【実習２】　変数とデータ型の実習

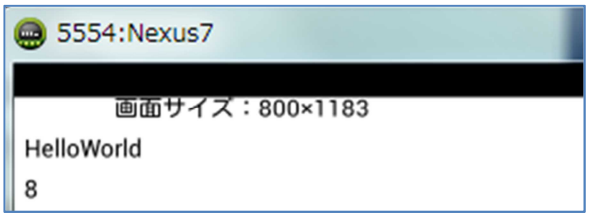
数字（数字の８）を表示します。

１　ソースコード

Hop010View.java の「//【実習 2】変数とデータ型」の箇所へ次のソースを追加入力してください。

```
//【実習２】変数とデータ型
int answerA;  //整数型 int の変数 answerA を確保する。
answerA = 8;  //変数 answerA に数字の 8 を代入する。
canvas.drawText(Integer.toString(answerA), 10, 100, paint);
※I(大文字あい)nteger
```

２　実行結果



数字の８が表示される。

３　解説

3.1 変数とデータ型

変数というのは数字や文字といったデータを入れておく箱です。プログラムには数字や文字といったデータを確保しておく場所が必要です。このデータを確保しておく場所のことを変数といいます。

プログラムは様々な種類のデータを扱います。名簿ソフトの場合は名前、住所、電話番号、年齢などがあります。

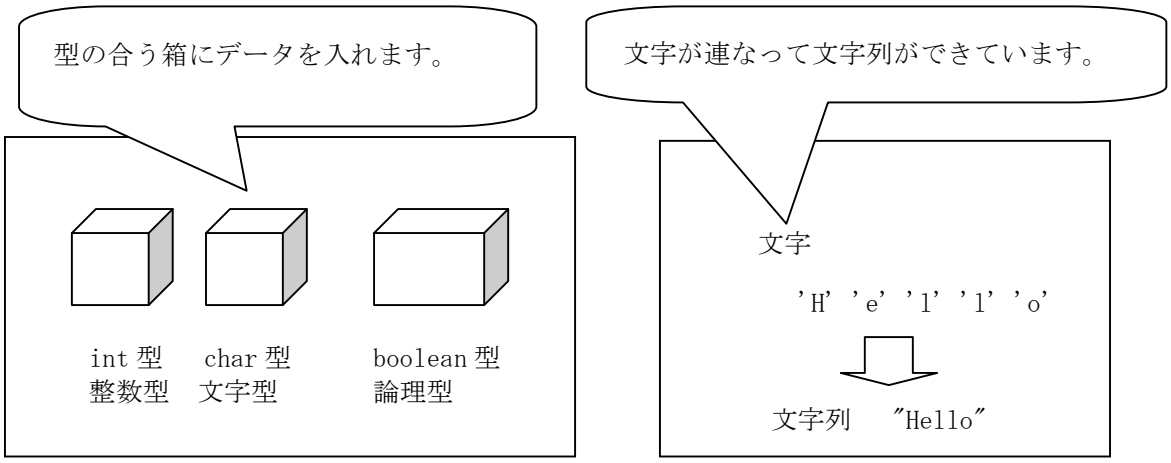
その中で名前、住所は「文字」であり、電話番号、年齢などは「数」です。このようなデータの種類のことをデータ型といいます。データを記憶するための変数はデータ型に合わせて用意してあります。

3.2 基本データ型

型	種類	データ型の基本	データ長	データの範囲
文字	char	文字型(Unicode)	2 Byte	16Bit の文字コード
数値	byte	バイト型	1 Byte	-128 ～ 127
	short	短整数型	2 Byte	-32,768 ～ 32,767
	int	整数型	4 Byte	-2,147,483,648 ～ 2,147,483,647
	long	長整数型	8 Byte	-9,223,372,036,854,775,808 ～9,223,372,036,854,775,807
	float	実数型（単精度浮動小数点型）	4 Byte	$\pm 1.4\times10^{-45}$ ～$\pm 3.4\times10^{38}$
	double	実数型（倍精度浮動小数点型）	8 Bytw	$\pm 4.9\times10^{-324}$～$\pm 1.8\times10^{308}$
論理	boolean	論理型（符号なしデータ）	bit,byte	真（true）、偽（false）

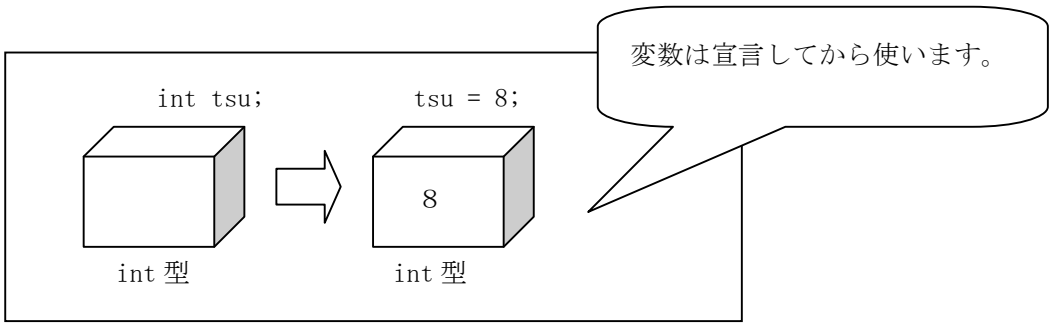
3.3 参照型（アドレス参照型）

型	種類	データ型の基本
	データ型[ ]	配列型
文字	String	文字列型



3.4 変数の宣言

変数を使うためには変数のデータ型と変数名を指定します。このことを変数の宣言といいます。変数の宣言をすることによってコンピュータが変数の準備を行ないます。変数の宣言は「型名　変数名;」で行うことができます。次の図は int 型の変数 tsu の宣言です。



3.5 変数の利用

変数に値を代入するには、等号(=)を利用します。

char 型は、データ長が２バイトの文字型です。文字をシングルクォーテーション(')で囲むことで表現できます。例えば、'和'という１文字を記憶する変数を指定するには、

```
Char Hensu;
Hensu = '和';
```

となります。

「和子」という文字列を記憶する変数を指定するには、String 型を使い、文字をダブルククォーテーション(")で囲み、次のように指定します。

```
String Hensu;
Hensu = "和子";
```

数値の場合、データの種類に応じて、デフォルトでは int 型または double 型となります。long 型の数値は、数値の末尾に L 又は l を、float 型の数値には、F 又は f

を指定します。

3.6 Java の命名規約

命名個所	命名規約	例
クラス Java ファイル	Pascal 形式 大文字で始めて単語の区切りを大文字にする。	PascalCasing、 JavaOnMobilePhone
変数 メソッド	Camel 形式 小文字で始めて単語の区切りを大文字にする。	camelCasing、 javaOnMobilePhone
定数	大文字だけで書いて、単語の区切りを「 _ （アンダーバー）」にする	HANI_FLG

3.7 その他

①canvas.drawText (Integer.toString(answerA) ～

canvas.drawText は、文字を表示する命令です。数字を表示するには、文字に変換する必要があります。Integer.toString(answerA)にすると、answerA という int 型変数を文字に変換します。なお、double 型変数の場合は、Double.toString（変数名）とします。

②canvas.drawText ("3 + 5 = " + answerA ～

（ ）内に文字列から記入した場合は、文字に変換する必要はありませんので、Integer.toString を省略できます。

【演習】

■次のプログラムを作成してください。

プロジェクト名 ： HopPro???? (????：年組席) (既存)

アプリケーション名： HopEx020View

①Hop010View のソースをコピーして、Hop010 の２箇所の文字を、HopEx020View に変更してください。

②テンプレート(Hop010.java) の①の箇所に HopEx020View を入力してください。

【演習21】

次のように携帯画面に出力するコードを作成してください。

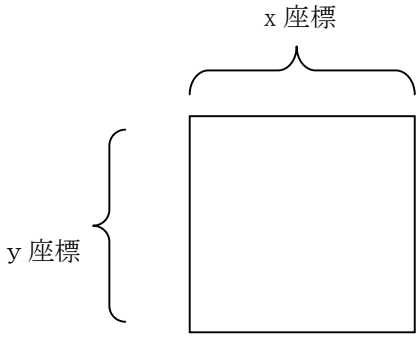
(表示位置 列:10 行:60)

処理：int 型変数 nenrei に 数字の 18 を代入して nenrei を画面表示する。

解説：

(1) canvas.drawText (Integer.toString(nenrei), 10, 60, paint);

※ 10, 60, ・・・ x 座標（列）が 10、y 座標（行）が 60 という意味です。



(2) canvas.drawText は、文字を表示する命令です。

数字を表示するには、文字に変換する必要があります。
Integer.toString(nenrei)にすると、nenrei という int 型変数を文字に変換します。

【演習22】

自分の名前を携帯画面に出力するコードを作成してください。

(表示位置 列:10 行:100)

処理：String 型変数 namae に （自分の名前）を代入して namae を画面表示する。

【演習23】

次のように携帯画面に出力するコードを作成してください。

（表示位置　列:10　行:140）

私の年齢は 18 歳です。

処理：int 型変数 nenrei に 18 を代入して nenrei を画面表示する。

解説：

(1)文字を連結するためには、＋　を使用する。

"私の年齢は、" + Integer.toString (nenrei) + "歳です。"

【演習24】

次のように携帯画面に出力するコードを作成してください。

（表示位置　列:10　行:180）

私の住所は津市です。

処理：String 型変数 address に 津市 を代入して address を画面表示する。

解説：canvas.drawText ("私の住所は" + address + "です。", 10, 180, paint);

【演習25】

次のように携帯画面に出力するコードを作成してください。

（表示位置　列:10　行:220）

円周率は 3.14 です。

処理：double 型変数 atai に 3.14 を代入して atai を画面表示する。

解説：double 型変数を文字型に変換するには、Double.toString (ensyuritsu1)とする。（　）内に文字列から記入した場合は、Double.toString を省略できます。