5. JAVAの型の変換？

Javaで型の変換をする場合、キャスト演算子を使用します。キャスト演算子を使用すると、一つのデータ型から別のデータ型に変換することができます。

以下は、int型からdouble型に変換する例です。

int num = 10;

double numDouble = (double) num;

このように、変換したいデータ型をカッコで囲んでキャスト演算子を記述し、変換したい値を渡すことで変換ができます。

ただし、注意点としては、データ型が互換性のない場合は、変換できないため、ClassCastExceptionが発生する可能性があります。また、値の範囲が変換後のデータ型に収まらない場合には、変換結果が不正確になる可能性があるため、注意が必要です。

5.1**自動型変換**

Javaでは、自動的に型変換を行う場合があります。これを自動型変換と呼びます。自動型変換は、値の代入や式の評価の際に行われます。

以下は、自動型変換を行う例です。

int num1 = 10;

double num2 = num1; // int型の値がdouble型に自動的に変換される

この場合、int型の変数num1の値が、double型の変数num2に代入されます。int型からdouble型への自動型変換が行われ、num1の値がdouble型に変換されます。

自動型変換の際には、以下のルールがあります。

1. 小さいデータ型から大きいデータ型への変換は自動的に行われる。 例: byte → short → int → long → float → double
2. 大きいデータ型から小さいデータ型への変換は明示的にキャスト演算子を使用する必要がある。 例: double → float → long → int → short → byte
3. オブジェクト型から別のオブジェクト型への変換は、継承関係にあるクラス同士であれば自動的に行われる。 例: Object → String, Integer → Object

自動型変換は、コードを簡潔にするために役立ちますが、意図しない変換が行われてしまう場合もあるため、注意が必要です。

以下の例を見てみましょう。

1 double a = 3.0F;

上記のコードは double 型の変数 a に float 型の値を代入している。 このとき3.0Fの値は自動的にdouble型に型変換が起こる。 これが可能な理由は、double型がfloat型よりも多くの数を表現できるからです。 タイプを変更しても情報の損失が起こらない場合、自動型変換が起こる。

逆に、以下の例ではエラーが発生します。 定数3.0は定数ですが、この定数はdouble型です。 この値を表現範囲が狭いfloatに入れようとするため、エラーが発生します。 (実行)

1 float a = 3.0;

自動型変換の原則は、表現範囲が狭いデータ型から広いデータ型への変換のみが許可されることです。

以下は、自動型変換が行われる規則を示しています。グラフィカル ユーザー インターフェイス が含まれている画像

自動的に生成された説明

byte型はshortにすることができますが、shortはbyte型にすることはできません。 longはfloatにすることができますが、floatはlongにすることはできません。

定数と定数を計算するとどうなりますか？ 次の例を見てみましょう。

int a = 3;

float b = 1.0F;

double c = a + b;

上記の演算は2回の型変換が起こる。 まず、aとbを加算するために、整数aと実数bのいずれかが型変換をしなければならない。 上の図によれば int と float がつくと int が float になるので変数 a に含まれている値 3 は float 型になる。 演算結果はfloat型です。 しかし、この値が含まれる変数Cの型はdoubleです。 floatがdouble型の変数に入れるためには、floatがdoubleに型変換をしなければなりません。 このようにして最終的に型変換された値が変数ｃに含まれた。

5.2 明示的な型変換

Javaでの明示的な型変換は、プリミティブ型または参照型の値を別の型に変換することを指します。これは、Javaのデータ型が異なる場合に必要になります。

以下に、Javaでの明示的な型変換の例を示します。

1)プリミティブ型の変換

// int型をdouble型に変換する例

int num1 = 10;

double num2 = (double) num1;

この例では、int型の変数**num1**をdouble型の変数**num2**に変換しています。**(double)**というキャスト演算子を使用して、num1の値をdouble型に変換しています。

2) 参照型の変換

// Object型をString型に変換する例

Object obj = "Hello";

String str = (String) obj;

この例では、Object型の変数**obj**に"Hello"という文字列を代入し、それをString型の変数**str**にキャストしています。キャスト演算子**(String)**を使用して、objの値をString型に変換しています。

なお、明示的な型変換は、変換元のデータ型と変換先のデータ型が互換性がない場合、実行時に例外が発生する可能性があるため、慎重に使用する必要があります。また、Javaでは自動的に型変換を行う暗黙的な型変換もサポートしているため、必要な場合に限り明示的な型変換を使用するようにしましょう。