

JPIN  
0.1.0-beta

構築: Doxygen 1.11.0



1 クラス索引	1
1.1 クラス一覧	1
2 ファイル索引	2
2.1 ファイル一覧	2
3 クラス詳解	3
3.1 Main クラス	3
3.1.1 詳解	3
3.1.2 関数詳解	3
3.1.2.1 main()	3
3.2 Notification クラス	4
3.2.1 詳解	5
3.2.2 構築子と解体子	5
3.2.2.1 Notification()	5
3.2.3 関数詳解	6
3.2.3.1 getMessage()	6
3.2.3.2 getTimestamp()	6
3.2.4 メンバ詳解	7
3.2.4.1 message	7
3.2.4.2 timestamp	7
3.3 NotificationList クラス	7
3.3.1 詳解	8
3.3.2 構築子と解体子	8
3.3.2.1 NotificationList()	8
3.3.3 関数詳解	8
3.3.3.1 addNotification()	8
3.3.3.2 getNotifications()	9
3.3.4 メンバ詳解	9
3.3.4.1 notifications	9
3.4 Sensor クラス	10
3.4.1 詳解	11
3.4.2 構築子と解体子	11
3.4.2.1 Sensor()	11
3.4.3 関数詳解	12
3.4.3.1 notifyStatus()	12
3.4.3.2 setOccupied()	12
3.4.4 メンバ詳解	13
3.4.4.1 id	13
3.4.4.2 isOccupied	13
3.4.4.3 location	14
3.4.4.4 notificationList	14
4 ファイル詳解	15

4.1 C:/Users/Kmax/main.java ファイル . . . . .	15
4.2 C:/Users/Kmax/main.java . . . . .	15
索引	17

## 第 1 章

### クラス索引

#### 第 1.1 節 クラス一覧

クラス・構造体・共用体・インターフェースの一覧です。

<a href="#">Main</a>	メインクラス。このクラスはプログラムのエントリーポイントを提供し、センサーの状態変更と通知の表示を行う。 . . . . .	<a href="#">3</a>
<a href="#">Notification</a>	通知を表すクラス。このクラスは通知のタイムスタンプとメッセージを持つ。 . . . . .	<a href="#">4</a>
<a href="#">NotificationList</a>	通知リストを管理するクラス。このクラスは通知のリストを持ち、通知の追加やリストの取得を行う。 . . . . .	<a href="#">7</a>
<a href="#">Sensor</a>	センサーを表すクラス。このクラスは各センサーの識別ID、設置場所、優先中かどうかの状態を持ち、状態が変更された際に通知を送る機能を持つ。 . . . . .	<a href="#">10</a>

## 第 2 章

### ファイル索引

#### 第 2.1 節 ファイル一覧

ファイル一覧です。

C:/Users/Kmax/[main.java](#) . . . . . 15

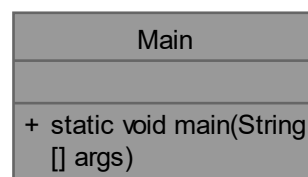
## 第 3 章

### クラス詳解

#### 第 3.1 節 Main クラス

メインクラス。このクラスはプログラムのエントリーポイントを提供し、センサーの状態変更と通知の表示を行う。

Main 連携図



#### 静的公開メンバ関数

- static void `main` (String[] args)  
プログラムのエントリーポイント。

##### 第 3.1.1 節 詳解

メインクラス。このクラスはプログラムのエントリーポイントを提供し、センサーの状態変更と通知の表示を行う。

`main.java` の 135 行目に定義があります。

##### 第 3.1.2 節 関数詳解

###### 第 3.1.2.1 節 `main()`

```
static void Main.main (  
    String[] args) [inline], [static]
```

プログラムのエントリーポイント。

引数

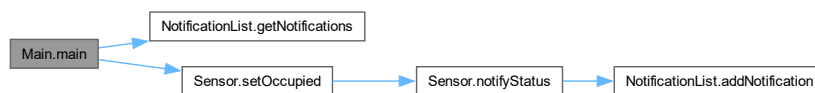
args	コマンドライン引数
------	-----------

`main.java` の 142 行目に定義があります。

```
00142     {  
00143         NotificationList notificationList = new NotificationList();  
00144         Sensor sensor = new Sensor(1, "Bathroom", notificationList);  
00145  
00146         sensor.setOccupied(true);  
00147         sensor.setOccupied(false);  
00148  
00149         for (Notification notification : notificationList.getNotifications()) {  
00150             System.out.println(notification.getTimestamp() + " - " + notification.getMessage());  
00151         }  
00152     }
```

参照先 `NotificationList.getNotifications()`, `Sensor.setOccupied()`.

呼び出し関係図:



このクラス詳解は次のファイルから抽出されました:

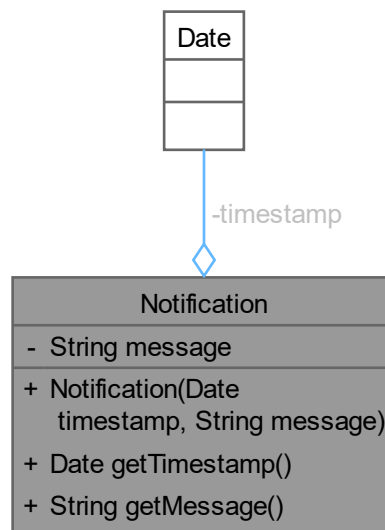
- `C:/Users/Kmax/main.java`

## 第 3.2 節 Notification クラス

通知を表すクラス。このクラスは通知のタイムスタンプとメッセージを持つ。



## Notification 連携図



## 公開メンバ関数

- `Notification` (`Date timestamp`, `String message`)  
通知のコンストラクタ。
- `Date getTimestamp ()`  
通知のタイムスタンプを取得する。
- `String getMessage ()`  
通知のメッセージを取得する。

## 非公開変数類

- `Date timestamp`
- `String message`

## 第 3.2.1 節 詳解

通知を表すクラス。このクラスは通知のタイムスタンプとメッセージを持つ。

`main.java` の 60 行目に定義があります。

## 第 3.2.2 節 構築子と解体子

第 3.2.2.1 節 `Notification()`

```
Notification.Notification (  
    Date timestamp,  
    String message) [inline]
```

通知のコンストラクタ。

引数

timestamp	通知のタイムスタンプ
message	通知のメッセージ

[main.java](#) の 70 行目に定義があります。

```
00070 {
00071     this.timestamp = timestamp;
00072     this.message = message;
00073 }
```

参照先 [message](#), [timestamp](#).

### 第 3.2.3 節 関数詳解

#### 第 3.2.3.1 節 getMessage()

String Notification.getMessage () [inline]

通知のメッセージを取得する。

戻り値

メッセージ

[main.java](#) の 89 行目に定義があります。

```
00089 {
00090     return message;
00091 }
```

参照先 [message](#).

#### 第 3.2.3.2 節 getTimestamp()

Date Notification.getTimestamp () [inline]

通知のタイムスタンプを取得する。

戻り値

タイムスタンプ

[main.java](#) の 80 行目に定義があります。

```
00080 {
00081     return timestamp;
00082 }
```

参照先 [timestamp](#).

### 第 3.2.4 節 メンバ詳解

#### 第 3.2.4.1 節 message

String Notification.message [private]

[main.java](#) の 62 行目に定義があります。

参照元 [getMessage\(\)](#), [Notification\(\)](#).

#### 第 3.2.4.2 節 timestamp

Date Notification.timestamp [private]

[main.java](#) の 61 行目に定義があります。

参照元 [getTimestamp\(\)](#), [Notification\(\)](#).

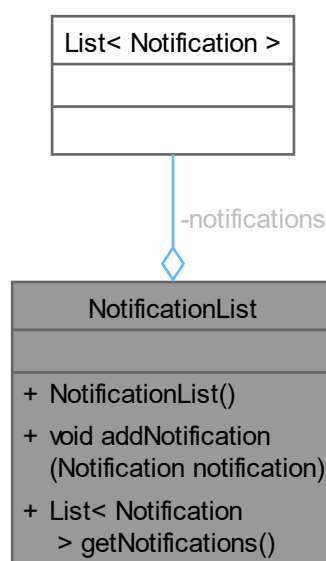
このクラス詳解は次のファイルから抽出されました:

- C:/Users/Kmax/[main.java](#)

## 第 3.3 節 NotificationList クラス

通知リストを管理するクラス。このクラスは通知のリストを持ち、通知の追加やリストの取得を行う。

NotificationList 連携図



## 公開メンバ関数

- `NotificationList ()`  
通知リストのコンストラクタ。
- `void addNotification (Notification notification)`  
通知をリストに追加する。
- `List< Notification > getNotifications ()`  
通知リストを取得する。

## 非公開変数類

- `List< Notification > notifications`

## 第 3.3.1 節 詳解

通知リストを管理するクラス。このクラスは通知のリストを持ち、通知の追加やリストの取得を行う。

`main.java` の 100 行目に定義があります。

## 第 3.3.2 節 構築子と解体子

## 第 3.3.2.1 節 NotificationList()

`NotificationList.NotificationList () [inline]`

通知リストのコンストラクタ。

`main.java` の 106 行目に定義があります。

```
00106      {
00107          this.notifications = new ArrayList<>();
00108      }
```

## 第 3.3.3 節 関数詳解

## 第 3.3.3.1 節 addNotification()

`void NotificationList.addNotification (Notification notification) [inline]`

通知をリストに追加する。

## 引数

notification	追加する通知
--------------	--------

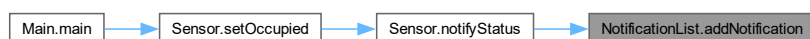
`main.java` の 115 行目に定義があります。

```
00115      {
00116          notifications.add(notification);
00117      }
```

参照先 `notifications`.

参照元 `Sensor.notifyStatus()`.

被呼び出し関係図:



## 第 3.3.3.2 節 getNotifications()

```
List< Notification > NotificationList.getNotifications () [inline]
```

通知リストを取得する。

戻り値

通知リスト

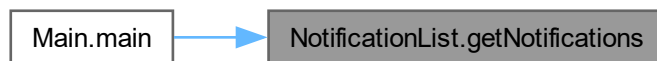
[main.java](#) の 124 行目に定義があります。

```
00124         {  
00125             return notifications;  
00126     }
```

参照先 [notifications](#).

参照元 [Main.main\(\)](#).

被呼び出し関係図:



## 第 3.3.4 節 メンバ詳解

## 第 3.3.4.1 節 notifications

```
List<Notification> NotificationList.notifications [private]
```

[main.java](#) の 101 行目に定義があります。

参照元 [addNotification\(\)](#), [getNotifications\(\)](#).

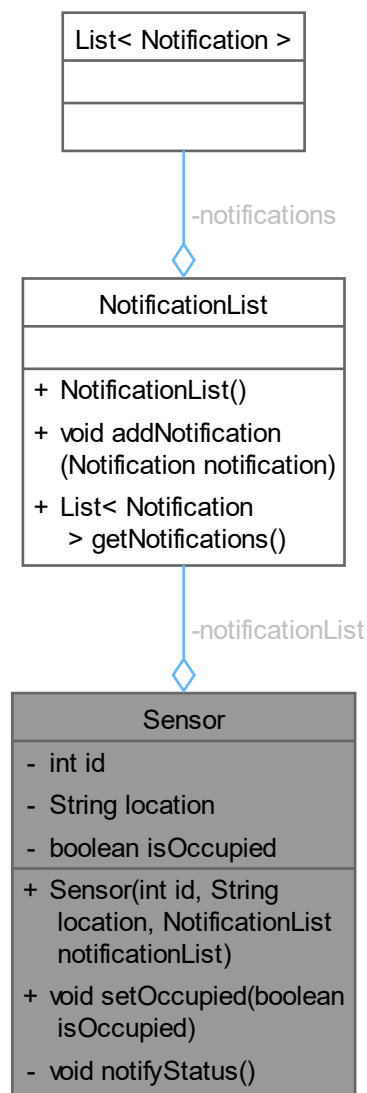
このクラス詳解は次のファイルから抽出されました:

- C:/Users/Kmax/[main.java](#)

### 第 3.4 節 Sensor クラス

センサーを表すクラス。このクラスは各センサーの識別ID、設置場所、優先中かどうかの状態を持ち、状態が変更された際に通知を送る機能を持つ。

Sensor 連携図



公開メンバ関数

- `Sensor` (`int id`, `String location`, `NotificationList notificationList`)  
センサーのコンストラクタ。
- `void setOccupied` (`boolean isOccupied`)  
センサーの状態を設定する。

## 非公開メンバ関数

- void `notifyStatus ()`  
状態を通知するための内部メソッド。

## 非公開変数類

- int `id`
- String `location`
- boolean `isOccupied`
- `NotificationList` `notificationList`

## 第 3.4.1 節 詳解

センサーを表すクラス。このクラスは各センサーの識別ID、設置場所、優先中かどうかの状態を持ち、状態が変更された際に通知を送る機能を持つ。

`main.java` の 16 行目に定義があります。

## 第 3.4.2 節 構築子と解体子

第 3.4.2.1 節 `Sensor()`

```
Sensor.Sensor (  
    int id,  
    String location,  
    NotificationList notificationList) [inline]
```

センサーのコンストラクタ。

## 引数

id	センサーの識別ID
location	センサーの設置場所
notificationList	通知リスト

`main.java` の 29 行目に定義があります。

```
00029  
00030     this.id = id; // センサー ID  
00031     this.location = location; // センサーの場所。  
00032     this.notificationList = notificationList;  
00033 }
```

参照先 `id`, `location`, `notificationList`.

### 第 3.4.3 節 関数詳解

#### 第 3.4.3.1 節 notifyStatus()

```
void Sensor.notifyStatus () [inline], [private]
```

状態を通知するための内部メソッド。

`main.java` の 48 行目に定義があります。

```
00048      {
00049          String message = isOccupied ? "Occupied" : "Vacant";
00050          Notification notification = new Notification(new Date(), message);
00051          notificationList.addNotification(notification);
00052      }
```

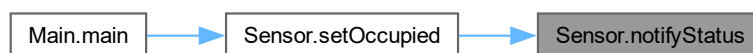
参照先 `NotificationList.addNotification()`, `isOccupied`, `notificationList`.

参照元 `setOccupied()`.

呼び出し関係図:



被呼び出し関係図:



#### 第 3.4.3.2 節 setOccupied()

```
void Sensor.setOccupied (
    boolean isOccupied) [inline]
```

センサーの状態を設定する。

引数

isOccupied	センサーが優先中かどうか
------------	--------------



---

`main.java` の 40 行目に定義があります。

```
00040      {  
00041          this.isOccupied = isOccupied;  
00042          notifyStatus();  
00043      }
```

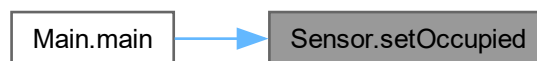
参照先 `isOccupied`, `notifyStatus()`.

参照元 `Main.main()`.

呼び出し関係図:



被呼び出し関係図:



### 第 3.4.4 節 メンバ詳解

#### 第 3.4.4.1 節 id

```
int Sensor.id [private]
```

`main.java` の 17 行目に定義があります。

参照元 `Sensor()`.

#### 第 3.4.4.2 節 isOccupied

```
boolean Sensor.isOccupied [private]
```

`main.java` の 19 行目に定義があります。

参照元 `notifyStatus()`, `setOccupied()`.

#### 第 3.4.4.3 節 location

String Sensor.location [private]

[main.java](#) の 18 行目に定義があります。

参照元 [Sensor\(\)](#).

#### 第 3.4.4.4 節 notificationList

[NotificationList](#) Sensor.notificationList [private]

[main.java](#) の 20 行目に定義があります。

参照元 [notifyStatus\(\)](#), [Sensor\(\)](#).

このクラス詳解は次のファイルから抽出されました:

- C:/Users/Kmax/[main.java](#)

## 第 4 章

### ファイル詳解

#### 第 4.1 節 C:/Users/Kmax/main.java ファイル

##### クラス

- class [Sensor](#)  
センサーを表すクラス。このクラスは各センサーの識別ID、設置場所、優先中かどうかの状態を持ち、状態が変更された際に通知を送る機能を持つ。
- class [Notification](#)  
通知を表すクラス。このクラスは通知のタイムスタンプとメッセージを持つ。
- class [NotificationList](#)  
通知リストを管理するクラス。このクラスは通知のリストを持ち、通知の追加やリストの取得を行う。
- class [Main](#)  
メインクラス。このクラスはプログラムのエントリーポイントを提供し、センサーの状態変更と通知の表示を行う。

#### 第 4.2 節 C:/Users/Kmax/main.java

##### [詳解]

```
00001
00008 import java.util.ArrayList;
00009 import java.util.Date;
00010 import java.util.List;
00011
00016 class Sensor {
00017     private int id; // 各センサーの識別 ID。これで、複数箇所のセンサーを識別。
00018     private String location; // センサー設置場所。これで場所を特定。
00019     private boolean isOccupied; // センサーが優先中かどうか。
00020     private NotificationList notificationList; // センサーの通知を保存。
00021
00029     public Sensor(int id, String location, NotificationList notificationList) {
00030         this.id = id; // センサー ID
00031         this.location = location; // センサーの場所。
00032         this.notificationList = notificationList;
00033     }
00034
00040     public void setOccupied(boolean isOccupied) {
00041         this.isOccupied = isOccupied;
00042         notifyStatus();
00043     }
00044
00048     private void notifyStatus() {
```

```

00049     String message = isOccupied ? "Occupied" : "Vacant";
00050     Notification notification = new Notification(new Date(), message);
00051     notificationList.addNotification(notification);
00052 }
00053
00054 // その他のメソッドや機能はここに追加
00055 }
00056
00060 class Notification {
00061     private Date timestamp;
00062     private String message;
00063
00070     public Notification(Date timestamp, String message) {
00071         this.timestamp = timestamp;
00072         this.message = message;
00073     }
00074
00080     public Date getTimestamp() {
00081         return timestamp;
00082     }
00083
00089     public String getMessage() {
00090         return message;
00091     }
00092
00093 // その他のメソッドや機能はここに追加
00094 }
00095
00100 class NotificationList {
00101     private List<Notification> notifications;
00102
00106     public NotificationList() {
00107         this.notifications = new ArrayList<>();
00108     }
00109
00115     public void addNotification(Notification notification) {
00116         notifications.add(notification);
00117     }
00118
00124     public List<Notification> getNotifications() {
00125         return notifications;
00126     }
00127
00128 // その他のメソッドや機能はここに追加
00129 }
00130
00135 public class Main {
00136
00142     public static void main(String[] args) {
00143         NotificationList notificationList = new NotificationList();
00144         Sensor sensor = new Sensor(1, "Bathroom", notificationList);
00145
00146         sensor.setOccupied(true);
00147         sensor.setOccupied(false);
00148
00149         for (Notification notification : notificationList.getNotifications()) {
00150             System.out.println(notification.getTimestamp() + " - " + notification.getMessage());
00151         }
00152     }
00153 }

```

## 索引

- addNotification
  - NotificationList, [8](#)
- C:/Users/Kmax/main.java, [15](#)
- getMessage
  - Notification, [6](#)
- getNotifications
  - NotificationList, [8](#)
- getTimestamp
  - Notification, [6](#)
- id
  - Sensor, [13](#)
- isOccupied
  - Sensor, [13](#)
- location
  - Sensor, [13](#)
- Main, [3](#)
  - main, [3](#)
- main
  - Main, [3](#)
- message
  - Notification, [7](#)
- Notification, [4](#)
  - getMessage, [6](#)
  - getTimestamp, [6](#)
  - message, [7](#)
  - Notification, [5](#)
  - timestamp, [7](#)
- NotificationList, [7](#)
  - addNotification, [8](#)
  - getNotifications, [8](#)
  - NotificationList, [8](#)
  - notifications, [9](#)
- notificationList
  - Sensor, [14](#)
- notifications
  - NotificationList, [9](#)
- notifyStatus
  - Sensor, [12](#)
- Sensor, [10](#)
  - id, [13](#)
  - isOccupied, [13](#)
  - location, [13](#)
  - notificationList, [14](#)
  - notifyStatus, [12](#)
  - Sensor, [11](#)
  - setOccupied, [12](#)
- setOccupied
  - Sensor, [12](#)
- timestamp
  - Notification, [7](#)