2014 年 10 月 20 日 小テスト（3 組用）

組	番	氏名	点数

設問 1

右記プログラムは任意の丸（直径は 20）が赤色または白色で描画される掲示板クラスを作成した．Keijiban クラスでは，[①]である Keijiban(int _num)を利用することで，縦横の丸の数を指定することが可能となっている．また，Keijiban2 クラスの scroll というメソッドを用いると，掲示板上の赤丸が上から下へと 1 つずつ移動し，下端の赤丸が上端から登場する．この Keijiban2 クラスを用いて縦横それぞれ 10 個の掲示板を描画したいと考え下記のメインプログラムを書いた．ここで myKeiji を[②]と呼ぶ．
メインプログラムの 4 行目の display メソッドが呼び出された時，左上から右に 3 つ目，下に 2 つ目（lights[2][1]）の○は[③]色になっている．
設問中，プログラム中の丸囲み数字について回答せよ．

メインプログラム

```
1  Keijian2 myKeiji = new Keijiban2( ④ );
2  myKeiji.scroll();
3  myKeiji.scroll();
4  myKeiji.display();
```

解答欄
①
②
③
④
⑤
⑥
⑦

```
class Keijiban {
  int [][] lights;
  int num;
  Keijiban( int _num ) {
    num = _num;
    lights = [    ⑤    ];
    for( int y=0; y<num; y++ ) {
      for( int x=0; x<num; x++ ){
        lights[x][y] = (x+y)%2;
      }
    }
  }
  void display() {
    for( int y=0; y<num; y++ ) {
      for( int x=0; x<num; x++ ){
        if ( lights[x][y] == 1 ) {
          fill( 255, 0, 0 );
        } else {
          fill( 255, 255, 255 );
        }
        ellipse( x*20+10, y*20+10, 20, 20 );
      }
    }
  }
}

class Keijiban2 extends Keijiban {
  void scroll() {
    int x=0;
    while ( x<num ) {
      int y=num-1;
      int temp = lights[x][num-1];
      while ( y>0 ) {
        lights[x][y] = [    ⑥    ];
        y--;
      }
      [    ⑦    ] = temp;
      x++;
    }
  }
}
```

（裏へ続く）

設問 2

下記メインプログラムは，右記のクラス定義を利用したプログラムである．init メソッドで指定した座標とスピードをもとに，move メソッドで動き，display メソッドで移動した先に描画されるものである．

メインプログラムの 8 行目で表示されるのは，下記の(ア)～(オ)のうち[①]であり，10 行目で表示されるのは(ア)～(オ)のうち[②]である．

(ア) ★　(イ) ·　(ウ) ○　(エ) △　(オ) ×

8 行目，10 行目でそれぞれ表示されるオブジェクトの中心座標（x, y）は[③]，[④]となる．PlanetSatellite クラスで利用可能なインスタンス変数（フィールド変数）は[⑤]である．なお，このままだと PlanetSatellite のインスタンスは移動しない．その理由は[⑥]である．設問中の①～⑥について回答せよ．

メインプログラム

```
1  Ball b = new Ball ();
2  PlanetSatellite ps = new PlanetSatellite();
3  Star st = new Star();
4  b.init( 50, 50, 10 );
5  ps.init( 100, 200, 5 );
6  st.init( 300, 200, 8 );
7  st.move();
8  st.display();
9  b.move();
10 b.display();
```

解答欄
①
②
③
④
⑤
⑥

Ball，Star，PlanetSatellite クラス

```
class Ball {
  int x;
  int y;
  int speed;
  Ball(){
    x = 0;
    y = 0;
    speed = 0;
  }
  void init( int _x, int _y, int _s ){
    x = _x;
    y = _y;
    speed = _s;
  }
  void move() {
    x = x + speed;
  }
  void display(){
    ellipse( x, y, 30, 30 );
  }
}

class Star extends Ball {
  void move(){
    y = y + speed;
  }
}

class PlanetSatellite extends Ball {
  int theta;
  void init( int _x, int _y, int _s ){
    theta  = 0;
  }
  void display() {
    theta = theta + 45;
    ellipse( x, y, 30, 30 );
    int rx = (int)(x+30*sin(radians(theta)));
    int ry = (int)(y+30*cos(radians(theta)));
    ellipse( rx, ry, 10, 10 );
  }
}
```

2014 年 10 月 20 日 小テスト（4 組用）

組	番	氏名	点数

設問 1

右記プログラムは，100 秒から 0 秒まで数字でカウントダウンする TimerBase クラスと，その TimerBase クラスを［ ① ］して描画のスタイルをプログレスバー形式に変更した TimerProgress クラスである．最初に start メソッドが呼び出された時の millis()メソッドを実行した時の返り値が 10000 で，あるタイミングで getTimeLeft()内で millis()メソッドを呼び出した時の返り値が 20000 だったときは getTimeLeft()は［ ② ］という値を，millis()の返り値が 200000 だったときは getTimeLeft()は［ ③ ］という値を返す．なお，millis()は，現在何ミリ秒かということを返すメソッドである．なお，TimerBase の display メソッドで数字を表示しているのに，［ ① ］した TimerProgress クラスの display メソッドを呼び出した時に数字が表示されないのは［ ④ ］が理由である．TimerBase クラスの⑤はコンストラクタの定義である．TimerProgress のインスタンス化の時に下記のように記述する場合にどう記述するべきか埋めよ．

```
TimerProgress tp = new TimerProgress();
```

黒い長方形に占める白い長方形の横幅で残り時間を表現するプログレスバーを表示しようと思った場合に⑥を埋めよ．なお，rect メソッドの仕様は右下に示す通りである．

解答欄
①
②
③
④
⑤
⑥

TimerBase および TimerProgress クラス

```
class TimerBase {
  int iStartMillis;

  [      ⑤      ]{
    iStartMillis = 0;
  }
  void start(){
    iStartMillis = millis();
  }
  void display(){
    int timeLeft = getTimeLeft();
    fill( 0 );
    text( timeLeft, 50, 50 );
  }
  int getTimeLeft(){
    int now = millis();
    if( 100-(now-iStartMillis)/1000 < 0 ){
      return 0;
    }
    return 100-(now-iStartMillis)/1000;
  }
}

class TimerProgress extends TimerBase {
  void display(){
    int timeLeft = getTimeLeft();
    fill( 0 );
    rect( 50, 50, 200, 50 );
    fill( 255 );
    rect( 50, 50, [    ⑥    ]);
  }
}
```

Name: rect()

Draws a rectangle to the screen. A rectangle is a four-sided shape with every angle at ninety degrees. By default, the first two parameters set the location of the upper-left corner, the third sets the width, and the fourth sets the height.

設問 2

右記のクラスの定義をもとに，下記メインプログラムの結果がどう出力されるか回答せよ．

ただし，プログラム内で利用されているメソッドはそれぞれ下記のように定義されている．

メインプログラムの 11 行目で表示されるのは，下記のア～オのうち[①]であり，5 行目で表示されるのは同じく下記ア～オのうち[②]である．

（ア）★　（イ）・　（ウ）○　（エ）△　（オ）×

11 行目の display メソッドが呼ばれた時の b の中心座標(x, y)=([③])であり，5 行目の display メソッドが呼ばれた時の tri の中心座標(x, y)=([④])である．

Ball クラスの中でオーバーライドされているメソッドは[⑤]である．

メインプログラム

```
1  Ball　b = new Ball( 10, 10, 20, 20 );
2  Triangle　tri = new Triangle( 50, 50, 5, 5 );
3  tri.move();
4  tri.move();
5  tri.display();
6  int i=0;
7  while( i < 10 ){
8    b.move();
9    i++;
10 }
11 b.display();
```

解答欄
①
②
③
④
⑤

Object クラスとその派生クラス

```
class Object {
  int x;
  int y;
  int speedX;
  int speedY;

  Object( int _x, int _y, int _sx, int _sy){
    x = _x;
    y = _y;
    speedX = _sx;
    speedY = _sy;
  }
  void move() {
    x = x + speedX;
    y = y + speedY;
  }
  void display(){
    point( x, y );
  }
}

class Ball extends Object {
  void display() {
    fill( 0, 255, 0 );
    ellipse( x, y, 30, 30 );
  }
  void move(){
    x = x + speedX;
  }
}

class Cross extends Object {
  void display() {
    line( x-15, y-15, x+15, y+15 );
    line( x-15, y+15, x+15, y-15 );
  }
}

class Triangle extends Object {
}
```