**Objektum katalógus**

**LED**

A hálózat nyelője. Nem rendelkezik kimenettel, csak egyetlen bementtel, amire egy vezeték van csatlakoztatva. Mindig pontosan ennek a vezetéknek az értékét adja vissza.

**andGate**

Logikai ÉS kaput megvalósító objektum. A bemeneteiről beolvasott értékekből kiszámolja és továbbadja a kimenetére az új értéket. Az új értékét az ÉS kapu igazságtáblája szerint számolja ki, mely két bemenet esetén a következő:

|  |  |  |
| --- | --- | --- |
| **bemenet** | | **kimenet** |
| A | B | A AND B |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |
| X | 0 | 0 |
| X | 1 | X |
| 0 | X | 0 |
| 1 | X | X |
| X | X | X |

Jelentések:

0: logikai HAMIS érték

1: logikai IGAZ érték

X: don’t care

**orGate**

Logikai VAGY kaput megvalósító objektum. A bemeneteiről beolvasott értékekből kiszámolja és továbbadja a kimenetére az új értéket. Az új értékét az VAGY kapu igazságtáblája szerint számolja ki, mely két bemenet esetén a következő:

|  |  |  |
| --- | --- | --- |
| **bemenet** | | **kimenet** |
| A | B | A OR B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |
| X | 0 | X |
| X | 1 | 1 |
| 0 | X | X |
| 1 | X | 1 |
| X | X | X |

**inverter**

Logikai invertert megvalósító objektum. Egyetlen egy bemenettel rendelkezik, és kimenetére ennek inverzét adja, ha annak van értelme. Igazságtáblája a következő:

|  |  |
| --- | --- |
| **bemenet** | **kimenet** |
| A | NOT A |
| 0 | 1 |
| 1 | 0 |
| X | X |

**Osztályok leírása**

### Output

* + - * **Felelősség**

Absztakt osztály a megjelenítő modulok (pl. LED) leszármaztatására.

* + - * **Ősosztályok**

DigitalObject

* + - * **Interfészek**

Implementálja a IComponent interfészt.

* + - * **Attribútumok**

Feedback :array

ID :string

* + - * **Metódusok**

Count()

GetID() :string

Step()

*Az ősosztály (DigitalObject) attribútumain, illetve metódusain kívül nem definiál újakat.*

### LED

* + - * **Felelősség**

Output absztrakt osztályt valósítja meg. Lekérdezi a hozzá csatolt vezeték (wireIn) értékét, és megjeleníti azt a felhasználó számára.

* + - * **Ősosztályok**

DigitalObject->Output

* + - * **Interfészek**

Implementálja a IComponent interfészt.

* + - * **Attribútumok**

Feedback :array

ID :string

SameAsPrevious :boolean

* + - * **Metódusok**

Count()

GetID() :string

Step()

*Az ősosztály (Output) attribútumain, illetve metódusain kívül nem definiál újakat.*

### Gate

* + - * **Felelősség**

Absztrakt osztály a logikai kapuk (pl. ÉS, VAGY, inverter) leszármaztatására. A bementi vezeték(ek)nek lekérdezi az értékét, és kiszámolja a kimenet új értékét, és be is állítja azt, az leszármazott osztály igazságtáblája szerint.

* + - * **Ősosztályok**

DigitalObject

* + - * **Interfészek**

Implementálja a IComponent interfészt.

* + - * **Attribútumok**

Feedbacks :array

Ha egy Gate egyik bemenete egy visszacsatolás kezdete, akkor tartalmaz egy Feedbacks tömböt mely referenciát tárol az összes, az adott visszacsatolásban részt vevő DigitalObject-re.

ID :string

PreviousValue: int

A legutolsó ciklus ( Step() ) eredményét (kimeneti értékét) tárolja, a stabilitásellenőrzés céljából (Count() metódus).

* + - * **Metódusok**

Calculate(array) : int

A metóduson belül lekérdezi a bemeneti vezetékek (wireIn) értékét a GetValue() :int meghívásával, mely felelősége visszaadni a vezeték értékét, mely érték alapján kiszámolja a kimeneti vezeték (wireOut) új értékét az igazságtáblájának megfelelő módon, és beállítja a vezeték értékét a SetValue(int) metódus segítségével.

Count()

GetID() :string

Step()

### andGate

* + - * **Felelősség**

Gate absztrakt osztályt valósítja meg, bemeneti értékeit lekérdezi (wireIn) a GetValue() :int metódussal, majd előállítja a kimeneti értéket (wireOut) a logika táblája alapján és beállítja azt, a SetValue(int) segítségével.

* + - * **Ősosztályok**

DigitalObject->Gate

* + - * **Interfészek**

Implementálja a IComponent interfészt.

* + - * **Attribútumok**

Feedbacks: array

ID :string

PreviousValue: int

* + - * **Metódusok**

Calculate(array) : int

Count()

GetID() :string

Step()

*Az ősosztály (Gate) attribútumain, illetve metódusain kívül nem definiál újakat.*

### orGate

* + - * **Felelősség**

Gate absztrakt osztályt valósítja meg, bemeneti értékeiből (wireIn) előállítja a kimeneti értéket (wireOut) a logika táblája alapján.

* + - * **Ősosztályok**

DigitalObject->Gate

* + - * **Interfészek**

Implementálja a IComponent interfészt.

* + - * **Attribútumok**

Feedbacks: array

ID :string

PreviousValue: int

* + - * **Metódusok**

Calculate(array) : int

Count()

GetID() :string

Step()

*Az ősosztály (Gate) attribútumain, illetve metódusain kívül nem definiál újakat.*

### Inverter

* + - * **Felelősség**

Gate absztrakt osztályt valósítja meg, bemeneti értékét (wireIn) lekérdezi a GetValue(int) metódussal, és előállítja a kimeneti értéket (wireOut) a logika táblája alapján, melyet a kenetére ad a SetValue(int) függvénnyel.Csak egyetlen bemenete lehet.

* + - * **Ősosztályok**

DigitalObject->Gate

* + - * **Interfészek**

Implementálja a IComponent interfészt.

* + - * **Attribútumok**

Feedbacks: array

ID :string

PreviousValue: int

* + - * **Metódusok**

Calculate(array) : int

Count()

GetID() :string

Step()

*Az ősosztály (Gate) attribútumain, illetve metódusain kívül nem definiál újakat.*