

เกณฑ์การตรวจมิติเทอม progmeth

1 คลาส Cybertron (8.5 คะแนน)

```
abstract class Cybertron {  
    /**  
     * the life point of robot.  
     */  
    private int life;  
  
    /**  
     * position of robot, in X and Y axis on screen.  
     */  
    private Point position;  
  
    /**  
     * status of robot, true if robot is in a vehicle shape, false otherwise.  
     */  
    private boolean isVehicle;
```

ต้องเป็น abstract (1 คะแนน)

ตัวแปร private ครบและถูกต้อง 1
คะแนน ถ้าผิดก็ไม่ได้คะแนนเลย

```
public int getLife() {  
    return life;  
}  
  
public Point getPosition() {  
    return position;  
}  
  
public int getX() {  
    return position.getX();  
}  
  
public int getY() {  
    return position.getY();  
}  
  
public void setLife(int life) {  
    this.life = life;  
}  
  
public void setPosition(Point position) {  
    this.position = position;  
}  
  
public void setX(int a) {  
    position.setX(a);  
}  
  
public void setY(int a) {  
    position.setY(a);  
}  
  
public void setVehicle(boolean isVehicle) {  
    this.isVehicle = isVehicle;  
}  
  
public boolean isVehicle() {  
    return isVehicle;  
}
```

เมธอดละ 0.5
คะแนน รวมเป็น 5
คะแนน อันไหนผิด
อันนั้นก็ไม่ได้คะแนน

```
/**  
 * This method move the robot in the X-axis. How far each robot can move  
 * depends on each type of robot.  
 *  
 * @param x  
 * integer specifying direction. positive value means the robot  
 * must attempt to move in a positive direction along the X-axis.  
 * Negative value means the robot must attempt to move in a  
 * negative direction along the X-axis.  
 */  
public abstract void moveHorizontal(int x);
```

นิยาม abstract method นี้ได้ถูกต้อง 0.5 คะแนน ถ้าผิดก็ไม่ได้คะแนน

```

/**
 * This method move the robot in the Y-axis. How far each robot can move
 * depends on each type of robot.
 *
 * @param y
 * integer specifying direction. positive value means the robot
 * must attempt to move in a positive direction along the Y-axis.
 * Negative value means the robot must attempt to move in a
 * negative direction along the Y-axis.
 */
public abstract void moveVertical(int y);
/**
 * This method just reverses the status of isVehicle.
 */
public void transform() {
    isVehicle = !isVehicle;
}

```

นิยาม abstract method นี้ได้ถูกต้อง 0.5 คะแนน ถ้าผิดก็ไม่ได้คะแนน

ได้ถูกต้อง 0.5 คะแนน ถ้าผิดก็ไม่ได้คะแนน

2. คลาส Autobot (27 คะแนน ในที่มีคะแนนเพิ่มให้ 0.5)

```
class Autobot extends Cybertron implements Robot {
```

นิยาม ตัวแปร private นี้ได้ถูกต้อง 0.5 คะแนน ถ้าผิดก็ไม่ได้คะแนนเลย

ข้อ 3 การ implement ได้ 1 คะแนน ถ้าไม่ทำก็ 0 ตรงนี้

```

private int spirit;

public int getSpirit() {
    return spirit;
}

public void setSpirit(int spirit) {
    this.spirit = spirit;
}

```

ตรงนี้บล็อคละ 0.5 คะแนน หลักการทำงานต้องตรงกับที่อยู่ใน
เฉลยนี้ทำนะ ถ้าผิดส่วนไหน ส่วนนั้นไม่ได้คะแนน

```

public Autobot(int newlife, int newx, int newy, boolean newIsvehicle, int spi){
    if (newlife >20 || newlife <0)
        newlife = 20;
    setLife(newlife);

    setPosition(new Point());

    if (newx >500 || newx <0)
        newx = 250;
    setX(newx);

    if (newy >500 || newy <0)
        newy = 250;
    setY(newy);

    setVehicle(newIsvehicle);

    if (spi >10 || spi <0)
        spi = 5;
    setSpirit(spi);
}

```

ตรงนี้บล็อคละ 0.5 คะแนน หลักการ
ทำงานต้องตรงกับที่อยู่ในเฉลยนี้ทำ (ดู
โจทย์ประกอบ)

ถ้าผิดส่วนไหน ส่วนนั้นไม่ได้คะแนน

```

@Override
public void moveHorizontal(int x) {
    int newX = 0;
    newX = getX() + (spirit*x);
    if (newX < 0)
        newX = 0;

    if (newX > 500)
        newX = 500;

    setX(newX);
}

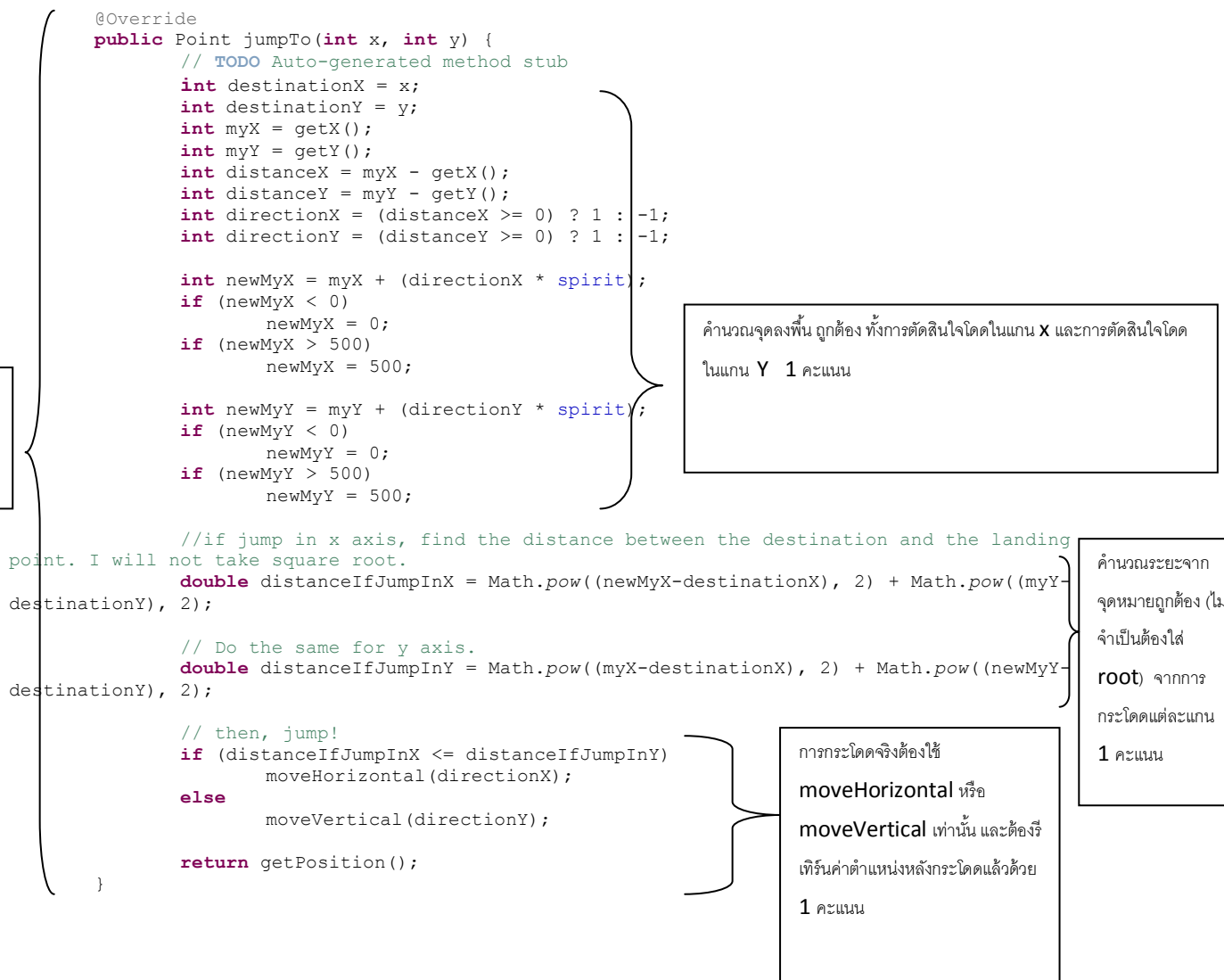
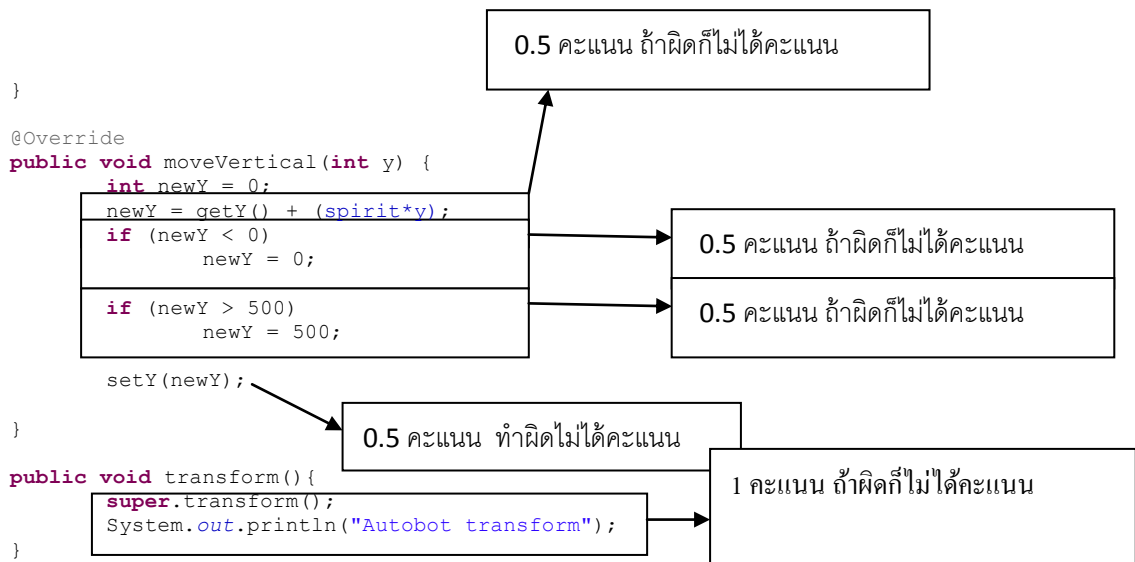
```

หาตำแหน่งใหม่ 0.5 คะแนน ถ้าผิดก็ไม่ได้คะแนน

0.5 คะแนน ถ้าผิดก็ไม่ได้คะแนน

0.5 คะแนน ถ้าผิดก็ไม่ได้คะแนน

0.5 คะแนน ถ้าทำผิดก็ไม่ได้คะแนน



}

3. แทรกไปกับข้อ 2

4. คลาส CargoShip (14 คะแนน **box ละ 1 คะแนน** ถ้าผิดหรือไม่มี ก็ให้ box นั้น 0 คะแนน)

```

public class CargoShip {
    private Robot[] store;
    private Point position;
}

public Robot[] getStore() {
    return store;
}

public void setStore(Robot[] store) {
    this.store = store;
}

public Point getPosition() {
    return position;
}

public void setPosition(Point position) {
    this.position = position;
}

public CargoShip(int x, int y) {
    if (x < 0 || x > 500)
        x = 250;
    if (y < 0 || y > 500)
        y = 250;
    position = new Point(x, y);
    position.setX(x);
    position.setY(y);
    this.store = new Robot[5];
}

public Robot launch(int add_x, int add_y) {
    Robot ejected = store[0];
    for(int i = 0; i < store.length-1; i++)
        store[i] = store[i+1];
    store[store.length-1] = null;
    int destinationX = getPosition().getX()+add_x;
    int destinationY = getPosition().getY()+add_y;
    ejected.jumpTo(destinationX, destinationY);
    return ejected;
}

public void store(Robot r){
    for(int i = 0; i < store.length; i++)
        if (store[i] == null){
            store[i] = r;
            return;
        }
}

public static void main(String[] args) {
    CargoShip whiteBase = new CargoShip(0, 0);
    Autobot a = new Autobot(20, 0, 0, false, 5);
    SuperWarpRobo b = new SuperWarpRobo(20, 0, 0);
    whiteBase.store(a);
    whiteBase.store(b);
    Robot x = whiteBase.launch(10, 15);
    Autobot m = new Autobot(0, 0, 0, false, 5); // ตรงนี้เป็นเลขอะไรก็ได้ แต่ถ้าไม่มี บรรทัดสุดท้ายจะไม่คอมไพล์
    SuperWarpRobo n;
}

```

ตัวแปรต้องนิยามถูกชนิด

ตัดล้นใจเช็คค่า x ได้ถูกต้อง

ตัดล้นใจเช็คค่า y ได้ถูกต้อง

สร้างออบเจกต์ตำแหน่งของ cargoShip ขึ้นมาถูกต้อง

เช็คค่าตำแหน่งของ cargoShip ได้ถูก

อาร์เรย์ของ cargoShip ต้องมี 5 ช่อง เป็นอาร์เรย์ว่าง

เอาของในตำแหน่งแรกออกไป ของตำแหน่งอื่นต้องเลื่อนทั้งหมด

เรียก jumpTo ให้ค่าตำแหน่งในการเรียกถูกต้อง

ต้องไม่ล้นรีเทิร์นตัวหุ่น

Store เขียนถูกหมด 1 คะแนน

สร้าง CargoShip ให้ถูกต้อง อยู่ตำแหน่ง 0,0

สร้าง หุ่นทั้งสองตัวถูกตามคำสั่งในโจทย์

เอาใส่อาร์เรย์ให้ถูกตำแหน่งตามโจทย์

เรียก launch หนึ่งครั้ง

ตรงนี้เป็นเลขอะไรก็ได้ แต่ถ้าไม่มี บรรทัดสุดท้ายจะไม่คอมไพล์

```

if (x instanceof Autobot){
    m = (Autobot)x;
    System.out.println(m.getX());
    System.out.println(m.getY());
} else {
    n = (SuperWarpRobo)x;
    System.out.println(n.getX());
    System.out.println(n.getY());
}

Robot x2 = whiteBase.launch(3,25);
Autobot m2;
SuperWarpRobo n2;

if (x2 instanceof Autobot){
    m2 = (Autobot)x2;
    System.out.println(m2.getX());
    System.out.println(m2.getY());
} else {
    n2 = (SuperWarpRobo)x2;
    System.out.println(n2.getX());
    System.out.println(n2.getY());
}

```

Check instance, cast และ บรีน

ถูกต้อง ก็ได้คะแนนไป

จะเอาไปเขียนเป็นเมทอดก็ได้จะได้ไม่

ต้องเขียนซ้ำ

```

System.out.println(m.isVehicle());
m.transform();
System.out.println(m.isVehicle());

```

แล้วเรียก transform ต้องให้คอมไพล์ผ่านนะ และต้องบรีนสภาพก่อนกับหลังให้เห็นชัด

ยังมีคลาส Point กับ SuperWarpRobo ที่ไม่คิดคะแนน แต่ต้องเขียน ด้วยนะ

```

public class Point {
    private int x, y;

    public int getX() {
        return x;
    }

    public void setX(int x) {
        this.x = x;
    }

    public int getY() {
        return y;
    }

    public void setY(int y) {
        this.y = y;
    }

    public Point() {
        x = 0;
        y = 0;
    }

    public Point(int x, int y) {
        super();
        this.x = x;
        this.y = y;
    }
}

```

```

public class SuperWarpRobo implements Robot {

    private int life;
    private int x, y;

    public int getLife() {
        return life;
    }

    public void setLife(int life) {
        this.life = life;
    }

    public int getX() {
        return x;
    }

    public void setX(int x) {
        this.x = x;
    }

    public int getY() {
        return y;
    }

    public void setY(int y) {
        this.y = y;
    }

    public SuperWarpRobo(int life, int x, int y) {
        super();
        this.life = life;
        this.x = x;
        this.y = y;
    }

    public Point jumpTo(int x, int y) {
        this.x = x;
        this.y = y;
        return new Point(x, y);
    }

    public void moveHorizontal(int x) {
        if (x < 0)
            this.x = this.x - 1;
        else {
            if (x == 0) { //do nothing
            } else {
                this.x = this.x + 1;
            }
        }

        if (this.x > 500)
            this.x = 500;
        if (this.x < 0)
            this.x = 0;
    }

    public void moveVertical(int y) {
        // ไม่เคลื่อนที่ในแกน Y
    }
}

```