

4 クラスの活用2

右の画面を表示するように配列を使ってプログラミングしています。

4.1 クラスの活用2のソースリスト

4.1.1 テンプレート(MainActivity.java) の①の箇所に Main030View を入力してください。

4.1.2 Main020View.java をコピーして
Main030View.java を新規作成してください。

ファイル名: src/jp/edu/mie/ Main030View.java

```
package jp.edu.mie;
. . .
public class Main030View extends View
{
    public Main030View(Context context)
    {
        . . .
    }
    protected void onDraw(Canvas canvas)
    {
        . . .
        canvas.drawText("画面サイズ: " + getWidth()
            + "×" + getHeight(), getWidth()/2, 40, paint);
```

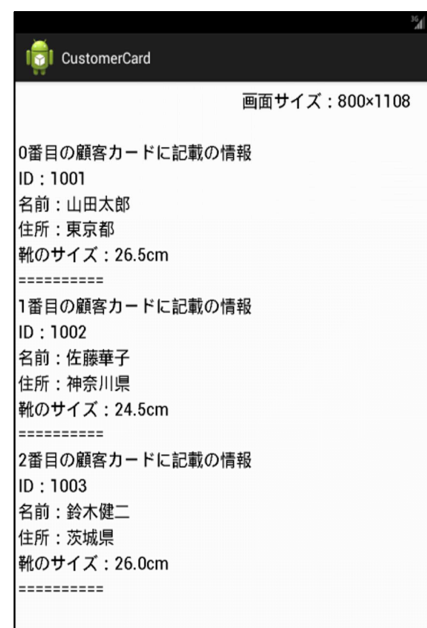
CustomerCard[] cards = new CustomerCard[100];//①配列の作成

```
cards[0] = new CustomerCard(); //CustomerCard クラスのインスタンス生成
cards[0].id = 1001;
cards[0].name = "山田太郎";
cards[0].address = "東京都";
cards[0].shoeSize = 26.5;
```

```
cards[1] = new CustomerCard(); //CustomerCard クラスのインスタンス生成
cards[1].id = 1002;
cards[1].name = "佐藤華子";
cards[1].address = "神奈川県";
cards[1].shoeSize = 24.5;
```

```
cards[2] = new CustomerCard(); //CustomerCard クラスのインスタンス生成
cards[2].id = 1003;
cards[2].name = "鈴木健二";
cards[2].address = "茨城県";
cards[2].shoeSize = 26.0;
```

```
canvas.drawText("1 番目の顧客カードに記載の情報", 10, 80, paint);
canvas.drawText("ID: " + cards[0].id, 10, 110, paint);
canvas.drawText("名前: " + cards[0].name, 10, 140, paint);
canvas.drawText("住所: " + cards[0].address, 10, 170, paint);
```



```

canvas.drawText("靴のサイズ：" + cards[0].shoeSize +
                "cm", 10, 200, paint);
canvas.drawText("=====", 10, 230, paint);

canvas.drawText("2 番目の顧客カードに記載の情報", 10, 300, paint);
canvas.drawText("ID：" + cards[1].id, 10, 330, paint);
canvas.drawText("名前：" + cards[1].name, 10, 360, paint);
canvas.drawText("住所：" + cards[1].address, 10, 390, paint);
canvas.drawText("靴のサイズ：" + cards[1].shoeSize +
                "cm", 10, 420, paint);
canvas.drawText("=====", 10, 450, paint);

canvas.drawText("3 番目の顧客カードに記載の情報", 10, 500, paint);
canvas.drawText("ID：" + cards[2].id, 10, 530, paint);
canvas.drawText("名前：" + cards[2].name, 10, 560, paint);
canvas.drawText("住所：" + cards[2].address, 10, 590, paint);
canvas.drawText("靴のサイズ：" + cards[2].shoeSize +
                "cm", 10, 620, paint);
canvas.drawText("=====", 10, 650, paint);
    }
}

```

4.2 クラスの活用2のソースリストの解説

①CustomerCard[] cards = new CustomerCard[100];

CustomerCard クラスの配列で 100 枚分の顧客カードが扱えます。

【演習 2】

クラス of 活用 2 で表示する y 座標を、変数 y を使って、繰り返し命令でプログラミングしてください。

<ヒント> 変化させる箇所

```

for(int i ~ )
{
    if(cards[i] == null){
        break; //参照のない要素があったらループ処理を終了
    }
    canvas.drawText(i + "番目の顧客カードに記載の情報", 10, , paint);
    canvas.drawText("ID：" + cards[ ].id, 10, , paint);
    canvas.drawText("名前：" + cards[ ].name, 10, , paint);
    canvas.drawText("住所：" + cards[ ].address, 10, , paint);
    canvas.drawText("靴のサイズ：" + cards[ ].shoeSize + "cm", 10, , paint);
    canvas.drawText("=====", 10, , paint);
}

```

5 AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="jp.edu.mie"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="
                    android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```