



Arbeitsauftrag Informatik

Name:

Vorname:

Klasse:

CrossRoads-Modell – Methoden der Klasse Hardware

Das CrossRoads-Modell zeigt eine beleuchtete Ampelkreuzung mit vier Fahrzeugampeln, acht bedarfsgesteuerten Fußgängerampeln für die Überwege, sechs Induktionsschleifen, einem Blitzer und einem Helligkeitssensor. Das Modell wird über USB mit dem PC verbunden und kann in der Entwicklungsumgebung über das unsichtbare Objekt `hardware` der Klasse `Hardware` verwendet werden.



Klassenkarte der Klasse Hardware

Hardware
<pre>+ledOn(int) +ledOff(int) +allOutputsOff() +getOutput(int): boolean +getLight(): int +isPressed(int): boolean +wasPressed(int): boolean +isContacted(int): boolean +showStatus() +pause(int) +startClock() +stopClock(): long +loop(Object,String) +loopWithDelay(Object,String,int) +listLoops() +stopLoop(Object,String) +stopLoop(String) +stopLoopId(int) +stopLoop(int) +stopAllLoops()</pre>



Arbeitsauftrag Informatik

Name:

Vorname:

Klasse:

Methoden für Sensoren/Aktoren

ledOn(int): void

LED einschalten, Parameter \triangleq LED-Nummer

ledOff(int): void

LED ausschalten, Parameter \triangleq LED-Nummer

allOutputsOff(): void

alle LEDs ausschalten

getOutput(int): boolean

Rückgabe des LED-Zustands, Parameter \triangleq LED-Nummer

getLight(): int

Rückgabe des Messwert [0, 1023] am Helligkeitssensor

isPressed(int): boolean

Rückgabe des Tastenzustands, Parameter \triangleq Tastennummer

wasPressed(int): boolean

Rückgabe, ob Tastendruck vorlag, Parameter \triangleq Tastennummer

isContacted(int): boolean

Rückgabe des Sensorzustands, Parameter \triangleq Sensornummer

Methoden zur Zeitsteuerung

pause(int): void

Programmablauf für Parameterwert in Millisekunden pausieren

startClock(): void

Stoppuhr beginnend bei Null starten

stopClock(): long

Stoppuhr stoppen und Rückgabe der Laufzeit in Millisekunden

Methode zur Visualisierung

showStatus(): void

Grafische Echtzeitanzeige des Modells

Methoden zur Nebenläufigkeit von Methoden

Bei der Loop-Nebenläufigkeit wird eine Methode eines Objekts als Endlosschleife parallel zu allen anderen Prozessen ausgeführt.

loop(Object, String): void

Methode des Objekts nebenläufig ausführen, Parameter geben Objekt und Methodenname an

Beispiel: hardware.loop(lampe1, "blinken");

loopWithDelay(Object, String, int): void

Nebenläufigkeit mit einer Pausezeit in Millisekunden

listLoops(): void

Konsolenausgabe aller nebenläufigen Prozesse mit Speicherindex, Methodenname, Loopindex

stopLoop(pObjekt: Object, pMethode: String): void

Ausführung der Methode des Objekts stoppen

stopLoop(pMethode: String): void

Ausführung der Methode stoppen

stopLoopId(pIndex: int): void

Ausführung der Methode mit dem Loopindex stoppen

stopLoop(pIndex: int): void

Ausführung der Methode mit dem Speicherindex stoppen

stopAllLoops(): void

alle nebenläufigen Methoden stoppen