pochodzących z niej obiektów i metod.

Zadanie 15

Utwórz metodę `inputAngle()` z jednym parametrem typu `String` (przeznaczonym do opisania wprowadzanej wielkości), zwracającą obiekt `Angle` reprezentujący kąt o mierze podanej w stopniach. Metoda ta powinna poprawnie interpretować takie dane: liczba całkowita stopni, liczba rzeczywista (ułamek dziesiętny) bez podawania symbolu stopnia oraz ciąg znaków zawierających stopnie, minuty i sekundy katowe z odpowiednimi symbolami jednostek (np. $24^{\circ}30'15''$).

Napisz program rozwiązujący następujące zadanie: W trójkącie równoramiennym kąt przy wierzchołku ma miarę α . Oblicz miarę kąta β przyległego do podstawy tego trójkąta. Sprawdź obliczenia, dodając miary wszystkich kątów wewnętrznych trójkąta.



Wskazówka

Podczas wprowadzania danych symbol stopnia ($^{\circ}$) możesz uzyskać, wpisując `Alt+0176` (przytrzymaj lewy klawisz `Alt` i wprowadź kod `0176` z klawiatury numerycznej). Symbole minuty i sekundy to apostrof i cudzysłów (dostępne bezpośrednio na klawiaturze). Miary kątów spełniają zależność $\alpha + 2\beta = 180^{\circ}$.