Minecraft

Note

This is a language independent task description. The method naming convention you need to use might be different from the one shown below (camel case) !

Context

You are given the task of creating **block** elements for the new version of the game: **Minecraft**: Minecraft Official Site

Blocks are the basic units of structure that can be directly placed in the world of the game. Right now we can place blocks next to each other only. This way we can create a wall that is one level high.

Every Block has the following properties

- `type` / like: `"mud"`, `"air"` or `"gold"`
- `color`
- `light transmission` / some blocks can get light go through (or even emit light)
- `whether it can be crossed or not`
- `list of the block types it can be adjacent to`

Every block has the following methods

- `canPlaceAfter(block)`
    - Decides whether the given block can be **placed** after the one given as a parameter. Every block can be placed only next to a block that is on its "adjacent to" list.
- `getTransmission()`
    - Tells how much percent of light can the block transmit to the other side of the wall.
- `printStatus()`
    - Prints the most important properties of the block.

Some blocks can have additional properties and methods.

The following elements are needed

- **mud**
  - color: `dark gray`
  - light transmission: `0`
  - `can not` be crossed
  - can be placed after: `brick, air, lava, door, window, gold`
  - prints:

```
Block of type: ...
Has light transmission: ...
Can not be crossed.
```

- **glowstone**
  - color: `white`
  - light transmission: `300`
  - `can not` be crossed
  - can be placed after: `glass, ice, air, lava`
  - it can transform its neighbor (the already existing block they are placed next to)
    - transforms `glass to gold` (that has light transmission 160)
  - prints:

```
Block of type: ...
Has light transmission: ...
Can not be crossed.
Made no transformation / Has transformed a block of type ... to ...
```

- **glass**

  - color: can be `defined at creation`. When not given: `transparent`.
  - light transmission: can be `defied at creation`. When not given: `99%`.
  - `can not be` crossed
  - can be placed after: `mud, brick, ice, air, glowstone, door`
  - prints:

```
Block of type: ...
Has light transmission: ...
Can not be crossed.
```

- **lava**
  - color: `yellow`
  - light transmission: `170`

o can not be crossed
o can be placed after: `mud, brick, glass, ice, air, glowstone`
o it can transform its neighbor (the already existing block they are placed next to)
  ▪ changes `mud` to `brick` and `ice` to `air`
o prints:

```
Block of type: ...
Has light transmission: ...
Can not be crossed.
Made no transformation / Has transformed a block of type ... to ...
```

- **air**

  o color: `transparent`
  o light transmission: `100`
  o can be crossed
  o can be placed after: `mud, brick, glass, door, window, gold`
  o prints:

```
Block of type: ...
Has light transmission: ...
Can be crossed.
```

- **window**

  o quality: must be `defined at creation` (1, 2 or 3)
  o producer name: must be `defined at creation`
  o color: must be `defined at creation`
  o light transmission: must be `defined at creation`
  o can be crossed: must be `defined at creation`
  o can be placed after: `mud, brick, air, door, glowstone`
  o prints:

```
Block of type: ...
Has light transmission: ...
Can be crossed. / Can not be crossed.
Manufactured by ... on quality level ...
```

- **gold**

  o color: `rgb: 255,215,0`
  o light transmission: `must be defined at creation`
  o can not be crossed

- o can be placed after: `glowStone, glass`
- o prints:

```
Block of type: ...
Has light transmission: ...
Can not be crossed.
```

- **ice**

  - o color: `transparent`
  - o light transmission: `97`
  - o `can not be` crossed
  - o can be placed after: `mud, brick, glass, lava, air, glowstone`
  - o it can transform its neighbor (the already existing block they are placed next to)
    - ▪ changes `lava to glass`
  - o prints:

```
Block of type: ...
Has light transmission: ...
Can not be crossed.
Made no transformation / Has transformed a block of type ... to ...
```

- **door**

  - o quality: must be `defined at creation` (1, 2 or 3)
  - o producer name: must be `defined at creation`
  - o color: must be `defined at creation`
  - o light transmission: must be `defined at creation`
  - o `can be` crossed
  - o can be placed after: `mud, brick, air, window`
  - o prints:

```
Block of type: ...
Has light transmission: ...
Can be crossed.
Manufactured by ... on quality level ...
```

- **brick**

  - o color: can be `defined at creation`. If not defined: `rgb: 144, 76, 42`
  - o light transmission: `0`
  - o `can not be` crossed
  - o can be placed after: `mud, glass, air, door, window`

o   prints:

```
Block of type: ...
Has light transmission: ...
Can not be crossed.
```

Your tasks

Create a **Wall** class that

- has a list of blocks

- has a `placeBlock()` method that adds a new block to the list
  - Placement is only successful when the actually placed block has the previous block type on its adjacent block list. Unsuccessful placement means: the wall does not change.
  - At placing you must consider the transformation ability of the new block (if it has any)
- a `printStatus()` method that prints the status of every block in the wall
- a `calculateLightTransmission()` method that returns the percentage (integer between 0 and 100) of the light that passes through the wall. This percentage is the average of the light transmission values of the blocks in the wall.

Apply the following logic:

- Create a wall using a constructor with no parameters
- Add the following new blocks to the wall using the placeBlock() method:
  - Air: created with no parameters
  - Brick: created with dark-gray color
  - Door: created with quality 2, producer 'Mordor', color black and light transmission 60
  - Glass: light transmission 80
  - GlowStone: no parameters
  - Window: quality 1, producer 'Liliput', color white, light transmission 70, can be crossed: true
  - Window: quality 2, producer 'Asgard', color light-gray, light transmission 90, can be crossed: true
  - Mud: no parameters
  - Gold: light transmission 50
  - Lava: no parameters

- GlowStone: no parameters
  - Ice: no parameters
  - Lava: no parameters
  - Gold: light transmission 30
  - Ice: no parameters
- Calculate and print the light transmission of the wall
- Print the status of the wall

Expected print result

```
Light transmitted: 121

Block of type: air
Has light transmission: 100
Can be crossed.

Block of type: brick
Has light transmission: 0
Can not be crossed.

Block of type: door
Has light transmission: 60
Can be crossed.
Manufactured by Mordor on quality level 2

Block of type: gold
Has light transmission: 160
Can not be crossed.

Block of type: glowstone
Has light transmission: 300
Can not be crossed.
Has transformed a block of type glass to gold

Block of type: window
Has light transmission: 70
Can be crossed.
Manufactured by Liliput on quality level 1

Block of type: brick
Has light transmission: 0
Can not be crossed.

Block of type: lava
Has light transmission: 170
Can not be crossed.
Has transformed a block of type mud to brick

Block of type: glowstone
```

```
Has light transmission: 300
Can not be crossed.
Made no transformation

Block of type: air
Has light transmission: 100
Can be crossed.

Block of type: glass
Has light transmission: 99
Can not be crossed.

Block of type: ice
Has light transmission: 97
Can not be crossed.
Has transformed a block of type lava to glass
```
After having a solution with this result, make sure that you have efficiently minimized code duplication.