

TimeRange class

Purpose:

TimeRange is an abstract class with two subclasses: TimePeriod and TimeSlot. It encapsulates interface elements shared by TimePeriod and TimeSlot, and also permits TimePeriod and TimeSlot to be used interchangeably elsewhere in the Schoer system.

Interface:

public int start();

```
// Returns the object's start time, in minutes since midnight.  
// For instance, 8:00 am is 480 (8 x 60)  
//           1:30 pm is 810 (13 x 60 + 30)
```

public int end();

```
// Returns the object's end time, in minutes since midnight.
```

public int duration();

```
// Returns the object's duration in minutes.  
// Note that duration() = end() - start() + 1.  
// For example, a TimeRange that starts at 480 (8:00 am)  
// and ends at 539 (8:59 am) has a duration of 60 minutes.
```

public boolean overlaps(TimeRange tr);

```
// Returns true if this object overlaps in time the parameter tr.
```

Other Information:

Because TimeRange is abstract, no method bodies will be coded within the TimeRange class. The programmer may choose to implement TimeRange as a Java interface. All sub-classes of TimeRange implement the java.io.Serializable interface.

TimePeriod class

Purpose:

TimePeriod encapsulates Schoer functionality specified in section 3.4, specifically (T-1), (T-2), (T-3), and (T-7). Its superclass is TimeRange.

Interface:

TimePeriod(String day, int hour, int minute, int duration);

```
// This constructor creates a new TimePeriod with the specified
// day, starting hour, starting minute, and duration.
// day must be one of "M", "T", "W", "Th", "F", "Sa", or "Su" (case insensitive).
// hour must be 0 to 23 (24 hour clock).
// minute must be 0 to 59.
// duration must be positive; units are minutes
// An IllegalArgumentException is thrown if any parameter is invalid.
// Clients: unknown at time of writing
```

TimePeriod(TimePeriod tp);

```
// This constructor creates a new TimePeriod identical to an existing
// TimePeriod.
```

public int start();

public int end();

public int duration();

public boolean overlaps(TimeRange tr);

```
// As specified in TimeRange documentation.
```

String getDay();

```
// Returns one of "M", "T", "W", "Th", "F", "Sa", "Su".
```

boolean equals(Object o);

```
// Overrides Object.equals. False if o is not a TimeRange.
```

```
// o can be a TimePeriod or a TimeSlot.
```

String toString();

```
// Overrides Object.toString(). Implements requirement (T-7).
```

TimeSlot class

Purpose:

TimeSlot encapsulates Schoor functionality specified in section 3.4, specifically (T-4), (T-5), (T-6), and (T-7). Its superclass is TimeRange.

Interface:

TimeSlot();

// Default constructor.

public int start();

public int end();

public int duration();

public boolean overlaps(TimeRange tr);

// As specified in TimeRange documentation.

void add(TimePeriod timePeriod);

// Adds a new TimePeriod to this TimeSlot's list of TimePeriods.

// Throws an IllegalArgumentException if the TimePeriod's start and

// end times do not match the TimeSlot's other start and end times.

// Throws an IllegalArgumentException if two TimePeriods are add'd

// for the same day.

void setDesirability(String d);

// Sets the TimeSlot's desirability rating.

// Throws IllegalArgumentException if parameter is not "A" or "B" or "C"

// (case is not important).

String getDesirability();

// Returns "A" or "B" or "C" (always upper case) or the empty String if

// the desirability has not been set.

boolean includesDay(String day);

// Returns *true* if one TimePeriod in the TimeSlot has the day specified,

// *false* otherwise.

// (Case of day parameter is not important.)

boolean equals(Object o);

// Overrides Object.equals. False if o is not a TimeRange.

// o can be a TimePeriod or a TimeSlot.

String toString();

// Overrides Object.toString(). Implements requirement (T-7).