

プログラミング入門2

第3回 クラスとインスタンス(1)

テーマ：クラスとインスタンス

- 本日の進め方
 - 入門1第8回で習った複合データの資料を参考にして、それに新しい概念を拡張する形で学習します。
- 概念説明： 本日のキーワード
 - 複合データ
 - クラスとインスタンス
 - インスタンス変数

(プログラミング入門2の講義全体の中でも特に重要な説明です。)

本日の主な題材

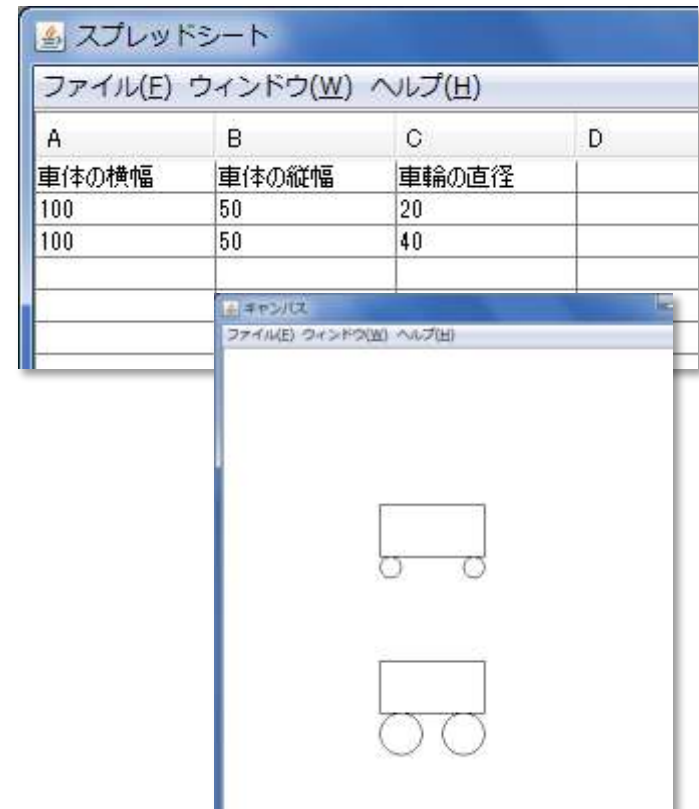
- MyCar のインスタンスを生成して、スプレッドシートやキャンバスに表示せよ。(問題23, 問題35)

car

bodywidth: 100
bodyHeight: 50
wheelDiameter:20

car1

bodywidth: 100
bodyHeight: 50
wheelDiameter: 40



ひとかたまりのデータ

- 「問題文」と「正解」をひとかたまりのデータにするとプログラムから扱いやすくなる

10 + 20 = ?	30
-------------	----

リンゴの色は? (1) 青, (2) 赤	2
----------------------	---

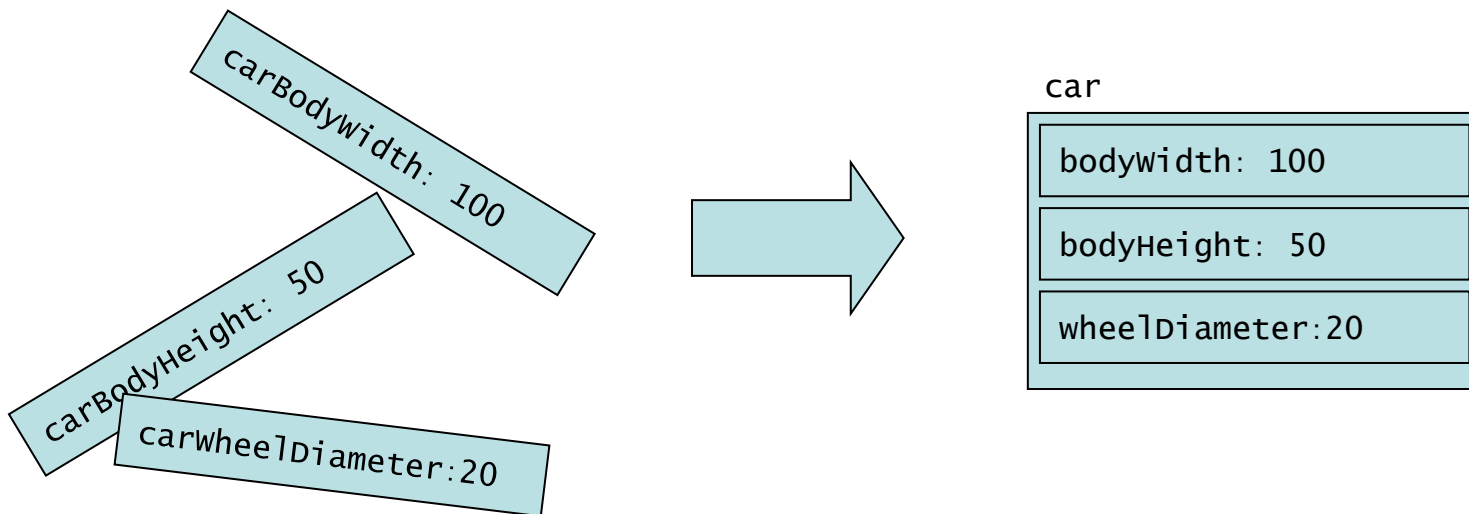
うるう年の1年の日数は?	10
--------------	----

これが今回の「複合データ」

「クイズの問題」という新しいデータ

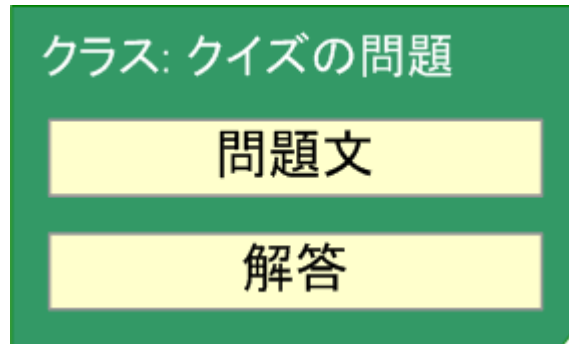
複合データ(Compound Data)

- 複数のデータをひとかたまりのデータとして見なしたものを、複合データ(Compound Data)と呼ぶ。
- 「複数のデータをひとかたまりのデータとして見なす」という考え方は、情報科学で最重要事項の一つである。



「クラス」と形式

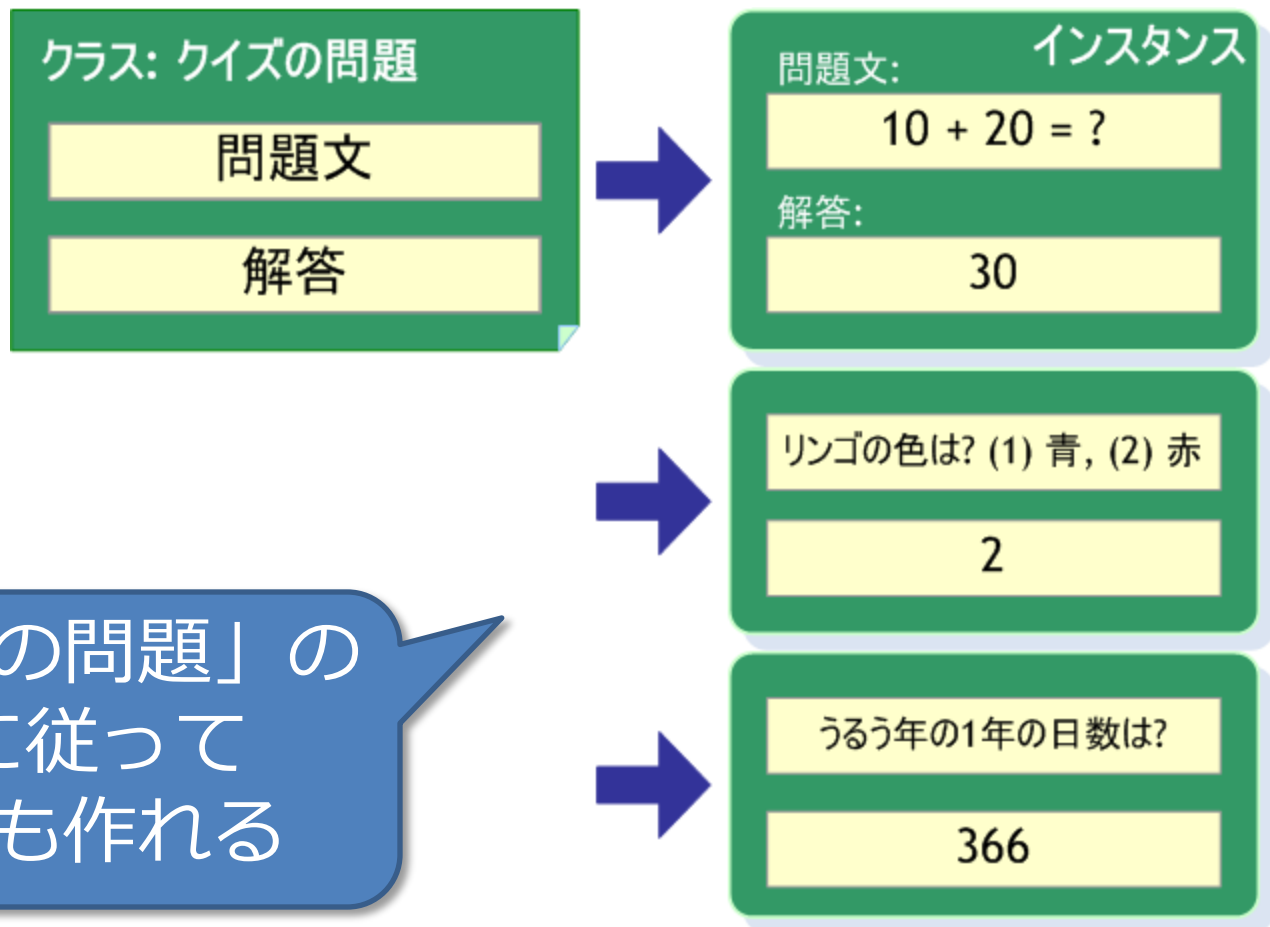
- Javaでは「クラス」という機能を使う
 - 「どのデータを組合せて複合データを作るか」という形式をコンピューターに伝える



「問題文と正解を組合せてクイズの問題を作る」
という形式をJavaのクラスで宣言

「インスタンス」と複合データ

- クラスから「インスタンス」を生成できる
 - クラスの形式に従った複合データ



「クイズの問題」の
形式に従って
いくつも作れる

複合データの使い方

- Javaでは3ステップで複合データを使う
 - クラスで複合データの形式を宣言
 - クラスからインスタンスを生成
 - インスタンスを使って複合したデータを操作

Javaの「クラス」は
複合データの「形式」

Javaの「インスタンス」は
実際の複合データ

Quiz (1)

```
public class Quiz {  
    ... (省略)  
    void start() {  
    }  
}  
  
class Question {  
    String statement;  
    int correct;  
}
```

startメソッドが空なので
まだ実行しても何もしません

最後の } のあとに
さらに書いてください



クラスの宣言

クラスの宣言

```
class <クラスの名前> {  
    <フィールドの宣言1>  
    <フィールドの宣言2>  
    ...  
}
```

複合するデータを
フィールドで宣言

フィールドの宣言

```
<フィールドの型> <フィールドの名前>;
```

1. クラスで複合データの形式を宣言 (クラス宣言)

- 入門2では、クラス宣言をメインプログラムのファイルとは独立して、別ファイルに記述します。(「クラスを宣言する」という)

```
public class Main {  
    public static void main(String[] args) {  
        new Main().start();  
    }  
  
    void start() {  
        ...  
    }  
}
```

メインプログラムファイル

クラス宣言の方にも
public を付ける

```
public class A {  
    型名1 インスタンス変数名1;  
    型名2 インスタンス変数名2;  
    ...  
}
```

クラスファイル

メインプログラム、クラスファイルという複数のファイルを使って、一つのプログラムを作ります。

解説（例題 20より）

下記で説明するクラスMyCarを作成せよ。

- クラスMyCar

クラスMyCarのインスタンスは以下のインスタンス変数を用いて「車」を表現するデータを保持する。

インスタンス変数	初期値	説明
int bodyWidth	100	車の横幅
int bodyHeight	50	車の縦幅
int wheelDiameter	20	車輪の直径

（クラス名 : MyCar）

```
1 package j2.lesson03;  
2  
3 public class MyCar {  
4     int bodyWidth = 100;  
5     int bodyHeight = 50;  
6     int wheelDiameter = 20;  
7 }
```

クラスMyCar
括弧内は初期値

bodywidth: (100)

bodyHeight: (50)

wheelDiameter: (20)

フィールドの初期値を設定したい時は、
フィールド宣言を、
 型名 フィールド名 = 初期値;
の形で記述します。

クラス＝これから作りたい複合データの親玉

インスタンス (生成)

インスタンスの生成

```
<クラスの名前> <変数の名前> = new <クラスの名前>();
```

変数の型に宣言したクラスの
名前を指定する

2. クラスからインスタンスを生成

```
クラス名 変数名 = new クラス名();
```

入門1の考え方

クラス名は型名としても登録されるので、より厳密には、

```
型名 変数名 = new クラス名();
```

入門2の考え方
「型」については、
後半の授業でより
詳しく説明します。

なお、クラスはプログラムを実行した時点に、自動的にロード(インストール)されるので、「クラスを生成する命令」というのは書かなくて良い。

(前スライドで説明した「クラス宣言」さえあればよい)

解説（例題21 より）

問題：クラスMyCarのインスタンスをひとつ作成するプログラムを作成せよ。（なお、実行しても見た目は何も起こらない。）（Ex21Car1）

入門2

```
package j2.lesson03;

3 public class Ex21Car1 {
4     public static void main(String[] args) {
5         new Ex21Car1().start();
6     }
7
8     void start() {
9         MyCar car = new MyCar();
10    }
11 }
```

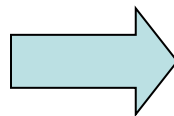
```
1 package j2.lesson03;
2
3 public class MyCar {
4     int bodyWidth = 100;
5     int bodyHeight = 50;
6     int wheelDiameter = 20;
7 }
```

クラスMyCar
括弧内は初期値

bodywidth: (100)

bodyHeight: (50)

wheelDiameter: (20)



car(インスタンス)

bodywidth: 100

bodyHeight: 50

wheelDiameter: 20

MyCar car = new MyCar()

「carはクラスMyCarのインスタンスである」という。

また、このインスタンスは、3つのインスタンス変数bodywidth, bodyHeight, wheelDiameterを持ち、それぞれの変数に初期値が代入される。

解説（例題31より）

問題：MyCarのインスタンスを2つ作成せよ。

例題31

```
1 package j2.lesson03;
2
3 public class Ex31Car2 {
4     public static void main(String[] args) {
5         new Ex31Car2().start();
6     }
7
8     void start() {
9         MyCar car = new MyCar();
10        MyCar car1 = new MyCar();
11    }
12 }
```

入門2

例題31

```
1 package j2.lesson03;  
2  
3 public class MyCar {  
4     int bodyWidth = 100;  
5     int bodyHeight = 50;  
6     int wheelDiameter = 20;  
7 }
```

```
void start() {  
    MyCar car = new MyCar();  
    MyCar car1 = new MyCar();  
}
```

クラスMyCar
括弧内は初期値

bodywidth: (100)

bodyHeight: (50)

wheelDiameter: (20)

MyCar car = new MyCar()

car

bodywidth: 100

bodyHeight: 50

wheelDiameter: 20

MyCar car1 = new MyCar()

car1

bodywidth: 100

bodyHeight: 50

wheelDiameter: 20

20

「carとcar1はクラスMyCarのインスタンスである」
carは、インスタンス変数を3つ持つ。
car1も、インスタンス変数を3つ持つ。

インスタンス（フィールドの使用）

フィールドの使用

<変数の名前>.<フィールドの名前>

代入したり値を取り出したりと
通常の変数をまとめたもののよう使える

あらかじめクラス内に宣言していない
フィールドは使えない

3. インスタンスを使って複合データを操作

(read) 複合データに含まれるデータを取り出す。
(データを読み出す、参照する)→式の中で次の形で用いる。

インスタンス . インスタンス変数名

(write) 複合データに含まれるデータを更新する。
(データを書き込む)→代入文を用いる。

インスタンス . インスタンス変数名 = 更新したいデータ;

入門1では、「フィールド」と呼んだが、
入門2では「インフタンスフィールド」または「インスタンス変数」と呼ぶ

解説 (readの例) 例題22より

問題：（例題21の拡張）作成したMyCarクラスのインスタンスの内容を表示せよ。



A screenshot of a spreadsheet application window titled "スプレッドシート". The menu bar includes "ファイル(E)", "ウィンドウ(W)", and "ヘルプ(H)". The spreadsheet has four columns labeled A, B, C, and D. The first row contains the labels "車体の横幅", "車体の縦幅", and "車輪の直径" in columns A, B, and C respectively. The second row contains the values "100", "50", and "20" in columns A, B, and C respectively. The remaining rows are empty.

A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	20	

（クラス名： Ex22Car1）

※既存のEx21Car1を編集する。

```
1 package j2.lesson03;
2
3 import gpjava.Spreadsheet;
4
5 public class Ex22Car1 {
6     public static void main(String[] args) {
7         new Ex22Car1().start();
8     }
9
10    void start() {
11        MyCar car = new MyCar();
12
13        int width = car.bodyWidth;
14        int height = car.bodyHeight;
15        int diameter = car.wheelDiameter;
16
17        Spreadsheet.show();
18        Spreadsheet.setString(0, 0, "車体の横幅");
19        Spreadsheet.setString(0, 1, "車体の縦幅");
20        Spreadsheet.setString(0, 2, "車輪の直径");
21        Spreadsheet.setInt(1, 0, width);
22        Spreadsheet.setInt(1, 1, height);
23        Spreadsheet.setInt(1, 2, diameter);
24    }
25 }
```



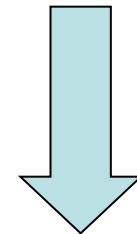
```
void start() {  
    MyCar car = new MyCar();  
  
    int width = car.bodyWidth;  
    int height = car.bodyHeight;  
    int diameter = car.wheelDiameter;  
  
    Spreadsheet.show();  
    Spreadsheet.setString(0, 0, "車体の横幅");  
    Spreadsheet.setString(0, 1, "車体の縦幅");  
    Spreadsheet.setString(0, 2, "車輪の直径");  
    Spreadsheet.setInt(1, 0, width);  
    Spreadsheet.setInt(1, 1, height);  
    Spreadsheet.setInt(1, 2, diameter);  
}
```

car

bodywidth: 100

bodyHeight: 50

wheelDiameter:20



インスタンス . インスタンス変数名

の形。上記はインスタンスに含まれるデータの参照(read)が3回行われている。

スプレッドシート			
ファイル(E) ウィンドウ(W) ヘルプ(H)			
A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	20	

解説(write)の例 (例題24より)

問題：(例題23の拡張) クラスMyCarの
インスタンス変数wheelDiameterの値を更新せよ。
また、そのインスタンスの内容を表示せよ。



A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	40	

(クラス名: Ex24Car1)

※既存のEx23Car1を編集する。

例題24

```
1 package j2.lesson03;
2
3
4 import gpjava.Spreadsheet;
5
6 public class Ex24Car1 {
7     public static void main(String[] args) {
8         new Ex24Car1().start();
9     }
10
11     void start() {
12         MyCar car = new MyCar();
13         car.wheelDiameter = 40;
14
15         int width = car.bodyWidth;
16         int height = car.bodyHeight;
17         int diameter = car.wheelDiameter;
18
19         Spreadsheet.show();
20         Spreadsheet.setString(0, 0, "車体の横幅");
21         Spreadsheet.setString(0, 1, "車体の縦幅");
22         Spreadsheet.setString(0, 2, "車輪の直径");
23         Spreadsheet.setInt(1, 0, width);
24         Spreadsheet.setInt(1, 1, height);
25         Spreadsheet.setInt(1, 2, diameter);
26     }
27 }
28
```

入門2

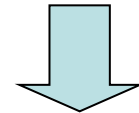
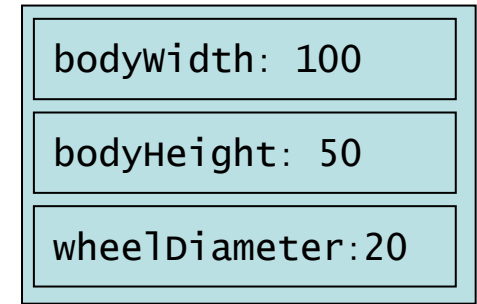
例題24

```
MyCar car = new MyCar();  
car.wheelDiameter = 40;  
  
int width = car.bodyWidth;  
int height = car.bodyHeight;  
int diameter = car.wheelDiameter;  
  
Spreadsheet.show();  
Spreadsheet.setString(0, 0, "車体の横幅");  
Spreadsheet.setString(0, 1, "車体の縦幅");  
Spreadsheet.setString(0, 2, "車輪の直径");  
Spreadsheet.setInt(1, 0, width);  
Spreadsheet.setInt(1, 1, height);  
Spreadsheet.setInt(1, 2, diameter);
```

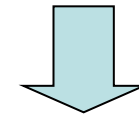
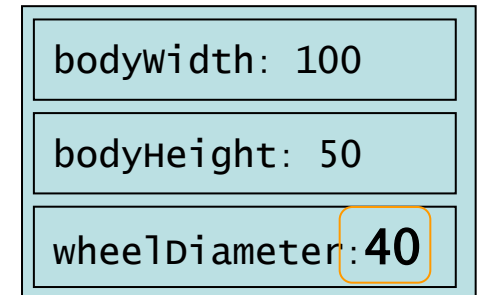
インスタンス.インスタンス変数名=更新したいデータ;

の形。上記ではインスタンスに対して1回の書き込み(write)がある。

car



car



スプレッドシート			
ファイル(E) ウィンドウ(W) ヘルプ(H)			
A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	40	
			28

応用：メソッドの引数に複合データ（インスタンス）をとる。

- メソッドの引数に複合データをとるようにすると、複数のデータをまとめた形で、一気にメソッドに渡すことができる。（もし、個別に渡す場合には、引数の数が増えてしまう。）
- （実際に、渡されるデータはインスタンスへの「参照」であるが、これは次回説明する。）

解説：メソッドの引数に インスタンスをとる（例題32より）

問題：（例題31の拡張）作成したMyCarクラスの2つのインスタンスの内容をそれぞれ表示せよ。次のメソッドを用いよ。

戻り値の型	メソッド名(引数)	機能
void	carData(MyCar aCar, int row)	クラスMyCarのインスタンスの内容を Spreadsheet の row 行に表示する

A	B	C	D	E
車体の横幅	車体の縦幅	車輪の直径		
100	50	20		
100	50	20		

※既存のEx31Car2を編集する。
（クラス名： Ex32Car2）

例題32

```
1 package j2.lesson03;
2
3 import gpjava.Spreadsheet;
4
5 public class Ex32Car2 {
6     public static void main(String[] args) {
7         new Ex32Car2().start();
8     }
9
10    void start() {
11        MyCar car = new MyCar();
12        MyCar car1 = new MyCar();
13        Spreadsheet.show();
14        header();
15        carData(car, 1);
16        carData(car1, 2);
17    }
18
19    void header() {
20        Spreadsheet.setString(0, 0, "車体の横幅");
21        Spreadsheet.setString(0, 1, "車体の縦幅");
22        Spreadsheet.setString(0, 2, "車輪の直径");
23    }
24
25    void carData(MyCar aCar, int row) {
26        int width = aCar.bodyWidth;
27        int height = aCar.bodyHeight;
28        int diameter = aCar.wheelDiameter;
29        Spreadsheet.setInt(row, 0, width);
30        Spreadsheet.setInt(row, 1, height);
31        Spreadsheet.setInt(row, 2, diameter);
32    }
33 }
34
```

入門2

```
void start() {
    MyCar car = new MyCar();
    MyCar car1 = new MyCar();
    Spreadsheet.show();
    header();
    carData(car, 1);
    carData(car1, 2);
}

void header() {
    Spreadsheet.setString(0, 0, "車体の横幅");
    Spreadsheet.setString(0, 1, "車体の縦幅");
    Spreadsheet.setString(0, 2, "車輪の直径");
}

void carData(MyCar aCar, int row) {
    int width = aCar.bodyWidth;
    int height = aCar.bodyHeight;
    int diameter = aCar.wheelDiameter;
    Spreadsheet.setInt(row, 0, width);
    Spreadsheet.setInt(row, 1, height);
    Spreadsheet.setInt(row, 2, diameter);
}
```

car

bodywidth: 100

bodyHeight: 50

wheelDiameter:20

car1

bodywidth: 100

bodyHeight: 50

wheelDiameter:20

スプレッドシート				
ファイル(E) ウィンドウ(W) ヘルプ(H)				
A	B	C	D	E
車体の横幅	車体の縦幅	車輪の直径		
100	50	20		
100	50	20		

戻り値の型	メソッド名(引数)	機能
void	carData(MyCar aCar, int row)	クラスMyCarのインスタンスの内容を Spreadsheet の row 行に表示する

本日のまとめ

- 複合データを利用するための手順 -

1. クラスで複合データの形式を宣言
 - 「複数のデータをひとかたまりのデータとして見なす」という考え方は、情報科学で最重要事項の一つである。
 - 複数のファイルで一つのプログラムを作成する。
2. クラスからインスタンスを生成
3. インスタンスを使って複合データを操作
 - 「操作」とは「データにアクセスすること」であり、readとwriteの2種類。
 - メソッドの引数にインスタンスを指定することもできる。

用語に関する注意事項

- 配列も複数のデータを含んだデータであるが、違う種類(型)のデータはまとめられない。今回紹介する方法は、違う種類(型)のデータもまとめることが出来る方法である。他の言語では「構造体」「レコード」などと呼ばれる。
- なお、配列も「複合データ」の一種である。しかし、本講義において「複合データ」という用語を断りなく用いた場合は、今回紹介したクラスを用いて作成したデータ、つまりインスタンスとして作成されるデータのことを指すこととする。

本日の例題と問題

- MyCar の描画例
 - Ex11, Ex12, Ex13
- 1個のインスタンス作成例
 - Ex20, Ex21, Ex22, Ex23, Ex24, Ex25 , Q10, Q11, Q21
- 2個のインスタンス作成例
 - Ex31, Ex32, Ex33, Ex34, Ex35, Q31, Q32
- 3個のインスタンス作成例
 - Ex41, Ex42, Ex43, Q41, Q42
- 動くボールの描画
 - (Ex50), (Ex51*), (Ex52*), (Q51*), (Q52*)

(Ex:例題, Q:問題, *は少し手間のかかる問題)

各自に適した順番で解けばよいが、上記の順番が自然な流れとなるよう構成されている。

例題集

パッケージ「j2.lesson03」を作成する。

パッケージやクラスの作成, 実行の仕方の説明は省略する。

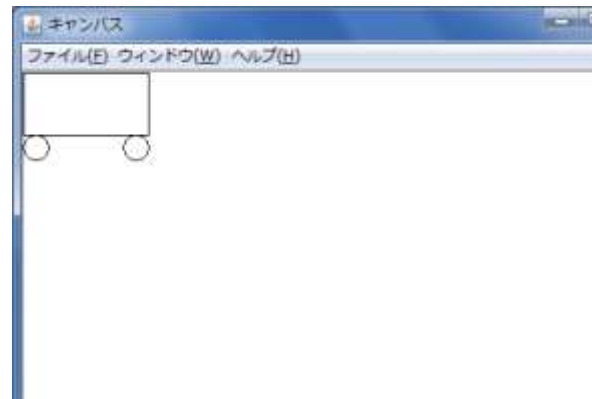
作り方を忘れた場合は過去のスライドや

<http://java2010.cis.k.hosei.ac.jp/01/material-01/>

を参考にせよ

■ 例題11 （基本的な描画機能の紹介）

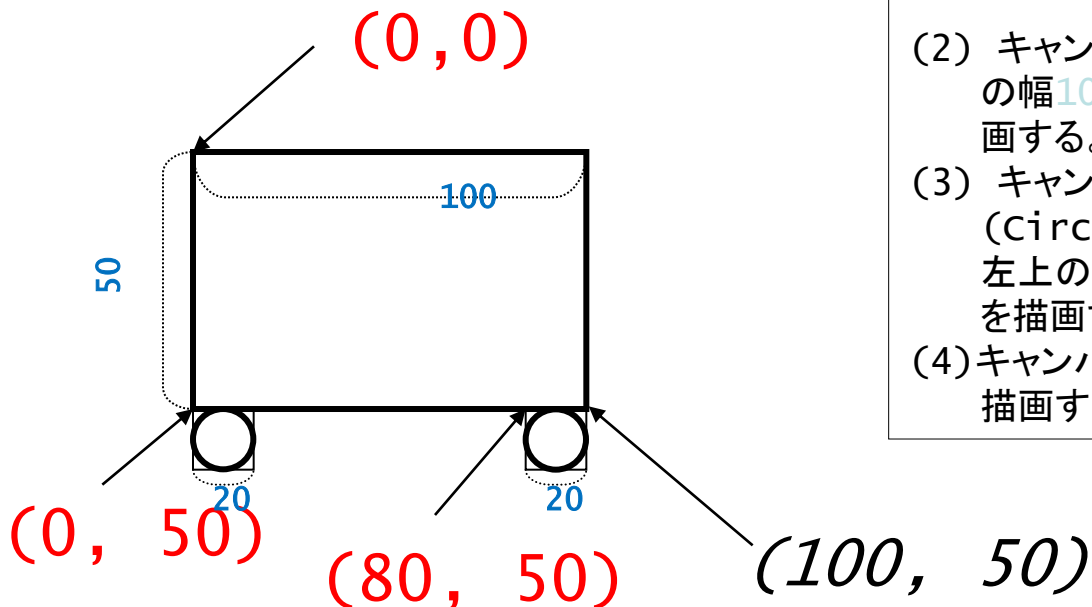
問題: 実行例の「車」を描画し、図形描画の機能とその動作を確かめよ。提供する外部メソッド群には、長方形(rectangle)と、円(circle)を描画する機能が含まれる。



(クラス名 : Ex11DrawCar1)

例題11:

```
1 package j2.lesson03;
2
3 import gpjava.Canvas;
4
5 public class Ex11DrawCar1 {
6     public static void main(String[] args) {
7         Canvas.show();
8         Canvas.drawRect(0, 0, 100, 50); // body
9         Canvas.drawOval(0, 50, 20, 20); // wheel (left)
10        Canvas.drawOval(80, 50, 20, 20); // wheel (right)
11    }
12 }
```



- (1) キャンバス(描画スペース)を作成する。
- (2) キャンバスの座標(0,0)を図形の左上とする、横の幅100、縦の幅50の長方形(Rectangle)を描画する。
- (3) キャンバスの座標(0,50)を基準に直径20の円(Circle)を描画する。(より正確には、(0,50)を左上の座標として縦横20の正方形に内接する真円を描画する。)
- (4) キャンバスの座標(80,50)を基準に直径20の円を描画する。

1つの長方形と、2つの円の組み合わせで車を描く

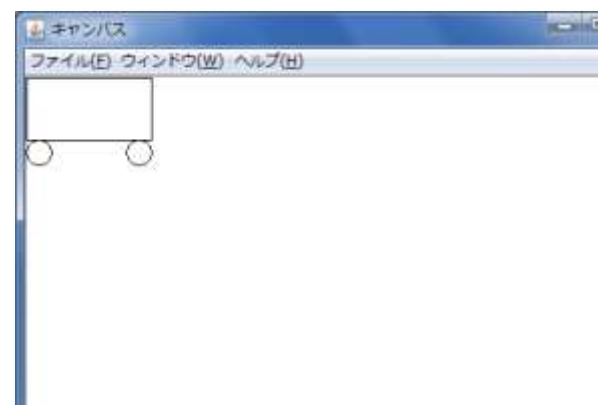
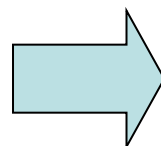
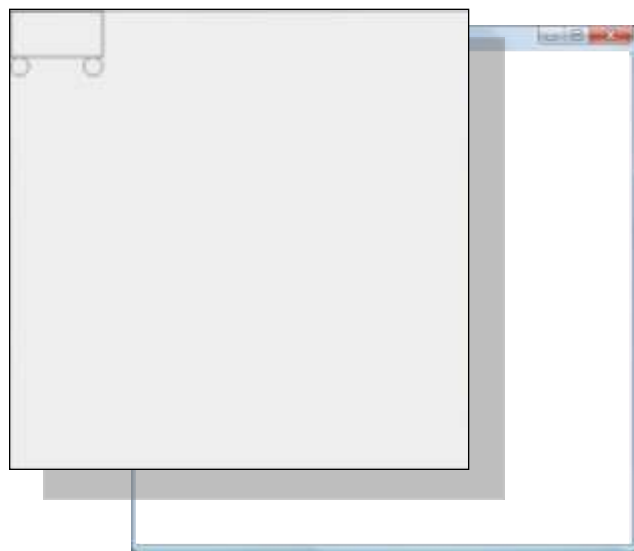
フレームとキャンバス

今回利用する描画機能で必要となる以下の用語を説明する。

キャンバス: 絵を描くことの出来るスペース(場所)であり、いわば画用紙の一枚を表す。

フレーム: キャンバスを固定しておく枠(額縁)である。ディスプレイ上では、フレームをマウスで動かすことで、位置を変えることができる。

キャンバス



フレーム

ピクセル、ドット

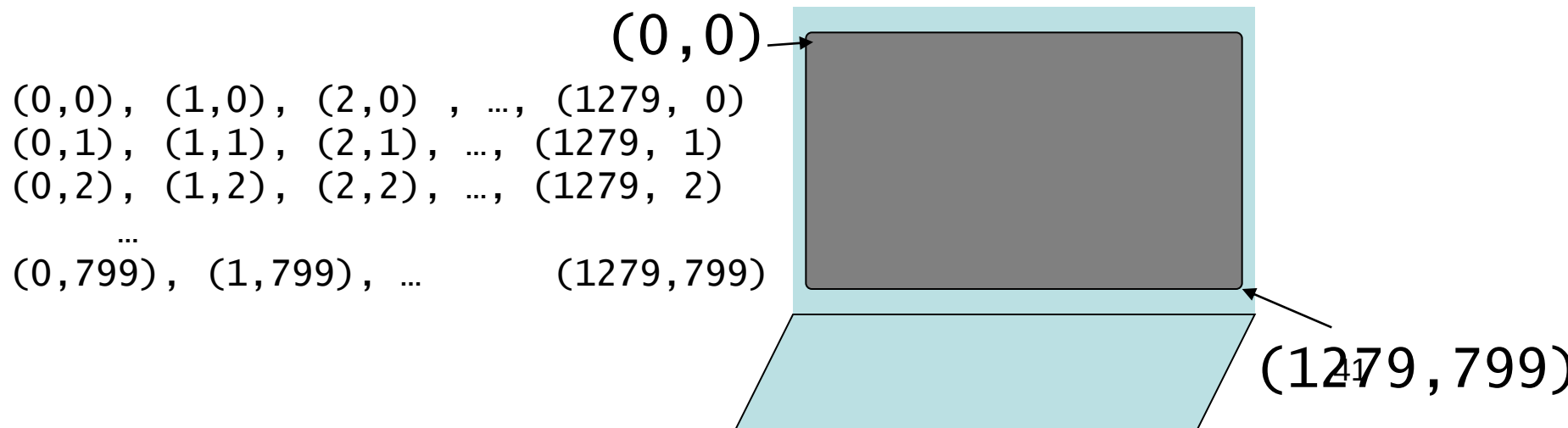
出力装置(ディスプレイ、プリンタ)などの最小表示単位(最も細かい出力要素)は、ドットあるいは、ピクセルと呼ばれる。例えば、あるノートパソコンのディスプレイは横1280ピクセル、縦800ピクセルである。

(これを「ディスプレイの解像度はWXGA」などという)

出力装置の座標系

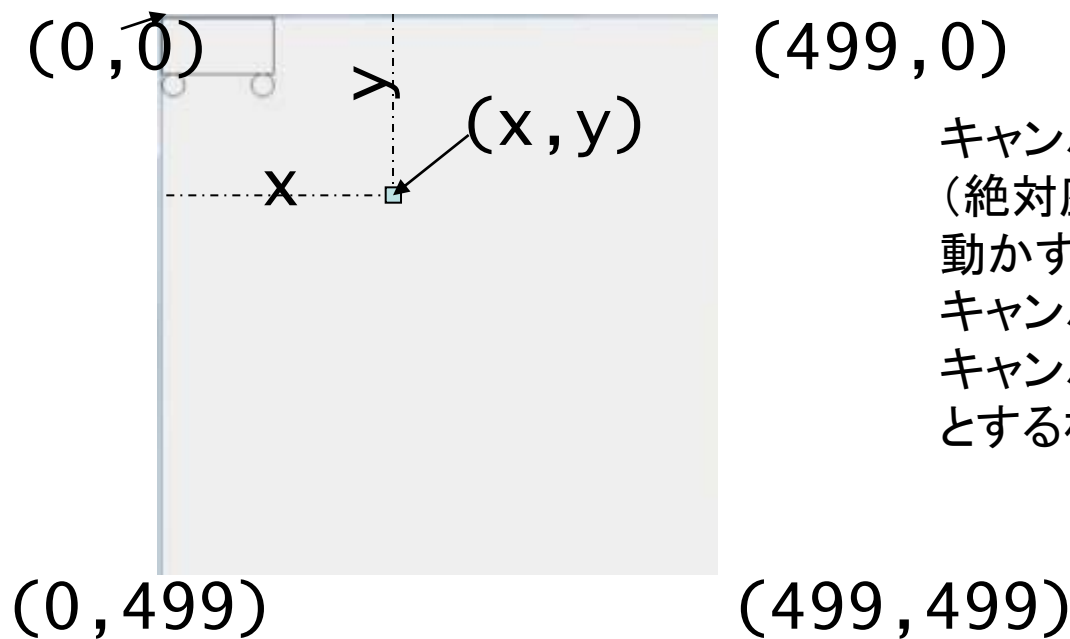
出力装置上のピクセルの一つ一つは座標によって特定できる。

たとえば、WXGAのディスプレイの場合、ピクセル数は 1280×800 個であり、それらを、座標 (x, y) を用いて特定する。たとえば、左上のピクセルの座標は $(0, 0)$ 、右下のピクセルの座標は $(1279, 799)$ である。



キャンバスの座標系

今回は、キャンバスの大きさは横500ピクセル×縦500ピクセルとしている。
キャンバスに図形を描画する際には、キャンバスの座標系を用いてピクセルを指定する。
左上のピクセルは $(0, 0)$ であり、右下は $(499, 499)$ である。左上のピクセルを、「原点」と呼ぶことがある。原点から横 x 、縦 y の位置にあるピクセルは (x, y) で指定する。



キャンバスのディスプレイ上での位置（絶対座標）は、フレームをマウスで動かすことで、自由に変えることができるが、キャンバス上でのピクセル位置の指定は、キャンバスの左上隅の座標を原点 $(0, 0)$ とする相対座標を用いる。

Canvasの機能(1)

返り値	メソッド名(引数)	機能
void	Canvas.show ()	フレーム付きのキャンバス(描画領域)を作成する。
void	Canvas.drawRect (double x, double y, double width, double height)	キャンバスの座標(x,y)を左上として、横width, 縦heightの長方形をキャンバス上に描画する。
void	Canvas.drawOval (double x, double y, double width, double height)	キャンバスの座標(x,y)を起点として、横width, 縦heightの長方形に内接する楕円をキャンバス上に描画する。 (つまりwidthとheightを等しくすると、直径がwidthの円となる。)

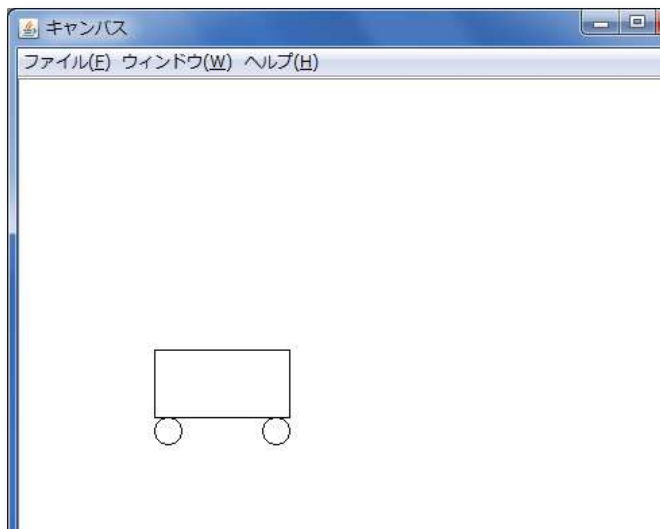
Canvasの機能(2)

返り値	メソッド名(引数)	機能
void	Canvas.setColor(int r, int g, int b)	描画する際の色をRGBの整数値で指定する。
void	Canvas.fillRect (double x, double y, double width, double height)	キャンバスの座標(x,y)を左上として、横width, 縦heightの長方形(内部を塗りつぶしたもの)をキャンバス上に描画する。
void	Canvas.fillOval (double x, double y, double width, double height)	キャンバスの座標(x,y)を起点として、横width, 縦heightの長方形に内接する楕円(内部を塗りつぶしたもの)をキャンバス上に描画する。(つまりwidthとheightを等しくすると、直径がwidthの円となる。)

■ 例題12

問題： Ex11と同じ形の「車」を、好きな座標を起点として描画できるようなメソッドmakeCarを作成せよ。

返り値	メソッド名(引数)	機能
void	makeCar(int x, int y, int bodyWidth, int bodyHeight, int wheelDiameter)	座標(x,y)を起点としてEx11と同じ形の「車」を描く。

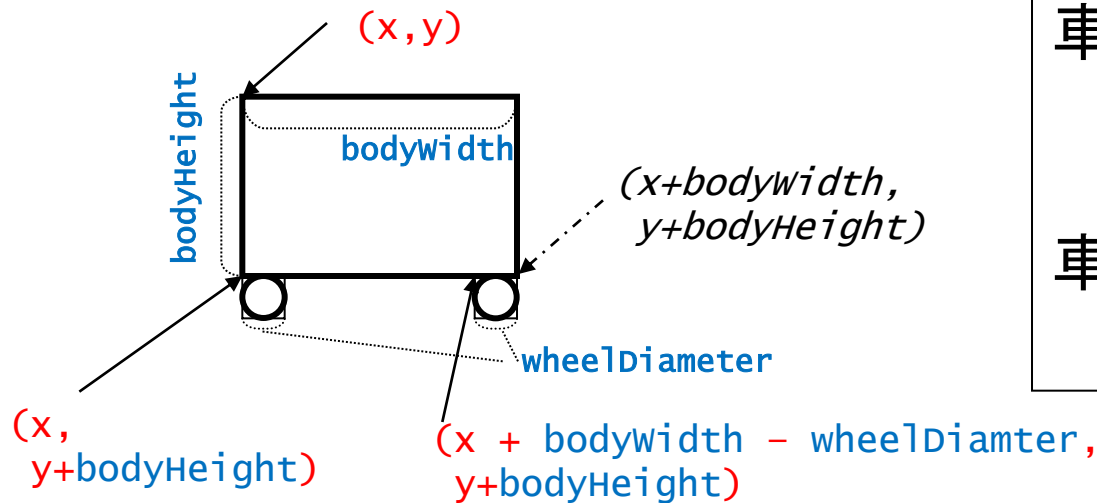


(クラス名 : Ex12MakeCar)

例題12

```
1 package j2.lesson03;
2
3 import gpjava.Canvas;
4
5 public class Ex12MakeCar {
6     public static void main(String[] args) {
7         new Ex12MakeCar().start();
8     }
9
10    void start() {
11        Canvas.show();
12        int x = 100;
13        int y = 200;
14        int bodyWidth = 100;
15        int bodyHeight = 50;
16        int wheelDiameter = 20;
17
18        makeCar(x, y, bodyWidth, bodyHeight, wheelDiameter);
19    }
20
21    void makeCar(int x, int y,
22                int bodyWidth, int bodyHeight, int wheelDiameter) {
23        Canvas.drawRect(x, y, bodyWidth, bodyHeight); // body
24        Canvas.drawOval(x, y + bodyHeight, wheelDiameter, wheelDiameter); // wheel (left)
25        Canvas.drawOval(x+bodyWidth-wheelDiameter, y+bodyHeight,
26                        wheelDiameter, wheelDiameter); // wheel (right)
27    }
28
29 }
```

例題12



車体のサイズ

幅: $bodywidth$

高さ: $bodyHeight$

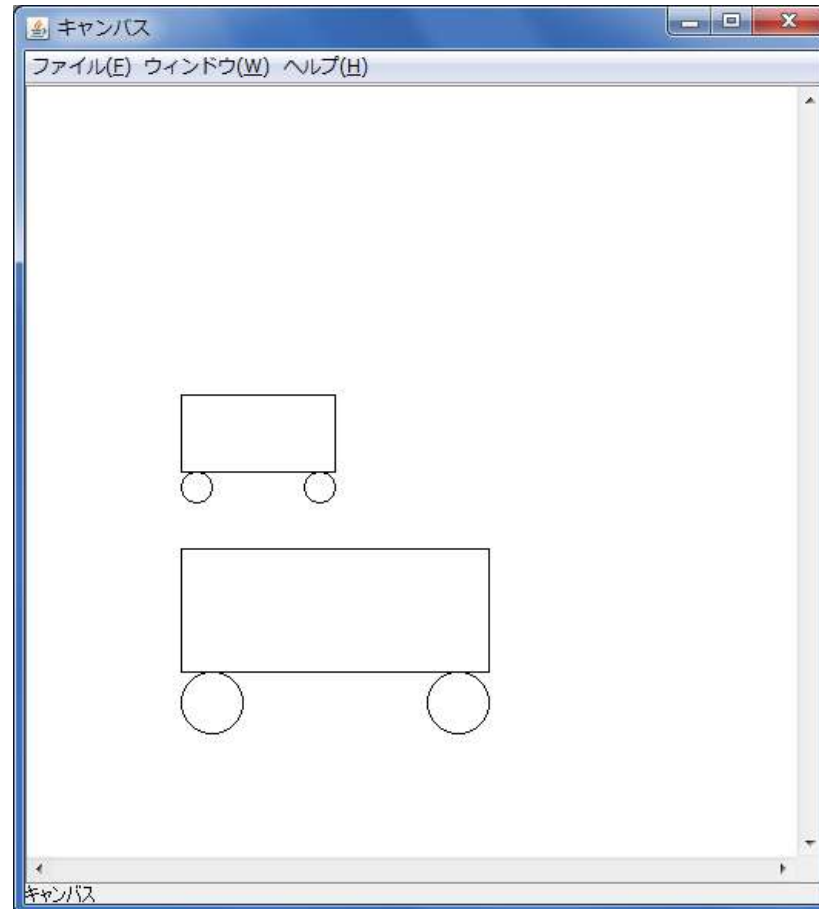
車輪サイズ

直径: $wheelDiameter$

```
21 void makeCar(int x, int y,  
22             int bodyWidth, int bodyHeight, int wheelDiameter) {  
23     Canvas.drawRect(x, y, bodyWidth, bodyHeight); // body  
24     Canvas.drawOval(x, y + bodyHeight, wheelDiameter, wheelDiameter); // wheel (left)  
25     Canvas.drawOval(x + bodyWidth - wheelDiameter, y + bodyHeight,  
26                     wheelDiameter, wheelDiameter); // wheel (right)  
27 }  
28  
29 }
```

■ 例題13

問題：異なるサイズをもつ2台の車を作成せよ。



(クラス名: Ex13MakeCar2)

例題13

```
1 package j2.lesson03;
2
3 import gpjava.Canvas;
4
5 public class Ex13MakeCar2 {
6
7     public static void main(String[] args) {
8         new Ex13MakeCar2().start();
9     }
10
11     void start() {
12         int carBodyWidth = 100;
13         int carBodyHeight = 50;
14         int carWheelDiameter = 20;
15
16         int car1BodyWidth = 200;
17         int car1BodyHeight = 80;
18         int car1WheelDiameter = 40;
19
20         Canvas.show();
21         int x = 100;
22         int y = 200;
23
24         makeCar(x, y, carBodyWidth, carBodyHeight, carWheelDiameter);
25
26         x = 100;
27         y = 300;
28
29         makeCar(x, y, car1BodyWidth, car1BodyHeight, car1WheelDiameter);
30     }
31
32     void makeCar(int x, int y,
33                 int bodyWidth, int bodyHeight, int wheelDiameter) {
34         Canvas.drawRect(x, y, bodyWidth, bodyHeight); // body
35         Canvas.drawOval(x, y + bodyHeight, wheelDiameter, wheelDiameter); // wheel (left)
36         Canvas.drawOval(x+bodyWidth-wheelDiameter, y+bodyHeight,
37                         wheelDiameter, wheelDiameter); // wheel (right)
38     }
39 }
```

■ 例題 20

下記で説明するクラスMyCarを作成せよ。

- クラスMyCar

このクラスのインスタンスは以下のインスタンス変数を用いて「車」を表現するデータを保持する。

インスタンス変数	初期値	説明
int bodyWidth	100	車の横幅
int bodyHeight	50	車の縦幅
int wheelDiameter	20	車輪の直径

(クラス名: MyCar)

例題20

```
1 package j2.lesson03;  
2  
3 public class MyCar {  
4     int bodyWidth = 100;  
5     int bodyHeight = 50;  
6     int wheelDiameter = 20;  
7 }
```

クラスMyCar
括弧内は初期値

bodywidth: (100)

bodyHeight: (50)

wheelDiameter: (20)

クラス＝これから作りたい複合データの親玉

■ 例題21

問題：クラスMyCarのインスタンスをひとつ作成するプログラムを作成せよ。(Ex21Car1)

(クラス名 : Ex21Car1)

```

1 package j2.lesson03;
2
3 public class Ex21Car1 {
4     public static void main(String[] args) {
5         new Ex21Car1().start();
6     }
7
8     void start() {
9         MyCar car = new MyCar();
10    }
11 }

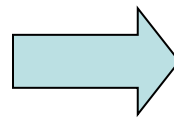
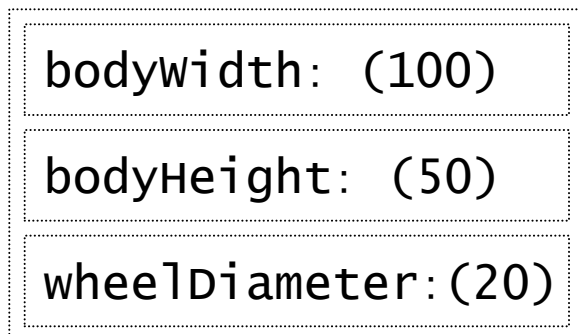
```

```

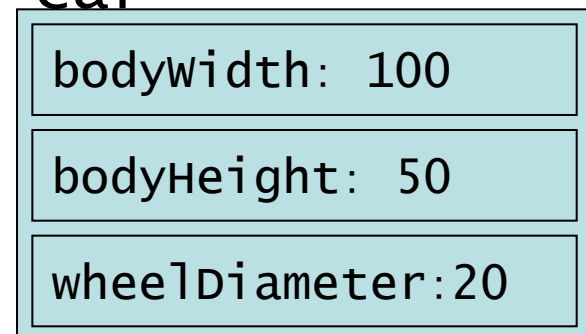
1 package j2.lesson03;
2
3 public class MyCar {
4     int bodyWidth = 100;
5     int bodyHeight = 50;
6     int wheelDiameter = 20;
7 }

```

クラスMyCar
括弧内は初期値



car



MyCar car = new MyCar();

「carはクラスMyCarのインスタンスである」という。
また、このインスタンスは、3つのインスタンス変数bodywidth, bodyHeight, wheelDiameterを持ち、それぞれの変数に初期値が代入される。

■ 例題22

問題：（例題21の拡張）作成したMyCarクラスのインスタンスの内容を表示せよ。



A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	20	

（クラス名： Ex22Car1）

※既存のEx21Car1を編集する。

例題22

```
1 package j2.lesson03;
2
3 import gpjava.Spreadsheet;
4
5 public class Ex22Car1 {
6     public static void main(String[] args) {
7         new Ex22Car1().start();
8     }
9
10    void start() {
11        MyCar car = new MyCar();
12
13        int width = car.bodyWidth;
14        int height = car.bodyHeight;
15        int diameter = car.wheelDiameter;
16
17        Spreadsheet.show();
18        Spreadsheet.setString(0, 0, "車体の横幅");
19        Spreadsheet.setString(0, 1, "車体の縦幅");
20        Spreadsheet.setString(0, 2, "車輪の直径");
21        Spreadsheet.setInt(1, 0, width);
22        Spreadsheet.setInt(1, 1, height);
23        Spreadsheet.setInt(1, 2, diameter);
24    }
25 }
```

例題22

```
MyCar car = new MyCar();

int width = car.bodyWidth;
int height = car.bodyHeight;
int diameter = car.wheelDiameter;

Spreadsheet.show();
Spreadsheet.setString(0, 0, "車体の横幅");
Spreadsheet.setString(0, 1, "車体の縦幅");
Spreadsheet.setString(0, 2, "車輪の直径");
Spreadsheet.setInt(1, 0, width);
Spreadsheet.setInt(1, 1, height);
Spreadsheet.setInt(1, 2, diameter);
```

car

bodywidth: 100

bodyHeight: 50

wheelDiameter:20



インスタンス . インスタンス変数名

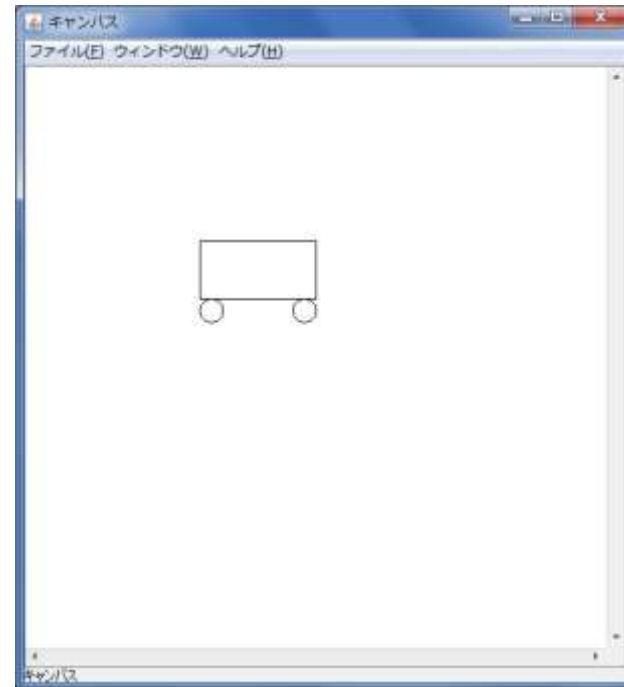
の形。上記はインスタンスに含まれるデータの参照(read)が3回行われている。

スプレッドシート			
ファイル(E) ウィンドウ(W) ヘルプ(H)			
A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	20	

■ 例題23

問題: (例題22の拡張) 作成したMyCarクラスのインスタンスを基にして、キャンバスに「車」を描画せよ

スプレッドシート		
ファイル(E) ウィンドウ(W) ヘルプ(H)		
A	B	C
車体の横幅	車体の縦幅	車輪の直径
100	50	20



(クラス名: Ex23Car1)
※既存のEx22Car1を編集する。

例題23

```
1 package j2.lesson03;
2
3 import gpjava.Spreadsheet;
4 import gpjava.Canvas;
5
6 public class Ex23Car1 {
7     public static void main(String[] args) {
8         new Ex23Car1().start();
9     }
10
11     void start() {
12         MyCar car = new MyCar();
13
14         int width = car.bodyWidth;
15         int height = car.bodyHeight;
16         int diameter = car.wheelDiameter;
17
18         Spreadsheet.show();
19         Spreadsheet.setString(0, 0, "車体の横幅");
20         Spreadsheet.setString(0, 1, "車体の縦幅");
21         Spreadsheet.setString(0, 2, "車輪の直径");
22         Spreadsheet.setInt(1, 0, width);
23         Spreadsheet.setInt(1, 1, height);
24         Spreadsheet.setInt(1, 2, diameter);
25
26         Canvas.show();
27         int x = 150;
28         int y = 150;
29         makeCar(x, y, width, height, diameter);
30     }
31
32     void makeCar(int x, int y,
33                 int bodyWidth, int bodyHeight, int wheelDiameter) {
34         Canvas.drawRect(x, y, bodyWidth, bodyHeight); // body
35         Canvas.drawOval(x, y + bodyHeight, wheelDiameter, wheelDiameter); // wheel (left)
36         Canvas.drawOval(x+bodyWidth-wheelDiameter, y+bodyHeight,
37                         wheelDiameter, wheelDiameter); // wheel (right)
38     }
39 }
```

```

14     int width = car.bodyWidth;
15     int height = car.bodyHeight;
16     int diameter = car.wheelDiameter;
17
18     Spreadsheet.show();
19     Spreadsheet.setString(0, 0, "車体の横幅");
20     Spreadsheet.setString(0, 1, "車体の縦幅");
21     Spreadsheet.setString(0, 2, "車輪の直径");
22     Spreadsheet.setInt(1, 0, width);
23     Spreadsheet.setInt(1, 1, height);
24     Spreadsheet.setInt(1, 2, diameter);
25
26     Canvas.show();
27     int x = 150;
28     int y = 150;
29     makeCar(x, y, width, height, diameter);
30 }
31
32 void makeCar(int x, int y,
33             int bodyWidth, int bodyHeight, int wheelDiameter) {
34     Canvas.drawRect(x, y, bodyWidth, bodyHeight); // body
35     Canvas.drawOval(x, y + bodyHeight, wheelDiameter, wheelDiameter); // wheel (left)
36     Canvas.drawOval(x+bodyWidth-wheelDiameter, y+bodyHeight,
37                     wheelDiameter, wheelDiameter); // wheel (right)
38 }
39 }

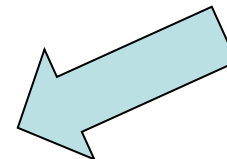
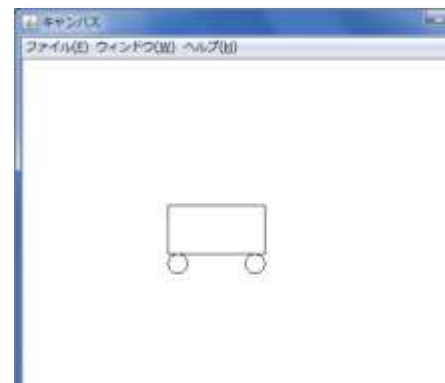
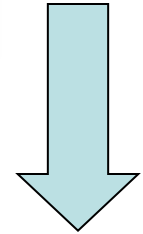
```

car

bodywidth: 100

bodyHeight: 50

wheelDiameter: 20



スプレッドシート		
ファイル(E) ウィンドウ(W) ヘルプ(H)		
A	B	C
車体の横幅	車体の縦幅	車輪の直径
100	50	20

■ 例題24

問題：(例題21の拡張) クラスMyCarの
インスタンス変数wheelDiameterの値を更新せよ。
また、そのインスタンスの内容を表示せよ。



A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	40	

(クラス名: Ex24Car1)

※既存のEx22Car1を編集する。

例題24

```
1 package j2.lesson03;
2
3 import gpjava.Spreadsheet;
4
5 public class Ex24Car1 {
6     public static void main(String[] args) {
7         new Ex24Car1().start();
8     }
9
10    void start() {
11        MyCar car = new MyCar();
12        car.wheelDiameter = 40;
13
14        int width = car.bodyWidth;
15        int height = car.bodyHeight;
16        int diameter = car.wheelDiameter;
17
18        Spreadsheet.show();
19        Spreadsheet.setString(0, 0, "車体の横幅");
20        Spreadsheet.setString(0, 1, "車体の縦幅");
21        Spreadsheet.setString(0, 2, "車輪の直径");
22        Spreadsheet.setInt(1, 0, width);
23        Spreadsheet.setInt(1, 1, height);
24        Spreadsheet.setInt(1, 2, diameter);
25    }
26 }
```

例題24

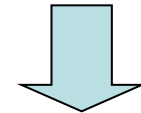
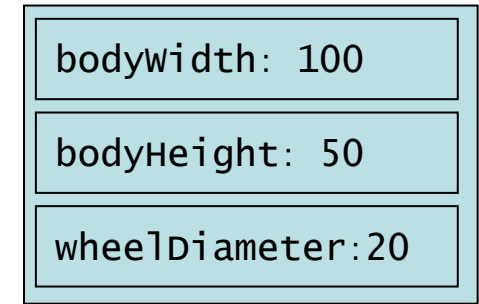
```
MyCar car = new MyCar();  
car.wheelDiameter = 40;  
  
int width = car.bodyWidth;  
int height = car.bodyHeight;  
int diameter = car.wheelDiameter;
```

```
Spreadsheet.show();  
Spreadsheet.setString(0, 0, "車体の横幅");  
Spreadsheet.setString(0, 1, "車体の縦幅");  
Spreadsheet.setString(0, 2, "車輪の直径");  
Spreadsheet.setInt(1, 0, width);  
Spreadsheet.setInt(1, 1, height);  
Spreadsheet.setInt(1, 2, diameter);
```

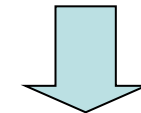
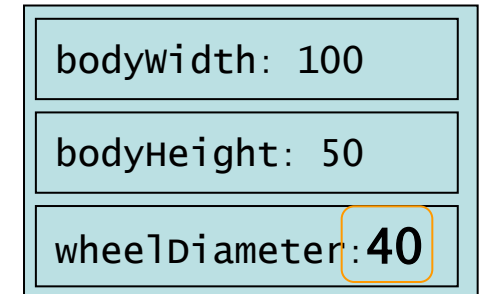
インスタンス.インスタンス変数名=更新したいデータ;

の形。上記ではインスタンスに対して1回の書き込み(write)がある。

car



car

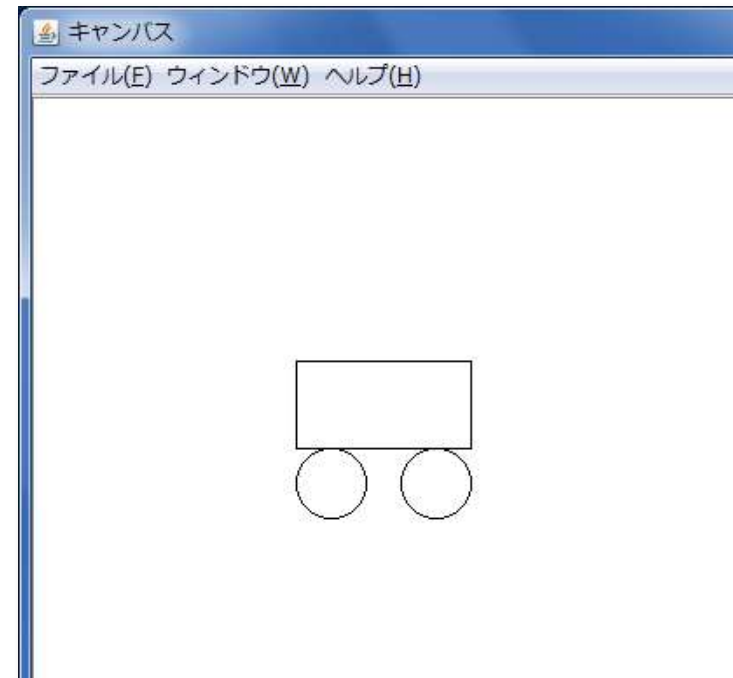


スプレッドシート			
ファイル(E) ウィンドウ(W) ヘルプ(H)			
A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	40	

■ 例題25

問題：(例題23の拡張) 例題24で用意したインスタンスをもとにして、キャンバスに「車」を描画せよ

スプレッドシート			
ファイル(E) ウィンドウ(W) ヘルプ(H)			
A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	40	



(クラス名: Ex25Car1)

※既存のEx23Car1を編集する。

例題25 例題23に13行目を加えたもの。

```
1 package j2.lesson03;
2
3 import gpjava.Spreadsheet;
4 import gpjava.Canvas;
5
6 public class Ex25Car1 {
7     public static void main(String[] args) {
8         new Ex25Car1().start();
9     }
10
11     void start() {
12         MyCar car = new MyCar();
13         car.wheelDiameter = 40;
14
15         int width = car.bodyWidth;
16         int height = car.bodyHeight;
17         int diameter = car.wheelDiameter;
18
19         Spreadsheet.show();
20         Spreadsheet.setString(0, 0, "車体の横幅");
21         Spreadsheet.setString(0, 1, "車体の縦幅");
22         Spreadsheet.setString(0, 2, "車輪の直径");
23         Spreadsheet.setInt(1, 0, width);
24         Spreadsheet.setInt(1, 1, height);
25         Spreadsheet.setInt(1, 2, diameter);
26
27         Canvas.show();
28         int x = 150;
29         int y = 150;
30         makeCar(x, y, width, height, diameter);
31     }
32
33     void makeCar(int x, int y,
34                 int bodyWidth, int bodyHeight, int wheelDiameter) {
35         Canvas.drawRect(x, y, bodyWidth, bodyHeight); // body
36         Canvas.drawOval(x, y + bodyHeight, wheelDiameter, wheelDiameter); // wheel (left)
37         Canvas.drawOval(x+bodyWidth-wheelDiameter, y+bodyHeight,
38                         wheelDiameter, wheelDiameter); // wheel (right)
39     }
40 }
```



```

void start() {
    MyCar car = new MyCar();
    car.wheelDiameter = 40;

    int width = car.bodyWidth;
    int height = car.bodyHeight;
    int diameter = car.wheelDiameter;

    Spreadsheet.show();
    Spreadsheet.setString(0, 0, "車体の横幅");
    Spreadsheet.setString(0, 1, "車体の縦幅");
    Spreadsheet.setString(0, 2, "車輪の直径");
    Spreadsheet.setInt(1, 0, width);
    Spreadsheet.setInt(1, 1, height);
    Spreadsheet.setInt(1, 2, diameter);

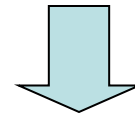
    Canvas.show();
    int x = 150;
    int y = 150;
    makeCar(x, y, width, height, diameter);
}

void makeCar(int x, int y,
             int bodyWidth, int bodyHeight, int wheelDiameter) {
    Canvas.drawRect(x, y, bodyWidth, bodyHeight); // body
    Canvas.drawOval(x, y + bodyHeight, wheelDiameter, wheelDiameter); // wheel (left)
    Canvas.drawOval(x+bodyWidth-wheelDiameter, y+bodyHeight,
                   wheelDiameter, wheelDiameter); // wheel (right)
}

```

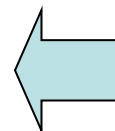
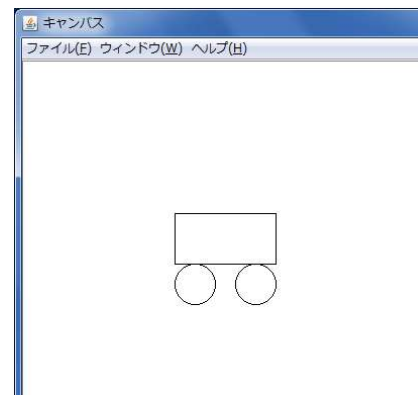
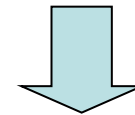
car

bodywidth: 100
bodyHeight: 50
wheelDiameter: 20



car

bodywidth: 100
bodyHeight: 50
wheelDiameter: 40



スプレッドシート			
ファイル(E) ウィンドウ(W) ヘルプ(H)			
A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	40	

■ 例題31

問題: MyCarのインスタンスを2つ作成せよ。

(クラス名: Ex31Car2)

例題31(1)

```
1 package j2.lesson03;
2
3 public class Ex31Car2 {
4     public static void main(String[] args) {
5         new Ex31Car2().start();
6     }
7
8     void start() {
9         MyCar car = new MyCar();
10        MyCar car1 = new MyCar();
11    }
12 }
```

例題31

```
1 package j2.lesson03;  
2  
3 public class MyCar {  
4     int bodyWidth = 100;  
5     int bodyHeight = 50;  
6     int wheelDiameter = 20;  
7 }
```

```
void start() {  
    MyCar car = new MyCar();  
    MyCar car1 = new MyCar();  
}
```

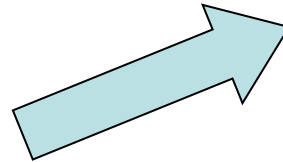
クラスMyCar
括弧内は初期値

bodywidth: (100)

bodyHeight: (50)

wheelDiameter: (20)

MyCar car = new MyCar()



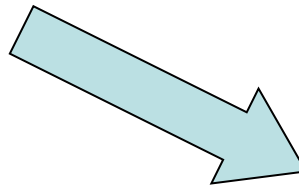
car

bodywidth: 100

bodyHeight: 50

wheelDiameter: 20

MyCar car1 = new MyCar()



car1

bodywidth: 100

bodyHeight: 50

wheelDiameter: 20

「carとcar1はクラスMyCarのインスタンスである」
carは、インスタンス変数を3つ持つ。
car1も、インスタンス変数を3つ持つ。

■ 例題32

問題：（例題31の拡張）作成したMyCarクラスの2つのインスタンスの内容をそれぞれ表示せよ。
（メソッドcarData, headerを用いる）

返り値の型	メソッド名(引数)	機能
void	carData(MyCar aCar, int row)	クラスMyCarのインスタンスの内容をSpreadsheetのrow行に表示する
void	header()	Spreadsheetにヘッダ行を表示する



スプレッドシート				
ファイル(E) ウィンドウ(W) ヘルプ(H)				
A	B	C	D	E
車体の横幅	車体の縦幅	車輪の直径		
100	50	20		
100	50	20		

※既存のEx31Car2を編集する。
（クラス名： Ex32Car2）

例題32

```
1 package j2.lesson03;
2
3 import gpjava.Spreadsheet;
4
5 public class Ex32Car2 {
6     public static void main(String[] args) {
7         new Ex32Car2().start();
8     }
9
10    void start() {
11        MyCar car = new MyCar();
12        MyCar car1 = new MyCar();
13        Spreadsheet.show();
14        header();
15        carData(car, 1);
16        carData(car1, 2);
17    }
18
19    void header() {
20        Spreadsheet.setString(0, 0, "車体の横幅");
21        Spreadsheet.setString(0, 1, "車体の縦幅");
22        Spreadsheet.setString(0, 2, "車輪の直径");
23    }
24
25    void carData(MyCar aCar, int row) {
26        int width = aCar.bodyWidth;
27        int height = aCar.bodyHeight;
28        int diameter = aCar.wheelDiameter;
29        Spreadsheet.setInt(row, 0, width);
30        Spreadsheet.setInt(row, 1, height);
31        Spreadsheet.setInt(row, 2, diameter);
32    }
33 }
```

```

void start() {
    MyCar car = new MyCar();
    MyCar car1 = new MyCar();
    Spreadsheet.show();
    header();
    carData(car, 1);
    carData(car1, 2);
}

void header() {
    Spreadsheet.setString(0, 0, "車体の横幅");
    Spreadsheet.setString(0, 1, "車体の縦幅");
    Spreadsheet.setString(0, 2, "車輪の直径");
}

void carData(MyCar aCar, int row) {
    int width = aCar.bodyWidth;
    int height = aCar.bodyHeight;
    int diameter = aCar.wheelDiameter;
    Spreadsheet.setInt(row, 0, width);
    Spreadsheet.setInt(row, 1, height);
    Spreadsheet.setInt(row, 2, diameter);
}

```

car

bodywidth: 100

bodyHeight: 50

wheelDiameter:20

car1

bodywidth: 100

bodyHeight: 50

wheelDiameter:20

スプレッドシート

ファイル(E) ウィンドウ(W) ヘルプ(H)

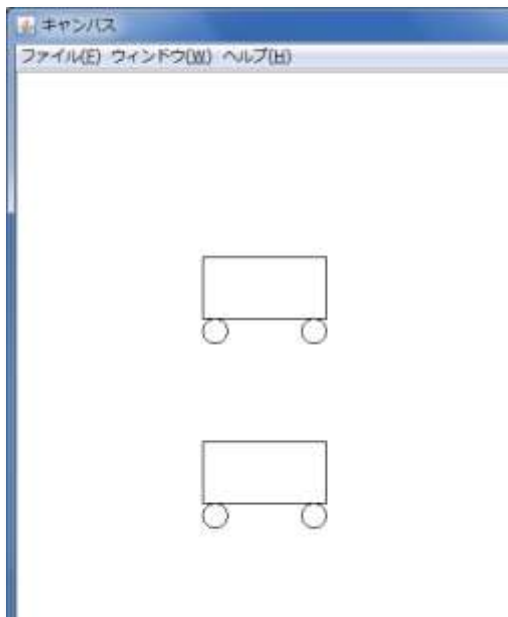
A	B	C	D	E
車体の横幅	車体の縦幅	車輪の直径		
100	50	20		
100	50	20		

戻り値の型	メソッド名(引数)	機能
void	carData(MyCar aCar, int row)	クラスMyCarのインスタンスの内容をSpreadsheetのrow行に表示する

■ 例題33

問題：（例題32の拡張）作成したMyCarクラスの2つのインスタンスをもとにして、キャンバスに「車」を2台描画せよ。（メソッドdrawCarを用いる）

戻り値の型	メソッド名(引数)	機能
void	drawCar(MyCar car, int x, int y)	座標(x,y)を起点として引数MyCarインスタンスに対応した「車」を描く。



（クラス名： Ex33Car2）

※既存のEx32Car2を編集する。

例題33

```
1 package j2.lesson03;
2
3 import gpjava.Canvas;
4 import gpjava.Spreadsheet;
5
6 public class Ex33Car2 {
7     public static void main(String[] args) {
8         new Ex33Car2().start();
9     }
10
11     void start() {
12         MyCar car = new MyCar();
13         MyCar car1 = new MyCar();
14
15         Spreadsheet.show();
16         Canvas.show();
17         header();
18
19         carData(car, 1);
20         drawCar(car, 150, 150);
21         carData(car1, 2);
22         drawCar(car1, 150, 300);
23     }
24
25     void header() {
26
27     }
28
29     void carData(MyCar aCar, int row) {
30
31     }
32
33     void drawCar(MyCar car, int x, int y) {
34         int width = car.bodyWidth;
35         int height = car.bodyHeight;
36         int diameter = car.wheelDiameter;
37         Canvas.drawRect(x, y, width, height); // body
38         Canvas.drawOval(x, y + height, diameter, diameter); // wheel (left)
39         Canvas.drawOval(x+width-diameter, y+height,
40                         diameter, diameter); // wheel (right)
41     }
42
43     }
44
45     }
46
47     }
48
49
50 }
```

+マークは、メソッドを折りたたんでいることを表す。
header(), carData() は、例題32と同じ

```
void drawCar(MyCar car, int x, int y) {
    int width = car.bodyWidth;
    int height = car.bodyHeight;
    int diameter = car.wheelDiameter;
    Canvas.drawRect(x, y, width, height); // body
    Canvas.drawOval(x, y + height, diameter, diameter); // wheel (left)
    Canvas.drawOval(x+width-diameter, y+height,
                    diameter, diameter); // wheel (right)
}
```

例題33

```
void start() {  
    MyCar car = new MyCar();  
    MyCar car1 = new MyCar();  
  
    Spreadsheet.show();  
    Canvas.show();  
    header();  
  
    carData(car, 1);  
    drawCar(car, 150, 150);  
    carData(car1, 2);  
    drawCar(car1, 150, 300);  
}
```

car

bodywidth: 100

bodyHeight: 50

wheelDiameter:20

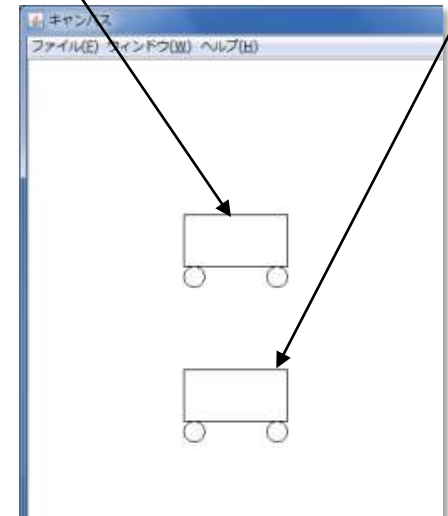
car1

bodywidth: 100

bodyHeight: 50

wheelDiameter:20

```
void drawCar(MyCar car, int x, int y) {  
    int width = car.bodyWidth;  
    int height = car.bodyHeight;  
    int diameter = car.wheelDiameter;  
    Canvas.drawRect(x, y, width, height); // body  
    Canvas.drawOval(x, y + height, diameter, diameter); // wheel (left)  
    Canvas.drawOval(x+width-diameter, y+height,  
                    diameter, diameter); // wheel (right)  
}
```



■ 例題34

問題：（例題32の拡張）クラスMyCarのインスタンスを2つ作成し、2番目に作成したインスタンスのインスタンス変数wheelDiameterの値を40に更新せよ。また、二つのインスタンスの内容を表示せよ。



A screenshot of a spreadsheet application window titled 'スプレッドシート' (Spreadsheet). The menu bar includes 'ファイル(E)', 'ウィンドウ(W)', and 'ヘルプ(H)'. The spreadsheet has four columns labeled A, B, C, and D. The first two rows contain data for two car instances. The first instance has a body width of 100, a body height of 50, and a wheel diameter of 20. The second instance has a body width of 100, a body height of 50, and a wheel diameter of 40.

A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	20	
100	50	40	

（クラス名： Ex34Car2）

※既存のEx32Car2を編集する。

例題34

```
1 package j2.lesson03;
2
3 import gpjava.Spreadsheet;
4
5 public class Ex34Car2 {
6     public static void main(String[] args) {
7         new Ex34Car2().start();
8     }
9
10    void start() {
11        MyCar car = new MyCar();
12        MyCar car1 = new MyCar();
13        car1.wheelDiameter = 40;
14
15        Spreadsheet.show();
16        header();
17        carData(car, 1);
18        carData(car1, 2);
19    }
20
21    void header() {
22        Spreadsheet.setString(0, 0, "車体の横幅");
23        Spreadsheet.setString(0, 1, "車体の縦幅");
24        Spreadsheet.setString(0, 2, "車輪の直径");
25    }
26
27    void carData(MyCar aCar, int row) {
28        int width = aCar.bodyWidth;
29        int height = aCar.bodyHeight;
30        int diameter = aCar.wheelDiameter;
31        Spreadsheet.setInt(row, 0, width);
32        Spreadsheet.setInt(row, 1, height);
33        Spreadsheet.setInt(row, 2, diameter);
34    }
```

```

void start() {
    MyCar car = new MyCar();
    MyCar car1 = new MyCar();
    car1.wheelDiameter = 40;

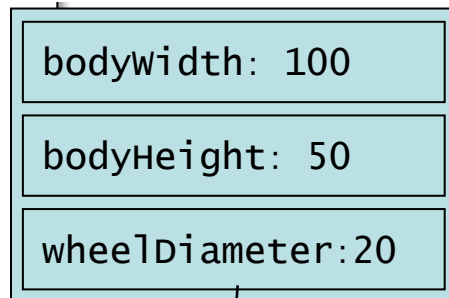
    Spreadsheet.show();
    header();
    carData(car, 1);
    carData(car1, 2);
}

void header() {
    Spreadsheet.setString(0, 0, "車体の横幅");
    Spreadsheet.setString(0, 1, "車体の縦幅");
    Spreadsheet.setString(0, 2, "車輪の直径");
}

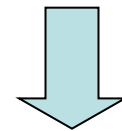
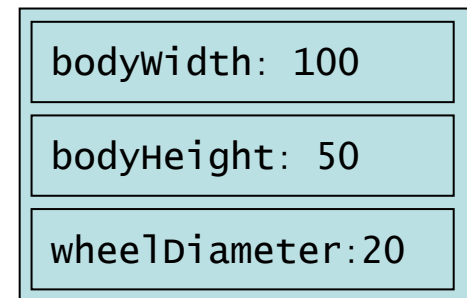
void carData(MyCar aCar, int row) {
    int width = aCar.bodyWidth;
    int height = aCar.bodyHeight;
    int diameter = aCar.wheelDiameter;
    Spreadsheet.setInt(row, 0, width);
    Spreadsheet.setInt(row, 1, height);
    Spreadsheet.setInt(row, 2, diameter);
}

```

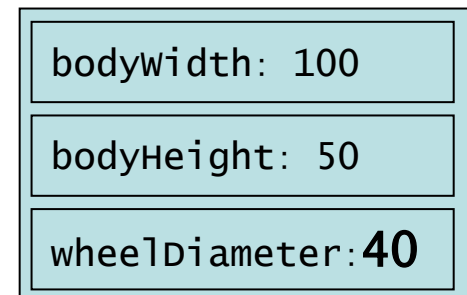
car



car1



car1



スプレッドシート

ファイル(E) ウィンドウ(W) ヘルプ(H)

A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	20	
100	50	40	

■ 例題35

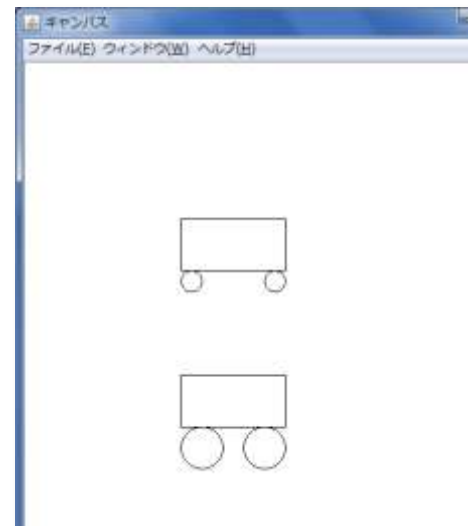
問題：例題34 に加えて2台の「車」をキャンバスに描画せよ。

返り値の型	メソッド名(引数)	機能
void	<i>drawCar(MyCar car, int x, int y)</i>	座標(x,y)を起点として引数MyCarインスタンスに対応した「車」を描く。 (前と同じ)



A screenshot of a spreadsheet application titled 'スプレッドシート'. The menu bar includes 'ファイル(E)', 'ウィンドウ(W)', and 'ヘルプ(H)'. The spreadsheet has four columns labeled A, B, C, and D. The first row contains headers: '車体の横幅' (Car body width) in A, '車体の縦幅' (Car body height) in B, and '車輪の直径' (Wheel diameter) in C. The second row contains values: 100 in A, 50 in B, and 20 in C. The third row contains values: 100 in A, 50 in B, and 40 in C. The remaining rows are empty.

A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	20	
100	50	40	



(クラス名: Ex35Car2)
※既存のEx34Car2を編集する。

例題35

```
1 package j2.lesson03;
2
3 import gpjava.Canvas;
4 import gpjava.Spreadsheet;
5
6 public class Ex35Car2 {
7     public static void main(String[] args) {
8         new Ex35Car2().start();
9     }
10
11     void start() {
12         MyCar car = new MyCar();
13         MyCar car1 = new MyCar();
14         car1.wheelDiameter = 40;
15
16         Spreadsheet.show();
17         Canvas.show();
18         header();
19
20         carData(car, 1);
21         drawCar(car, 150, 150);
22         carData(car1, 2);
23         drawCar(car1, 150, 300);
24     }
25
26     void header() {
27
28     }
29
30     void carData(MyCar aCar, int row) {
31
32     }
33
34     void drawCar(MyCar car, int x, int y) {
35         int width = car.bodyWidth;
36         int height = car.bodyHeight;
37         int diameter = car.wheelDiameter;
38         Canvas.drawRect(x, y, width, height); // body
39         Canvas.drawOval(x, y + height, diameter, diameter); // wheel (left)
40         Canvas.drawOval(x+width-diameter, y+height,
41                         diameter, diameter); // wheel (right)
42     }
43 }
44
45 }
```

+マークは、メソッド
を折りたたんでいる
ことを表す。
header(), carData()
は、例題34と同じ

```

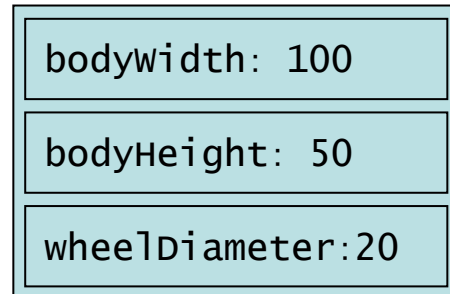
void start() {
    MyCar car = new MyCar();
    MyCar car1 = new MyCar();
    car1.wheelDiameter = 40;

    Spreadsheet.show();
    Canvas.show();
    header();

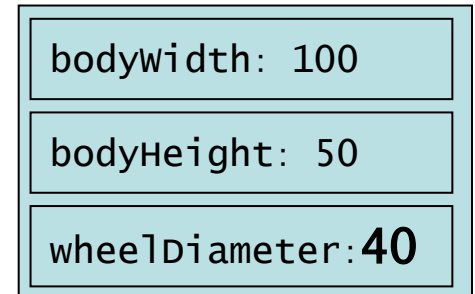
    carData(car, 1);
    drawCar(car, 150, 150);
    carData(car1, 2);
    drawCar(car1, 150, 300);
}

```

car



car1



スプレッドシート

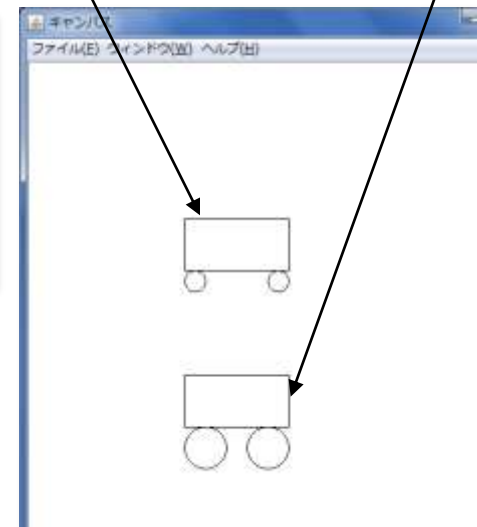
ファイル(E) ウィンドウ(W) ヘルプ(H)

A	B	C	D
車体の横幅	車体の縦幅	車輪の直径	
100	50	20	
100	50	40	

```

void drawCar(MyCar car, int x, int y) {
    int width = car.bodyWidth;
    int height = car.bodyHeight;
    int diameter = car.wheelDiameter;
    Canvas.drawRect(x, y, width, height); // body
    Canvas.drawOval(x, y + height, diameter, diameter); // wheel (left)
    Canvas.drawOval(x+width-diameter, y+height,
                    diameter, diameter); // wheel (right)
}

```



例題41～43（参考）

- これまでと同様の問題であるので、目を通して分かるようであれば、打ち込む必要はない

■ 例題41

問題： MyCarのインスタンスを3つ作成し、それぞれが異なる形(大きさ)となるよう、インスタンス変数の値を変更せよ。

(クラス名: Ex41Car3)

例題41

```
1 package j2.lesson03;
2
3 public class Ex41Car3 {
4     public static void main(String[] args) {
5         new Ex41Car3().start();
6     }
7
8     void start() {
9         MyCar car = new MyCar();
10        MyCar car1 = new MyCar();
11        car1.wheelDiameter = 40;
12        MyCar car2 = new MyCar();
13        car2.bodyWidth = 300;
14    }
15 }
```

■ 例題42

問題：例題41で作成した3つのインスタンスの内容を表示せよ。

返り値の型	メソッド名(引数)	機能
<code>void</code>	<code>carData(MyCar aCar, nit row)</code>	クラスMyCarのインスタンスの内容を Spreadsheet の row 行に表示する(前と同じ)



A	B	C	D	E
車体の横幅	車体の縦幅	車輪の直径		
100	50	20		
100	50	40		
300	50	20		

※既存のEx41Car3を編集する。
(クラス名: Ex42Car3)

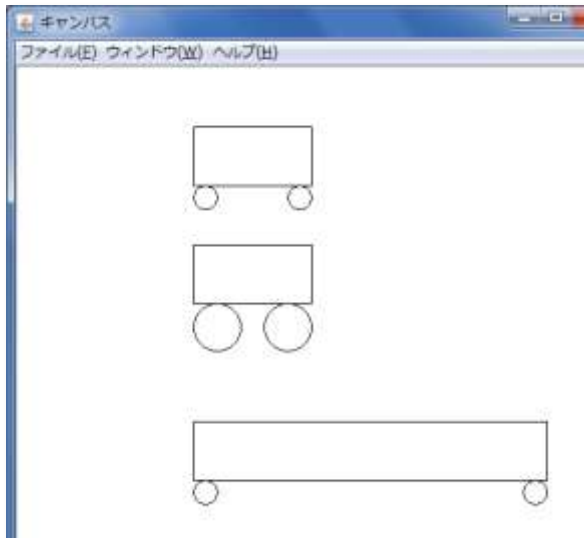
例題42

```
1 package j2.lesson03;
2
3 import gpjava.Spreadsheet;
4
5 public class Ex42Car3 {
6     public static void main(String[] args) {
7         new Ex42Car3().start();
8     }
9
10    void start() {
11        MyCar car = new MyCar();
12        MyCar car1 = new MyCar();
13        car1.wheelDiameter = 40;
14        MyCar car2 = new MyCar();
15        car2.bodyWidth = 300;
16
17        Spreadsheet.show();
18        header();
19        carData(car, 1);
20        carData(car1, 2);
21        carData(car2, 3);
22    }
23
24    void header() {
25        Spreadsheet.setString(0, 0, "車体の横幅");
26        Spreadsheet.setString(0, 1, "車体の縦幅");
27        Spreadsheet.setString(0, 2, "車輪の直径");
28    }
29
30    void carData(MyCar aCar, int row) {
31        int width = aCar.bodyWidth;
32        int height = aCar.bodyHeight;
33        int diameter = aCar.wheelDiameter;
34        Spreadsheet.setInt(row, 0, width);
35        Spreadsheet.setInt(row, 1, height);
36        Spreadsheet.setInt(row, 2, diameter);
37    }
38 }
```

■ 例題43

問題：例題43 MyCarのインスタンスを3つ作成し、その内容にもとにして3台の「車」をキャンバスに描画せよ。

返り値の型	メソッド名(引数)	機能
void	<code>drawCar(MyCar car, int x, int y)</code>	座標(x,y)を起点として引数MyCarインスタンスに対応した「車」を描く。 (前と同じもの)



※既存のEx42Car3を編集する。

(クラス名： Ex43Car3)

例題43

```
1 package j2.lesson03;
2
3 import gpjava.Canvas;
4 import gpjava.Spreadsheet;
5
6 public class Ex43Car3 {
7     public static void main(String[] args) {
8         new Ex43Car3().start();
9     }
10
11     void start() {
12         MyCar car = new MyCar();
13         MyCar car1 = new MyCar();
14         car1.wheelDiameter = 40;
15         MyCar car2 = new MyCar();
16         car2.bodyWidth = 300;
17
18         Spreadsheet.show();
19         Canvas.show();
20         header();
21
22         carData(car, 1);
23         drawCar(car, 150, 50);
24         carData(car1, 2);
25         drawCar(car1, 150, 150);
26         carData(car2, 3);
27         drawCar(car2, 150, 300);
28     }
29
30     void header() {
31
32     }
33
34     void carData(MyCar aCar, int row) {
35
36     }
37
38     void drawCar(MyCar car, int x, int y) {
39         int width = car.bodyWidth;
40         int height = car.bodyHeight;
41         int diameter = car.wheelDiameter;
42         Canvas.drawRect(x, y, width, height); // body
43         Canvas.drawOval(x, y + height, diameter, diameter); // wheel (left)
44         Canvas.drawOval(x+width-diameter, y+height,
45                         diameter, diameter); // wheel (right)
46     }
47 }
48
49 }
```

+マークは、メソッド
を折りたたんでいる
ことを表す。
header(), carData()
は、例題42と同じ

■例題 50 (以降は余裕のある人向き)

次に説明するクラスMovingBallを作成せよ。

- クラスMovingBall

このクラスのインスタンスは以下のインスタンス変数を用いて(2次元の)位置と速度を持つ「ボール」を表現するデータを保持する。

インスタンス変数	初期値	説明
int x	0	現在位置のx座標
int y	0	現在位置のy座標
int vx	0	x方向の速度
int vy	0	y方向の速度
int diameter	10	ボールの直径

(クラス名:MovingBall)

例題50

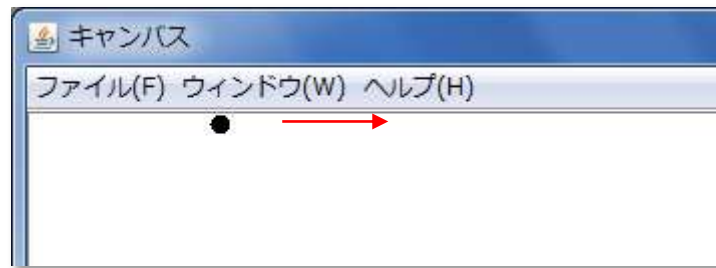
```
1 package j2.lesson03;  
2  
3 public class MovingBall {  
4     int x;  
5     int y;  
6     int vx;  
7     int vy;  
8     int diameter = 10;  
9 }
```

■ 例題51

- 問題: 次のアニメーションを作成せよ。

座標(0,0)に、ボール(円)を配置し、これを初期位置として、x軸方向に1ステップ10ドットで速度を進ませよ。(0,500)の位置に達したら終了とせよ。次のメソッドdrawBallを用いよ。

返り値の型	メソッド名(引数)	機能
void	drawBall(MovingBall aBall)	引数aBallに与えられるインスタンスの情報をもとに、キャンバス上にボールを描画する。



(クラス名: Ex51MovingBall)

例題51

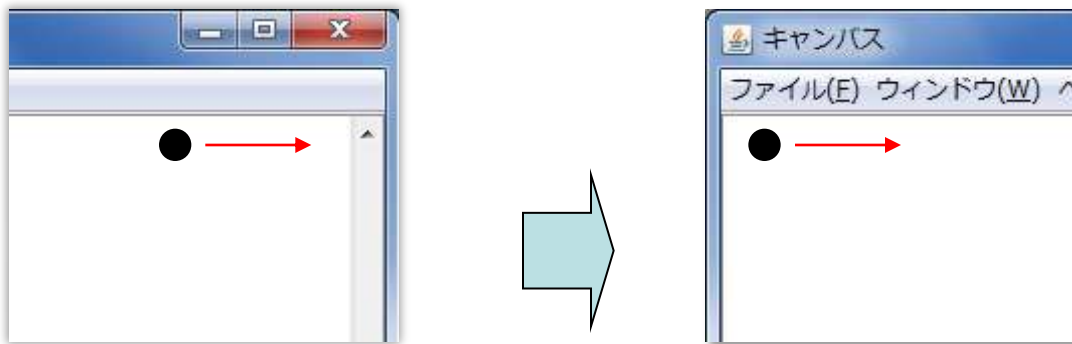
```
1 package j2.lesson03;
2
3 import gpjava.Canvas;
4
5 public class Ex51MovingBall {
6     public static void main(String[] args) {
7         new Ex51MovingBall().start();
8     }
9
10    void start() {
11        Canvas.show();
12        MovingBall ball = new MovingBall();
13        ball.vx = 10;
14        ball.vy = 0;
15
16        do {
17            drawBall(ball);
18            Canvas.waitForCountdown(50);
19            Canvas.clear();
20            ball.x += ball.vx;
21        } while (ball.x < 500);
22    }
23
24    void drawBall(MovingBall aBall) {
25        Canvas.fillOval(aBall.x, aBall.y, aBall.diameter, aBall.diameter);
26    }
27 }
28
29
```

■ 例題52

例題51を変更して、次のアニメーションを作成せよ。

座標(0,0)を、ボールの初期位置として、x軸方向に1ステップ10ドットの速度で進ませる。なお(0,500)より右に進んだら、ボールを(0,0)の位置に戻し、引き続き同じ速度で進むようにせよ。

(キャンバスの右端に到達したら、左端からボールが出てくるイメージ)



(クラス名: Ex52MovingBall)

例題52

```
1 package j2.lesson03;
2
3 import gpjava.Canvas;
4
5 public class Ex52MovingBall {
6     public static void main(String[] args) {
7         new Ex52MovingBall().start();
8     }
9     void start() {
10         Canvas.show();
11         MovingBall ball = new MovingBall();
12         ball.vx = 10;
13         ball.vy = 0;
14
15         for (int i = 0; i < 10000; i++) { //draw 10000 frames
16             drawBall(ball);
17             Canvas.waitForCountdown(50);
18             Canvas.clear();
19             int newX = ball.x + ball.vx;
20             if (newX >= 500) {
21                 ball.x = 0;
22             } else {
23                 ball.x = newX;
24             }
25         }
26     }
27
28     void drawBall(MovingBall aBall) {
29         Canvas.fillOval(aBall.x, aBall.y, aBall.diameter, aBall.diameter);
30     }
31 }
```