**Methode „OnTimerExpired()“**

Ist für Subgame abstrakt, Unterklassen implementieren.  
Alle Subgames haben einen Timer.  
TerminableTaskSubgames ändern ihre HasTaskExpired-Eigenschaft nach dessen Ablauf.  
IndefiniteTaskSubgames änden ihre ExpectsReaction-Eigenschaft.

IndefiniteTaskSubgames können eine Benachrichtigung einrichten, die über eine

**SubgameFlandernstrasseOpen**

SubgameFlandernstrasseOpen besitzt in einer XML die Öffnungszeiten der Hochschule am Standort Flandernstraße aus Uhrzeitpaaren mit jeweils einem Wochentag.  
Die einzelnen Öffnungsintervalle werden bei Subgame-Start als Elemente der Klasse WeekbasedDateTimeInterval in ein LinkedList-Member geladen.

WeekbasedDateTimeInterval Members:  
private DateTime start;  
private DateTime end;  
public WeekbasedDateTimeInterval(DateTime start, DateTimeEnd);  
public bool Contains(DateTime);

Die Uhrzeit wird zu Beginn des Tasks an zufälliger Stelle gesetzt und schreitet bei jedem Update() um eine Minute voran.  
Bei Abrufen der ExpectsReaction-Eigenschaft wird geprüft, ob die aktuelle Zeit in einem der WeekbasedDateTimeIntervals liegt.  
  
SubgameFlandernstrasseOpen Members:  
protected override bool ExpectsReaction{get;};  
private DateTime currentTime;  
private LinkedList<WeekbasedDateTimeInterval> openingHours;  
private void AdvanceTimeMins();  
private void UpdateTimeView();  
private void SetRandomTime();  
private void LoadXML();