Exploring the `java.time` API

Maurice Naftalin
@mauricenaftalin

Summary

Summary

Summary

Core classes

Summary

Core classes

- `LocalDateTime, LocalDate, LocalTime, Duration`

Summary

Core classes

- `LocalDateTime, LocalDate, LocalTime, Duration`

Core class methods

Summary

Core classes

- `LocalDateTime`, `LocalDate`, `LocalTime`, `Duration`

Core class methods

- **Creation, field access, adjustment, comparison, conversion**

Summary

Core classes

- `LocalDateTime`, `LocalDate`, `LocalTime`, `Duration`

Core class methods

- Creation, field access, adjustment, comparison, conversion

Illustrated in operations on `WorkPeriod`

Summary

Core classes

- `LocalDateTime`, `LocalDate`, `LocalTime`, `Duration`

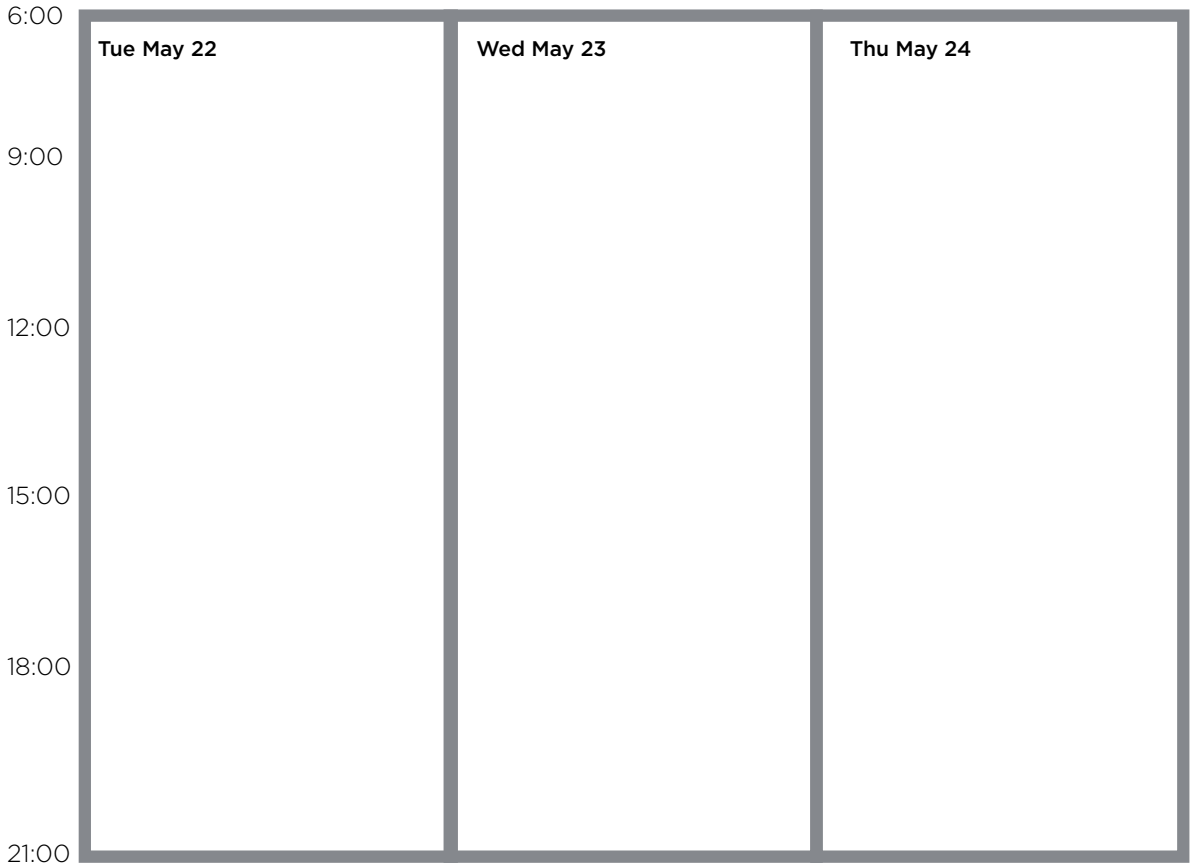
Core class methods

- Creation, field access, adjustment, comparison, conversion

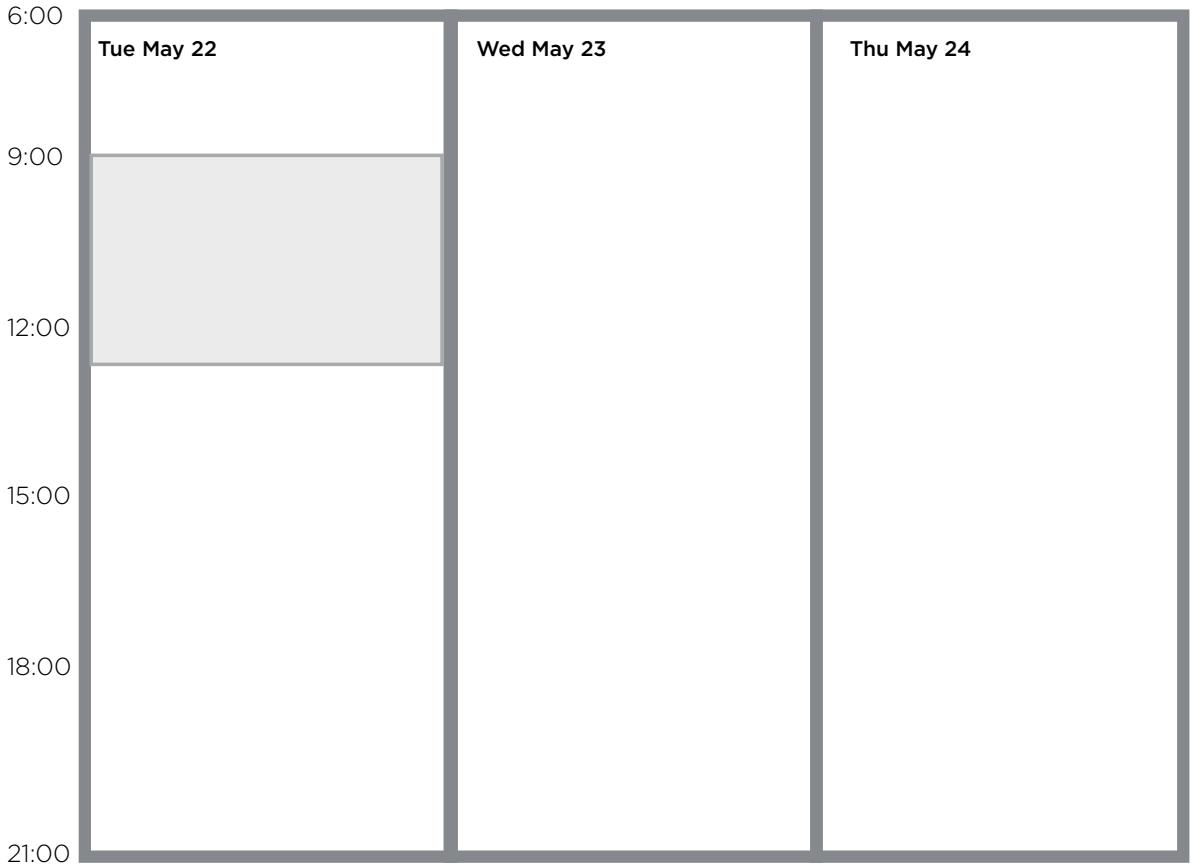
Illustrated in operations on `WorkPeriod`

- **Creation and splitting**

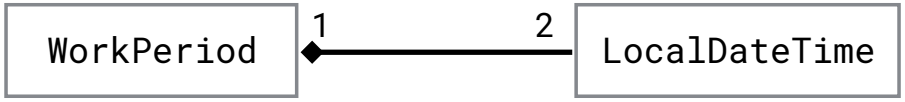
What is a `WorkPeriod`?



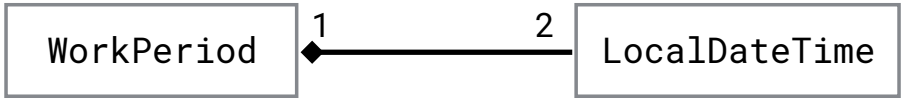
What is a `WorkPeriod`?



What is a `WorkPeriod`?



What is a `WorkPeriod`?




```
Duration wpDuration = Duration.between(wp.getStartTime(), wp.getEndTime());
```

ThreeTen Extra Documentat

About

ThreeTen-Extra provides additional date-time classes that complement those in Java SE 8.

Not every piece of date/time logic is destined for the JDK. Some concepts are too specialized or too bulky to make it in. This project provides some of those additional classes as a well-tested and reliable jar. It is curated by the primary author of the Java 8 date and time library, Stephen Colebourne.

ThreeTen-Extra is licensed under the business-friendly BSD 3-clause license.

Features

The following features are included:

- `DayOfMonth` - a day-of-month without month or year
- `DayOfYear` - a day-of-year without year
- `AmPm` - before or after midday
- `Quarter` - the four quarters, Q1, Q2, Q3 and Q4
- `YearQuarter` - combines a year and quarter, 2014-Q4
- `YearWeek` - combines a week-based-year and a week, 2014-W06
- `Days`, `Weeks`, `Months` and `Years` - amounts of time
- `Interval` - an interval between two instants
- `PeriodDuration` - combines `Period` and `Duration`
- Weekend adjusters
- Coptic calendar system
- Ethiopic calendar system
- Julian calendar system
- Support for the TAI and UTC time-scales

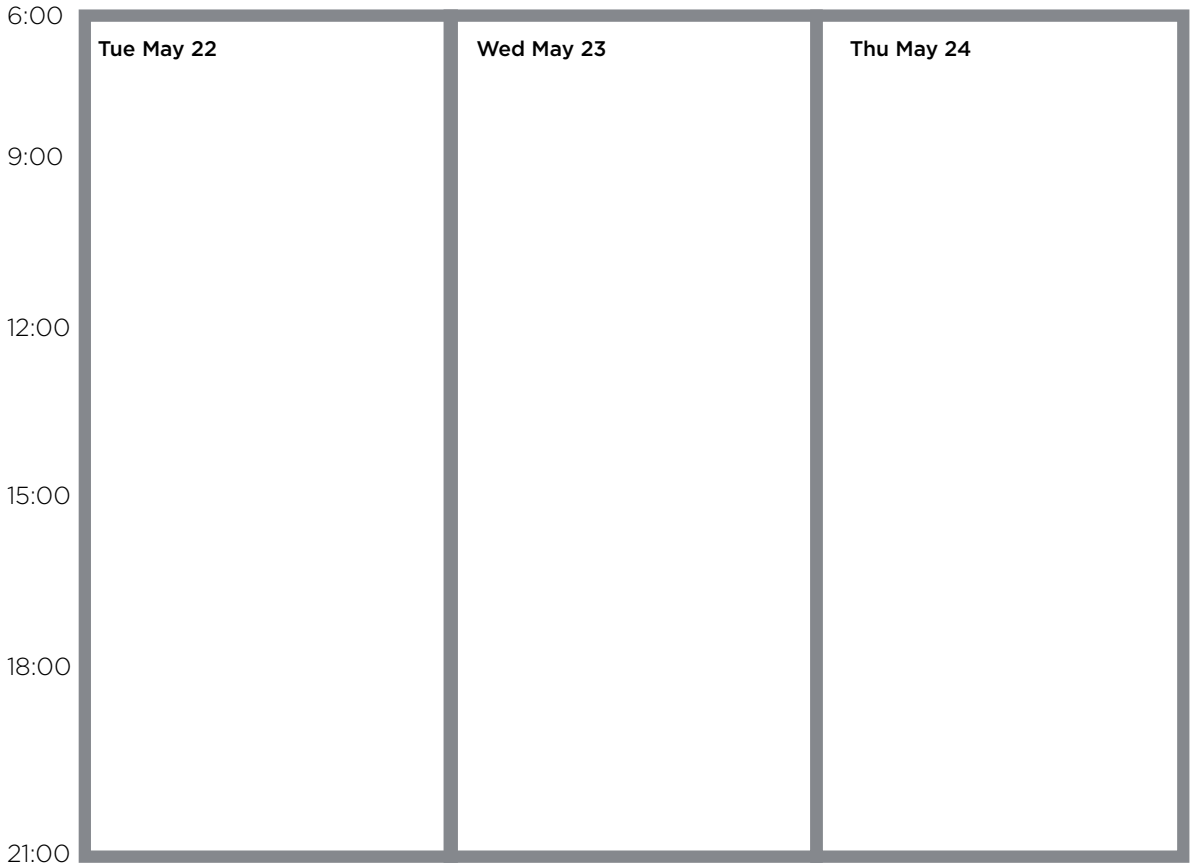
Releases

Release 1.2 is the current release. This release is considered stable and worthy of the 1.x tag.

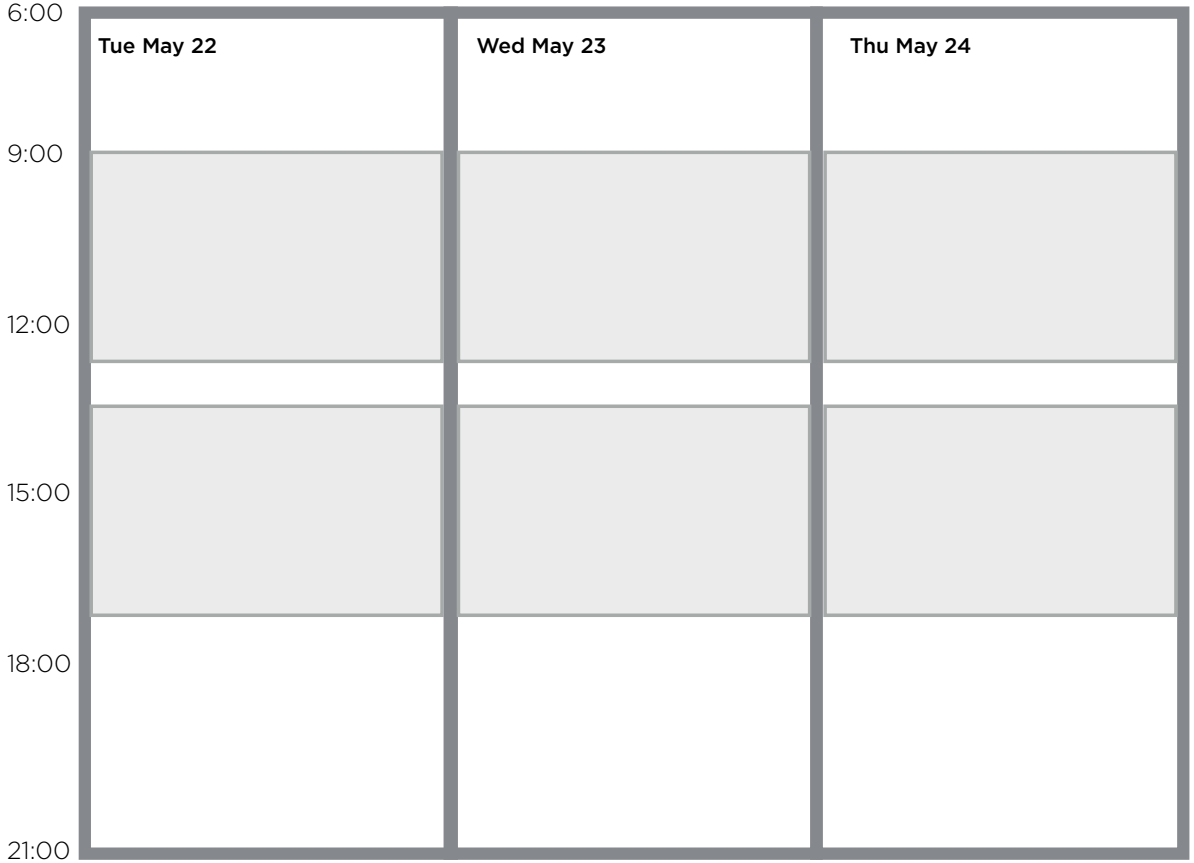
ThreeTen-Extra requires Java SE 8 or later and has no dependencies.

http://www.threeten.org/threeten-extra/

Generating `WorkPeriods`



Generating `WorkPeriods`



Generating WorkPeriods

```
Clock testClock = Clock.fixed(Instant.EPOCH, ZoneOffset.UTC);
LocalDate testDate = LocalDate.now(testClock);

// create a calendar
Calendar calendar = new Calendar();

// add some tasks to the calendar
calendar.addTask( hours: 1,  minutes: 0,  description: "Answer urgent e-mail");
calendar.addTask( hours: 4,  minutes: 0,  description: "Write deployment report");
calendar.addTask( hours: 4,  minutes: 0,  description: "Plan security configuration");

// add some work periods to the calendar
calendar.addWorkPeriods(Utils.generateWorkPeriods(testDate,  dayCountInclusive: 3));

// add an event to the calendar, specifying its time zone
ZonedDateTime meetingTime = ZonedDateTime.of(testDate.atTime( hour: 8,  minute: 30),
        ZoneId.of(("America/New_York")));

// create a working schedule
Schedule schedule = calendar.createSchedule(testClock);
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation				
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation				
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation				
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24)
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		`of(int,int,int)`		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24)
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		`of(int,int,int)`		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24
    List<WorkPeriod>                                              24May2017, 3);
    assertEquals(6,
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

LocalDate factory methods
of(int year, int month, int dayOfMonth)
of(int year, **Month** month, int dayOfMonth)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24
    List<WorkPeriod>                                          24May2017, 3);
    assertEquals(6,
    assertEquals(Arra
            workPeriods.stream()
                    .map(WorkPeriod::getStartTime)
                    .map(LocalDateTime::getDayOfWeek)
                    .distinct()
                    .collect(toList()));
}
```

LocalDate factory methods
of(int year, int month, int dayOfMonth)
of(int year, **Month** month, int dayOfMonth)
ofEpochDay(long epochDay)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24
    List<WorkPeriod>                                          24May2017, 3);
    assertEquals(6,
    assertEquals(Arra
            workPerio
                    .map(                    )
                    .map(LocalDateTime::getDayOfWeek)
                    .distinct()
                    .collect(toList()));
}
```

LocalDate factory methods
of(int year, int month, int dayOfMonth)
of(int year, **Month** month, int dayOfMonth)
ofEpochDay(long epochDay)
ofYearDay(int year, int dayOfYear)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24
    List<WorkPeriod>                                          24May2017, 3);
    assertEquals(6,
    assertEquals(Arra
            workPerio
                    .ma
                    .ma
                    .distinct()
                    .collect(toList()));
}
```

LocalDate factory methods
of(int year, int month, int dayOfMonth)
of(int year, **Month** month, int dayOfMonth)
ofEpochDay(long epochDay)
ofYearDay(int year, int dayOfYear)
now(), **now**(**ZoneId**)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24
    List<WorkPeriod>                                          24May2017, 3);
    assertEquals(6, 
    assertEquals(Arra
            workPerio
                    .ma
                    .ma
                    .di
                    .collect(toList()));
}
```

LocalDate factory methods
of(int year, int month, int dayOfMonth)
of(int year, **Month** month, int dayOfMonth)
ofEpochDay(long epochDay)
ofYearDay(int year, int dayOfYear)
now(), **now**(**ZoneId**)
now(**Clock** clock)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access				
Adjustment				
Comparison				
Conversion				

```
import static java.time.DayOfWeek.*;
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access				
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWorkPeriods() {
    LocalDate thur24May2017 = LocalDate.of(2018, 5, 24);
    List<WorkPeriod> workPeriods = Utils.generateWorkPeriods(thur24May2017, 3);
    assertEquals(6, workPeriods.size());
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
           workPeriods.stream()
                  .map(WorkPeriod::getStartTime)
                  .map(LocalDateTime::getDayOfWeek)
                  .distinct()
                  .collect(toList()));
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGene
    LocalDate thur24M
    List<WorkPeriod>                                    ls(thur24May2017, 3);
    assertEquals(6, w
    assertEquals(Arra
            workPerioc
                    .ma
                    .ma
                    .di
                    .co
}
```

LocalDateTime accessor methods	
int	getDayOfMonth()
DayOfWeek	getDayOfWeek()
int	getDayOfYear()
int	getHour()
int	getMinute()
Month	getMonth()
int	getMonthValue()
int	getNano()
int	getSecond
int	getYear()

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment				
Comparison				
Conversion				

```
@Test
public void testGenerateWo
    LocalDate thur24May2017
    List<WorkPeriod> workPe                                    r24May2017, 3);
    assertEquals(6, workPer
    assertEquals(Arrays.asList(THURSDAY, FRIDAY, MONDAY),
            workPeriods.stream()
                    .map(WorkPeriod::getStartTime)
                    .map(LocalDateTime::getDayOfWeek)
                    .distinct()
                    .collect(toList()));
}
```

LocalDateTime accessor methods

int	**DayOf**
int	
int	
int	**getMinute()**
Month	**getMonth()**
int	**getMonthValue()**
int	**getNano()**
int	**getSecond**
int	**getYear()**

LocalDateTime field accessor methods

int	**get(TemporalField)**
long	**getLong(TemporalField)**

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

LocalDateTime field accessor methods	
int	get(TemporalField)
long	getLong(TemporalField)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment				
Comparison				
Conversion				

LocalDateTime field accessor methods	
int	**get(TemporalField)**
long	**getLong(TemporalField)**

```
import static java.time.temporal.ChronoField.*;
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

LocalDateTime field accessor methods	
int	**get(TemporalField)**
long	**getLong(TemporalField)**

```
import static java.time.temporal.ChronoField.*;
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

LocalDateTime field accessor methods	
int	**get(TemporalField)**
long	**getLong(TemporalField)**

```
import static java.time.temporal.ChronoField.*;

workPeriod.getStartTime().get(AMPM_OF_DAY)
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

LocalDateTime field accessor methods	
int	**get(TemporalField)**
long	**getLong(TemporalField)**

```
import static java.time.temporal.ChronoField.*;

workPeriod.getStartTime().get(AMPM_OF_DAY)
workPeriod.getStartTime().get(DAY_OF_WEEK)
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		`of(int,int,int)`		
Field access			`getDayOfWeek()`	
Adjustment				
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		`of(int,int,int)`		
Field access			`getDayOfWeek()`	
Adjustment				
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment				
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment				
Comparison				
Conversion				

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		`of(int,int,int)`		
Field access			`getDayOfWeek()`	
Adjustment				
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment				
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
            .filter(Utils::isWorkingDay)
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment				
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
            .filter(Utils::isWorkingDay)
            .limit(dayCount)
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
            .filter(Utils::isWorkingDay)
            .limit(dayCount)
            .collect(toList());
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
   return Stream.iterate(startDate, d -> d.plusDays(1))
         .filter(Utils::isWorkingDay)
         .limit(dayCount)
         .collect(toList());
}
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		`of(int,int,int)`		
Field access			`getDayOfWeek()`	
Adjustment				
Comparison				
Conversion				

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		`of(int,int,int)`		
Field access			`getDayOfWeek()`	
Adjustment				
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment				
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		**of(int,int,int)**		
Field access			**getDayOfWeek()**	
Adjustment				
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment				
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

LocalDate adjustment methods

```
List<Lo                                                    dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<Lo                                                    dayCount) {
    retur
```

LocalDate adjustment methods

plusDays(long), plusMonths(long), plusWeeks(long), plusYears(long)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<Lo                                                    dayCount) {
    retu
```

LocalDate adjustment methods
plusDays(long), **plusMonths**(long), **plusWeeks**(long), **plusYears**(long)
minusDays(long), **minusMonths**(long),...

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<Lo                                                    dayCount) {
    retu
```

LocalDate adjustment methods
plusDays(long), **plusMonths**(long), **plusWeeks**(long), **plusYears**(long)
minusDays(long), **minusMonths**(long),...
plus(long, **TemporalUnit**), **minus**(long, **TemporalUnit**)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
(from JDK source)
public enum ChronoUnit implements TemporalUnit {
    NANOS(…),MICROS(…),MILLIS(…),SECONDS(…),MINUTES(…),HOURS(…),…
}
List<Lo                                                    dayCount) {
    retu
```

LocalDate adjustment methods
plusDays(long), **plusMonths**(long), **plusWeeks**(long), **plusYears**(long)
minusDays(long), **minusMonths**(long),...
plus(long, **TemporalUnit**), **minus**(long, **TemporalUnit**)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<Lo                                                    dayCount) {
    retu
```

LocalDate adjustment methods
plusDays(long), **plusMonths**(long), **plusWeeks**(long), **plusYears**(long)
minusDays(long), **minusMonths**(long),...
plus(long, **TemporalUnit**), **minus**(long, **TemporalUnit**)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<Lo                                                    dayCount) {
    retu
```

LocalDate adjustment methods
plusDays(long), **plusMonths**(long), **plusWeeks**(long), **plusYears**(long)
minusDays(long), **minusMonths**(long),...
plus(long, **TemporalUnit**), **minus**(long, **TemporalUnit**)
plus(**TemporalAmount**), **minus**(**TemporalAmount**)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<Lo                                                    dayCount) {
    retu
```

LocalDate adjustment methods
plusDays(long), **plusMonths**(long), **plusWeeks**(long), **plusYears**(long)
minusDays(long), **minusMonths**(long),...
plus(long, **TemporalUnit**), **minus**(long, **TemporalUnit**)
plus(**TemporalAmount**), **minus**(**TemporalAmount**)
withDayOfMonth(int), **withDayOfYear**(int), **withMonth**(int), **withYear**(int)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
            .filter(Utils::isWorkingDay)
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
private static boolean isWorkingDay(LocalDate d) {
    return d.getDayOfWeek() != DayOfWeek.SATURDAY &&
           d.getDayOfWeek() != DayOfWeek.SUNDAY;
}
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
          .filter(Utils::isWorkingDay)
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access			getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
private static boolean isWorkingDay(LocalDate d) {
    return d.getDayOfWeek() != DayOfWeek.SATURDAY &&
           d.getDayOfWeek() != DayOfWeek.SUNDAY;
}
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iterate(startDate, d -> d.plusDays(1))
          .filter(Utils::isWorkingDay)
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
private static boolean isWorkingDay(LocalDate d) {
    return d.getDayOfWeek() != DayOfWeek.SATURDAY &&
           d.getDayOfWeek() != DayOfWeek.SUNDAY;
}
List<LocalDate> (                                    tartDate, int dayCount) {
    return Stream                                    Days(1))
            .filter(
```

LocalDate accessor methods	
int	getDayOfMonth()
DayOfWeek	getDayOfWeek()
int	getDayOfYear()
Month	getMonth()
int	getMonthValue()
int	getYear()

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
private static boolean isWorkingDay(LocalDate d) {
    return d.getDayOfWeek() != DayOfWeek.SATURDAY &&
           d.getDayOfWeek() != DayOfWeek.SUNDAY;
}
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
    return Stream.iter
            .filter(Utils
```

LocalDate accessor methods	
int	getDayOfMonth()
DayOf	
int	
Month	
int	getMonthValue()
int	getYear()

LocalDate field accessor methods	
int	get(TemporalField)
long	getLong(TemporalField)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
private static boolean isWorkingDay(LocalDate d) {
   return d.getDayOfWeek() != DayOfWeek.SATURDAY &&
          d.getDayOfWeek() != DayOfWeek.SUNDAY;
}
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
   return Stream.iterate(startDate, d -> d.plusDays(1))
         .filter(Utils::isWorkingDay)
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
private static boolean isWorkingDay(LocalDate d) {
   return d.getDayOfWeek() != DayOfWeek.SATURDAY &&
          d.getDayOfWeek() != DayOfWeek.SUNDAY;
}
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
   return Stream.iterate(startDate, d -> d.plusDays(1))
         .filter(Utils::isWorkingDay)
         .limit(dayCount)
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
private static boolean isWorkingDay(LocalDate d) {
   return d.getDayOfWeek() != DayOfWeek.SATURDAY &&
          d.getDayOfWeek() != DayOfWeek.SUNDAY;
}
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
   return Stream.iterate(startDate, d -> d.plusDays(1))
         .filter(Utils::isWorkingDay)
         .limit(dayCount)
         .collect(toList());
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
private static boolean isWorkingDay(LocalDate d) {
   return d.getDayOfWeek() != DayOfWeek.SATURDAY &&
          d.getDayOfWeek() != DayOfWeek.SUNDAY;
}
List<LocalDate> generateWorkingDays(LocalDate startDate, int dayCount) {
   return Stream.iterate(startDate, d -> d.plusDays(1))
         .filter(Utils::isWorkingDay)
         .limit(dayCount)
         .collect(toList());
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);


List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);


List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation		of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);


List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);


List<WorkPe                                                    dayCount) {
   List<Loca                                                     dayCount);
   return ge                                                   eriodLength,
                                                               eriodLength);
}
```

LocalTime factory methods
of(int hour, int minute)
of(int hour, int minute, int second)
of(int hour, int minute, int second, int nanoOfSecond)
ofNanoOfDay(long nanoOfDay)
ofSecondOfDay(long secondOfDay)
now(), **now**(**Clock**), **now**(**ZoneId**)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);


List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);

List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);

List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Duration workPeriodLength = Duration.ofHours(3).plusMinutes(30);
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Duration workPeriodLength = Duration.ofHours(3).plusMinutes(30);
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Duration workPeriodLength = Duration.ofHours(3).plusMinutes(30);
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Dur
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

Duration factory methods

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Dur
Lis
    List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
    return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                            pmStart, workPeriodLength);
}
```

Duration factory methods
ofDays(long), **ofHours**(long), **ofMinutes**(long), **ofSeconds**(long), **ofMillis**(long), **ofNanos**(long)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Dur
Lis
    return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                                    pmStart, workPeriodLength);
}
```

Duration factory methods
ofDays(long), **ofHours**(long), **ofMinutes**(long), **ofSeconds**(long), **ofMillis**(long), **ofNanos**(long)
of(long, **TemporalUnit**)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Dur
Lis
    L
    I
                                                    pmStart, workPeriodLength);
}
```

Duration factory methods
ofDays(long), **ofHours**(long), **ofMinutes**(long), **ofSeconds**(long), **ofMillis**(long), **ofNanos**(long)
of(long, **TemporalUnit**)
between(**Temporal**, **Temporal**)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Dur
Lis
    L
    I
}
```

Duration factory methods
ofDays(long), **ofHours**(long), **ofMinutes**(long), **ofSeconds**(long), **ofMillis**(long), **ofNanos**(long)
of(long, **TemporalUnit**)
between(**Temporal**, **Temporal**)
from(**TemporalAmount**)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Duration workPeriodLength = Duration.ofHours(3).plusMinutes(30);
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Duration workPeriodLength = Duration.ofHours(3).plusMinutes(30);
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Duration workPeriodLength = Duration.ofHours(3).plusMinutes(30);
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmSt
Duration workP                                          utes(30);
List<WorkPerio                                          t dayCount) {
   List<LocalD                                          te, dayCount);
   return gene                                          kPeriodLength,
                                                        pmStart, workPeriodLength);
}
```

Duration adjustment methods
plusDays(long), plusHours(long), plusMinutes(long), plusSeconds(long),plusMillis(long), plusNanos(long)
minusDays(long), minusHours(long),...
plus(Duration), minus(Duration)
minus(long, TemporalUnit), plus(long, TemporalUnit)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
LocalTime amStart = LocalTime.of(9, 0);
LocalTime pmStart = LocalTime.of(13, 30);
Duration workPeriodLength = Duration.ofHours(3).plusMinutes(30);
List<WorkPeriod> generateWorkPeriods(LocalDate date, int dayCount) {
   List<LocalDate> workingDays = generateWorkingDays(date, dayCount);
   return generateWorkPeriods(workingDays, amStart, workPeriodLength,
                                           pmStart, workPeriodLength);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	`of(int,int)`	`of(int,int,int)`		`ofHours(long)`
Field access		`getDayOfWeek()`	`getDayOfWeek()`	`plusMinutes(long)`
Adjustment		`plusDays(long)`		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	`of(int,int)`	`of(int,int,int)`		`ofHours(long)`
Field access		`getDayOfWeek()`	`getDayOfWeek()`	`plusMinutes(long)`
Adjustment		`plusDays(long)`		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
       LocalDateTime thisAmStart = LocalDateTime.of(d, amStart);
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	`of(int,int)`	`of(int,int,int)`		`ofHours(long)`
Field access		`getDayOfWeek()`	`getDayOfWeek()`	`plusMinutes(long)`
Adjustment		`plusDays(long)`		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
      LocalDateTime thisAmStart = LocalDateTime.of(d, amStart)
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)		ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
       LocalDateTime thisAmStart = LocalDateTime.of(d, amStart)
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	`of(int,int)`	`of(int,int,int)`	`of(LocalDate,LocalTime)`	`ofHours(long)`
Field access		`getDayOfWeek()`	`getDayOfWeek()`	`plusMinutes(long)`
Adjustment		`plusDays(long)`		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                                                                    ) {
    for (LocalDate d : days) {
        LocalDateTime thisAmStart = LocalDateTime.of(d, amStart)
```

LocalDateTime factory methods

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                                                                    ) {
    L
    f
        LocalDateTime thisAmStart = LocalDateTime.of(d, amStart)
```

LocalDateTime factory methods

of(int year, int month, int dayOfMonth, int hour, int minute, int second, int nanoOfSecond)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                                                                    ) {
    L
    f
        LocalDateTime thisAmStart = LocalDateTime.of(d, amStart)
```

LocalDateTime factory methods

of(int year, int month, int dayOfMonth, int hour, int minute, int second)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                                                                    ) {
    L
    f
        LocalDateTime thisAmStart = LocalDateTime.of(d, amStart)
```

LocalDateTime factory methods

of(int year, int month, int dayOfMonth, int hour, int minute)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                                                                    {
```

LocalDateTime factory methods
of(int year, int month, int dayOfMonth, int hour, int minute)
of(int year, **Month** month, int dayOfMonth, int hour, int minute, int second, int nanoOfSecond)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                                                                    {
```

LocalDateTime factory methods
of(int year, int month, int dayOfMonth, int hour, int minute)
of(int year, **Month** month, int dayOfMonth, int hour, int minute, int second)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                                                                    {
```

LocalDateTime factory methods
of(int year, int month, int dayOfMonth, int hour, int minute)
of(int year, **Month** month, int dayOfMonth, int hour, int minute)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
```

LocalDateTime factory methods
of(int year, int month, int dayOfMonth, int hour, int minute)
of(int year, **Month** month, int dayOfMonth, int hour, int minute)
of(**LocalDate** date, **LocalTime** time)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
```

LocalDateTime factory methods
of(int year, int month, int dayOfMonth, int hour, int minute)
of(int year, **Month** month, int dayOfMonth, int hour, int minute)
of(**LocalDate** date, **LocalTime** time)
now(), **now(Clock)**, **now(ZoneId)**

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                                                                    {
```

LocalDateTime factory methods
of(int year, int month, int dayOfMonth, int hour, int minute)
of(int year, **Month** month, int dayOfMonth, int hour, int minute)
of(**LocalDate** date, **LocalTime** time)
now(), **now(Clock)**, **now(ZoneId)**
ofEpochSecond(long epochSecond, int nanoOfSecond, **ZoneOffset** offset)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                                                                    {
```

LocalDateTime factory methods
of(int year, int month, int dayOfMonth, int hour, int minute)
of(int year, **Month** month, int dayOfMonth, int hour, int minute)
of(**LocalDate** date, **LocalTime** time)
now(), **now(Clock)**, **now(ZoneId)**
ofEpochSecond(long epochSecond, int nanoOfSecond, **ZoneOffset** offset)
ofInstant(**Instant** instant, **ZoneId** zone)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
       LocalDateTime thisAmStart = LocalDateTime.of(d, amStart);
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
      LocalDateTime thisAmStart = LocalDateTime.of(d, amStart);
      periods.add(new WorkPeriod(thisAmStart, thisAmStart.plus(amDuration)));
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
      LocalDateTime thisAmStart = LocalDateTime.of(d, amStart);
      periods.add(new WorkPeriod(thisAmStart, thisAmStart.plus(amDuration)));
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)		
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
      LocalDateTime thisAmStart = LocalDateTime.of(d, amStart);
      periods.add(new WorkPeriod(thisAmStart, thisAmStart.plus(amDuration)));
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> peri
   for (LocalDate d : da
      LocalDateTime thi
      periods.add(new W
```

LocalDateTime adjustment methods
plusDays(long), **plusHours**(long), **plusMinutes**(long), **plusSeconds**(long),**plusMillis**(long), **plusNanos**(long)
minusDays(long), **minusHours**(long),...
plus(**Duration**), **minus**(**Duration**)
minus(long, **TemporalUnit**), **plus**(long, **TemporalUnit**)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
      LocalDateTime thisAmStart = LocalDateTime.of(d, amStart);
      periods.add(new WorkPeriod(thisAmStart, thisAmStart.plus(amDuration)));
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
      LocalDateTime thisAmStart = LocalDateTime.of(d, amStart);
      periods.add(new WorkPeriod(thisAmStart, thisAmStart.plus(amDuration)));
      LocalDateTime thisPmStart = LocalDateTime.of(d, pmStart);
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	`of(int,int)`	`of(int,int,int)`	`of(LocalDate,LocalTime)`	`ofHours(long)`
Field access		`getDayOfWeek()`	`getDayOfWeek()`	`plusMinutes(long)`
Adjustment		`plusDays(long)`	`plus(Duration)`	
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
      LocalDateTime thisAmStart = LocalDateTime.of(d, amStart);
      periods.add(new WorkPeriod(thisAmStart, thisAmStart.plus(amDuration)));
      LocalDateTime thisPmStart = LocalDateTime.of(d, pmStart);
      periods.add(new WorkPeriod(thisPmStart, thisPmStart.plus(pmDuration)));
```

The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
List<WorkPeriod> generateWorkPeriods(List<LocalDate> days, LocalTime amStart, Duration amDuration,
                                     LocalTime pmStart, Duration pmDuration) {
   List<WorkPeriod> periods = new ArrayList<>();
   for (LocalDate d : days) {
      LocalDateTime thisAmStart = LocalDateTime.of(d, amStart);
      periods.add(new WorkPeriod(thisAmStart, thisAmStart.plus(amDuration)));
      LocalDateTime thisPmStart = LocalDateTime.of(d, pmStart);
      periods.add(new WorkPeriod(thisPmStart, thisPmStart.plus(pmDuration)));
   }
   return periods;
}
```

The `java.time` API — examples

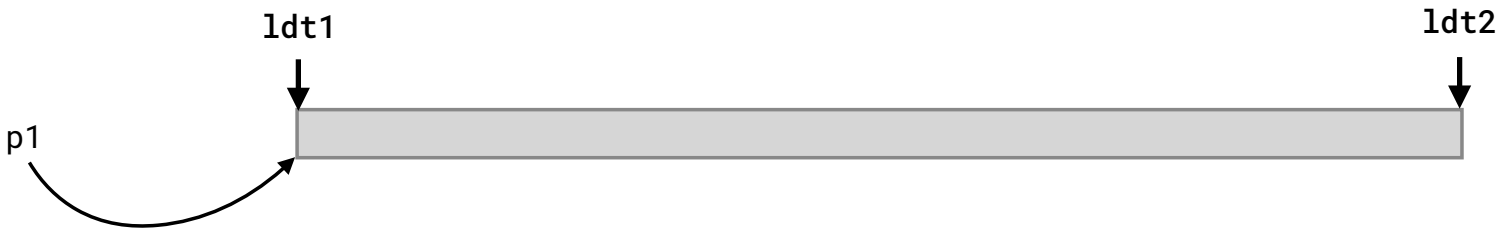
	LocalTime	LocalDate	LocalDateTime	Duration
Creation	`of(int,int)`	`of(int,int,int)`	`of(LocalDate,LocalTime)`	`ofHours(long)`
Field access		`getDayOfWeek()`	`getDayOfWeek()`	`plusMinutes(long)`
Adjustment		`plusDays(long)`	`plus(Duration)`	
Comparison				
Conversion				

Splitting periods

```
WorkPeriod p1 = new WorkPeriod(ldt1,ldt2);
```

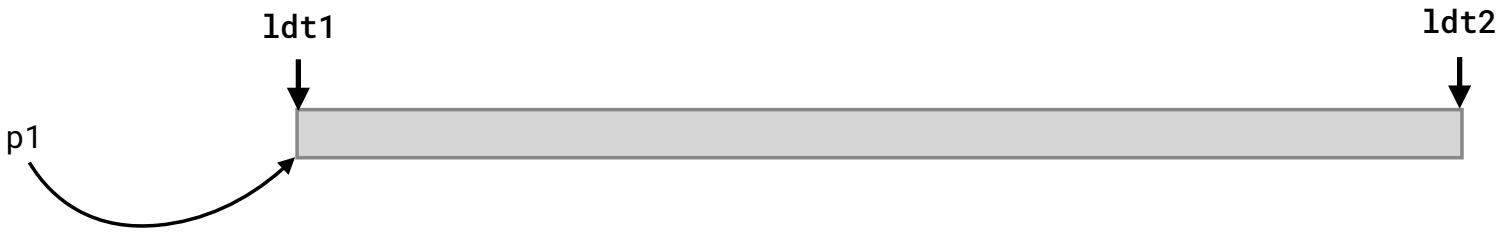
Splitting periods

```
WorkPeriod p1 = new WorkPeriod(ldt1,ldt2);
```



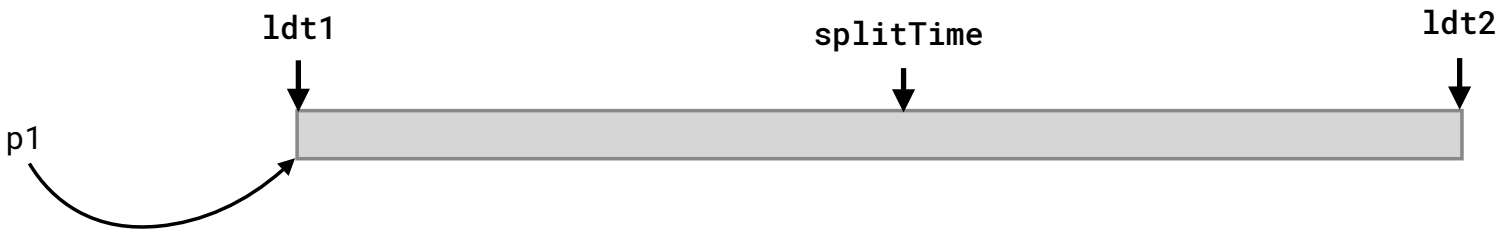
Splitting periods

```
WorkPeriod p1 = new WorkPeriod(ldt1,ldt2);
Optional<WorkPeriod> p2 = p1.split(splitTime);
```



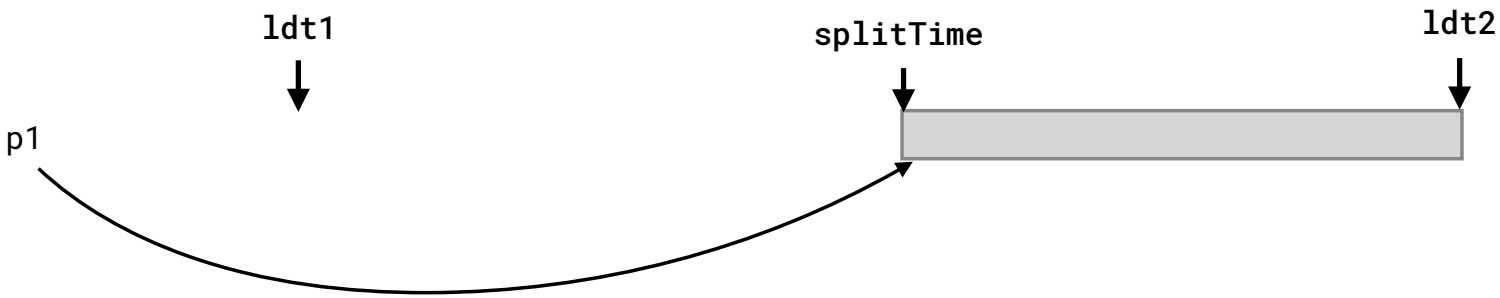
Splitting periods

```
WorkPeriod p1 = new WorkPeriod(ldt1,ldt2);
Optional<WorkPeriod> p2 = p1.split(splitTime);
```



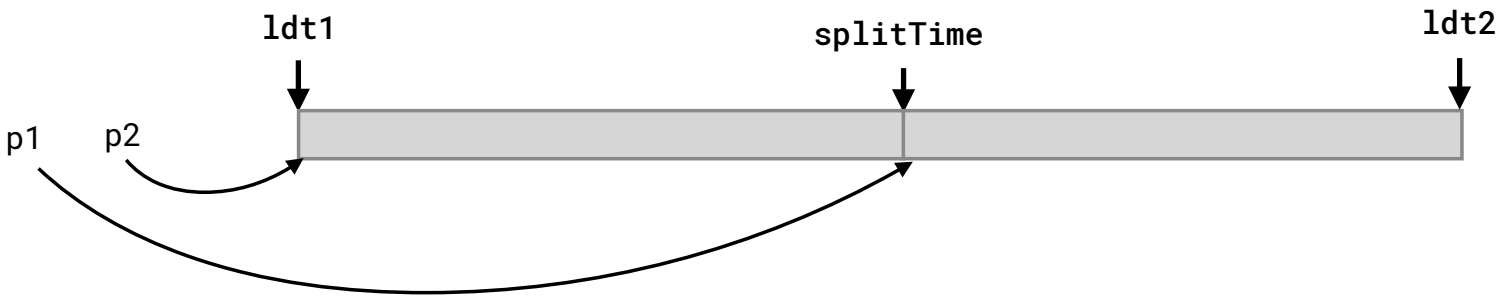
Splitting periods

```
WorkPeriod p1 = new WorkPeriod(ldt1,ldt2);
Optional<WorkPeriod> p2 = p1.split(splitTime);
```



Splitting periods

```
WorkPeriod p1 = new WorkPeriod(ldt1,ldt2);
Optional<WorkPeriod> p2 = p1.split(splitTime);
```



The `java.time` API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	`of(int,int)`	`of(int,int,int)`	`of(LocalDate,LocalTime)`	`ofHours(long)`
Field access		`getDayOfWeek()`	`getDayOfWeek()`	`plusMinutes(long)`
Adjustment		`plusDays(long)`	`plus(Duration)`	
Comparison				
Conversion				

```
@Test
public void basicSplitTest() {
    LocalDateTime startTime = LocalDate.now().atTime(23, 0);
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
@Test
public void basicSplitTest() {
    LocalDateTime startTime = LocalDate.now().atTime(23, 0);
    LocalDateTime endTime = LocalDate.now().plusDays(1).atTime(1, 0);
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
@Test
public void basicSplitTest() {
    LocalDateTime startTime = LocalDate.now().atTime(23, 0);
    LocalDateTime endTime = LocalDate.now().plusDays(1).atTime(1, 0);
    WorkPeriod p = new WorkPeriod(startTime, endTime);
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
@Test
public void basicSplitTest() {
    LocalDateTime startTime = LocalDate.now().atTime(23, 0);
    LocalDateTime endTime = LocalDate.now().plusDays(1).atTime(1, 0);
    WorkPeriod p = new WorkPeriod(startTime, endTime);
    Optional<WorkPeriod> newPeriod = p.split();
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
@Test
public void basicSplitTest() {
    LocalDateTime startTime = LocalDate.now().atTime(23, 0);
    LocalDateTime endTime = LocalDate.now().plusDays(1).atTime(1, 0);
    WorkPeriod p = new WorkPeriod(startTime, endTime);
    Optional<WorkPeriod> newPeriod = p.split();
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
@Test
public void basicSplitTest() {
    LocalDateTime startTime = LocalDate.now().atTime(23, 0);
    LocalDateTime endTime = LocalDate.now().plusDays(1).atTime(1, 0);
    WorkPeriod p = new WorkPeriod(startTime, endTime);
    Optional<WorkPeriod> newPeriod = p.split();

    LocalDateTime midnight = LocalDate.now().plusDays(1).atStartOfDay();
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
@Test
public void basicSplitTest() {
    LocalDateTime startTime = LocalDate.now().atTime(23, 0);
    LocalDateTime endTime = LocalDate.now().plusDays(1).atTime(1, 0);
    WorkPeriod p = new WorkPeriod(startTime, endTime);
    Optional<WorkPeriod> newPeriod = p.split();

    LocalDateTime midnight = LocalDate.now().plusDays(1).atStartOfDay();
    assertEquals(Optional.of(new WorkPeriod(startTime,midnight)), newPeriod);
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
@Test
public void basicSplitTest() {
    LocalDateTime startTime = LocalDate.now().atTime(23, 0);
    LocalDateTime endTime = LocalDate.now().plusDays(1).atTime(1, 0);
    WorkPeriod p = new WorkPeriod(startTime, endTime);
    Optional<WorkPeriod> newPeriod = p.split();

    LocalDateTime midnight = LocalDate.now().plusDays(1).atStartOfDay();
    assertEquals(Optional.of(new WorkPeriod(startTime,midnight)), newPeriod);
    assertEquals(new WorkPeriod(midnight, endTime),p);
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
@Test
public void basicSplitTest() {
    LocalDateTime startTime = LocalDate.now().atTime(23, 0);
    LocalDateTime endTime = LocalDate.now().plusDays(1).atTime(1, 0);
    WorkPeriod p = new WorkPeriod(startTime, endTime);
    Optional<WorkPeriod> newPeriod = p.split();

    LocalDateTime midnight = LocalDate.now().plusDays(1).atStartOfDay();
    assertEquals(Optional.of(new WorkPeriod(startTime,midnight)), newPeriod);
    assertEquals(new WorkPeriod(midnight, endTime),p);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
public Optional<WorkPeriod> split(LocalDateTime splitTime) {
   if (startTime.isBefore(splitTime) && splitTime.isBefore(endTime)) {
      WorkPeriod newPeriod =
         new WorkPeriod(startTime, Duration.between(startTime, splitTime));
      startTime = splitTime;
      return Optional.of(newPeriod);
   } else {
      return Optional.empty();
   }
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
public Optional<WorkPeriod> split(LocalDateTime splitTime) {
   if (startTime.isBefore(splitTime) && splitTime.isBefore(endTime)) {
      WorkPeriod newPeriod =
         new WorkPeriod(startTime, Duration.between(startTime, splitTime));
      startTime = splitTime;
      return Optional.of(newPeriod);
   } else {
      return Optional.empty();
   }
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison				
Conversion				

```
public Optional<WorkPeriod> split(LocalDateTime splitTime) {
   if (startTime.isBefore(splitTime) && splitTime.isBefore(endTime)) {
      WorkPeriod newPeriod =
         new WorkPeriod(startTime, Duration.between(startTime, splitTime));
      startTime = splitTime;
      return Optional.of(newPeriod);
   } else {
      return Optional.empty();
   }
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion				

```
public Optional<WorkPeriod> split(LocalDateTime splitTime) {
    if (startTime.isBefore(splitTime) && splitTime.isBefore(endTime)) {
        WorkPeriod newPeriod =
            new WorkPeriod(splitTime, endTime);
        startTime = splitTime;
        return Optional.of(newPeriod);
    } else {
        return Optional.empty();
    }
}
```

LocalDateTime comparison methods
isAfter(ChronoLocalDateTime<?>)
isBefore(ChronoLocalDateTime<?>)
isEqual(ChronoLocalDateTime<?>)
compareTo(ChronoLocalDateTime<?>)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion				

```
public Optional<WorkPeriod> split() {
    LocalDateTime midnight = startTime.toLocalDate().plusDays(1).atStartOfDay();
    return split(midnight);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion				

```
public Optional<WorkPeriod> split() {
    LocalDateTime midnight = startTime.toLocalDate().plusDays(1).atStartOfDay();
    return split(midnight);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion				

```
public Optional<WorkPeriod> split() {
    LocalDateTime midnight = startTime.toLocalDate().plusDays(1).atStartOfDay();
    return split(midnight);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion			toLocalDate()	

```
public Optional<WorkPeriod> split() {
    LocalDateTime midnight = startTime.toLocalDate().plusDays(1).atStartOfDay();
    return split(midnight);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion			toLocalDate()	

```
public Optional<WorkPeriod> split() {
    LocalDateTime midnight = startTime.toLocalDate().plusDays(1).atStartOfDay();
    return split(midnight);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion		atStartOfDay()	toLocalDate()	

Conversion methods

```
public Optional<WorkPeriod> split() {
    LocalDateTime midnight = startTime.toLocalDate().plusDays(1).atStartOfDay();
    return split(midnight);
}
```

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion		atStartOfDay()	toLocalDate()	

```
pub                                                                    ;
    return split(midnight);
}
```

Conversion methods	
LocalDateTime -> LocalDate/Time	**toLocalDate()**
	toLocalTime()

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion		atStartOfDay()	toLocalDate()	

pub

}

;

Conversion methods	
LocalDateTime -> LocalDate/Time	**toLocalDate**()
	toLocalTime()
LocalDate -> LocalDateTime	**atStartOfDay**()
	atTime(int hour, int minute, int second, int nanos)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion		atStartOfDay()	toLocalDate()	

pub

}

;

Conversion methods	
LocalDateTime -> LocalDate/Time	**toLocalDate**()
	toLocalTime()
LocalDate -> LocalDateTime	**atStartOfDay**()
	atTime(int hour, int minute, int second, int nanos)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion		atStartOfDay()	toLocalDate()	

pub

}

;

Conversion methods	
LocalDateTime -> LocalDate/Time	**toLocalDate**()
	toLocalTime()
LocalDate -> LocalDateTime	**atStartOfDay**()
	atTime(int hour, int minute, int second, int nanos)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion		atStartOfDay()	toLocalDate()	

Conversion methods	
LocalDateTime -> LocalDate/Time	**toLocalDate**()
	toLocalTime()
LocalDate -> LocalDateTime	**atStartOfDay**()
	atTime(int hour, int minute, int second, int nanos)
	atTime(LocalTime)

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion		atStartOfDay()	toLocalDate()	

pub

;

}

Conversion methods	
LocalDateTime -> LocalDate/Time	**toLocalDate()**
	toLocalTime()
LocalDate -> LocalDateTime	**atStartOfDay()**
	atTime(int hour, int minute, int second, int nanos)
	atTime(LocalTime)
LocalTime -> LocalDateTime	**atDate(LocalDate)**

The java.time API — examples

	LocalTime	LocalDate	LocalDateTime	Duration
Creation	of(int,int)	of(int,int,int)	of(LocalDate,LocalTime)	ofHours(long)
Field access		getDayOfWeek()	getDayOfWeek()	plusMinutes(long)
Adjustment		plusDays(long)	plus(Duration)	
Comparison			isBefore(ChronoLocalDateTime)	
Conversion		atStartOfDay()	toLocalDate()	

Splitting a non-empty `WorkPeriod`

Splitting a non-empty `WorkPeriod`

Splitting a non-empty `WorkPeriod`

In the demo for this module, we'll see

Splitting a non-empty `WorkPeriod`

In the demo for this module, we'll see

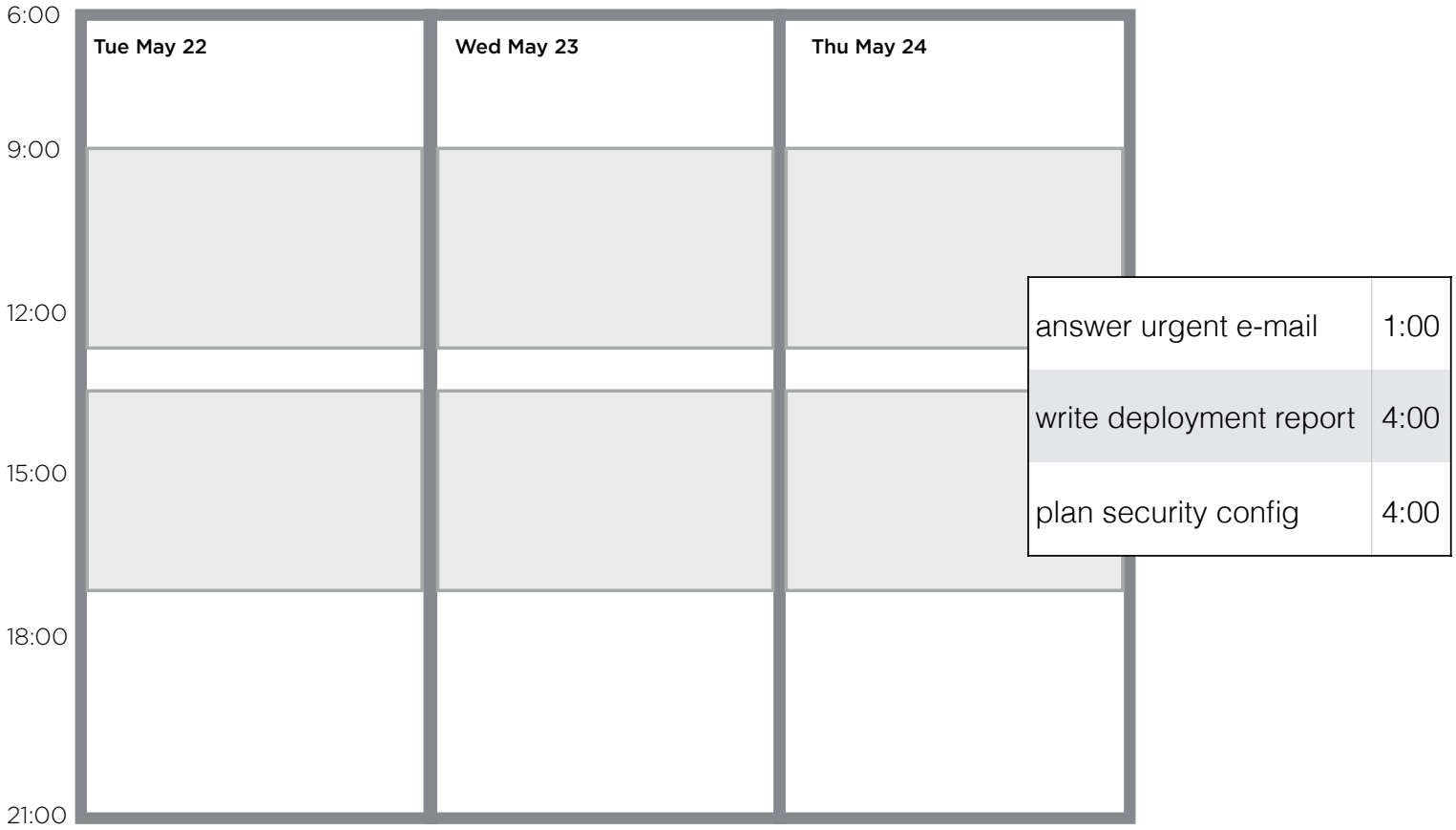
- **a demo class with a main method that creates a `WorkPeriod` containing three tasks**

Splitting a non-empty `WorkPeriod`

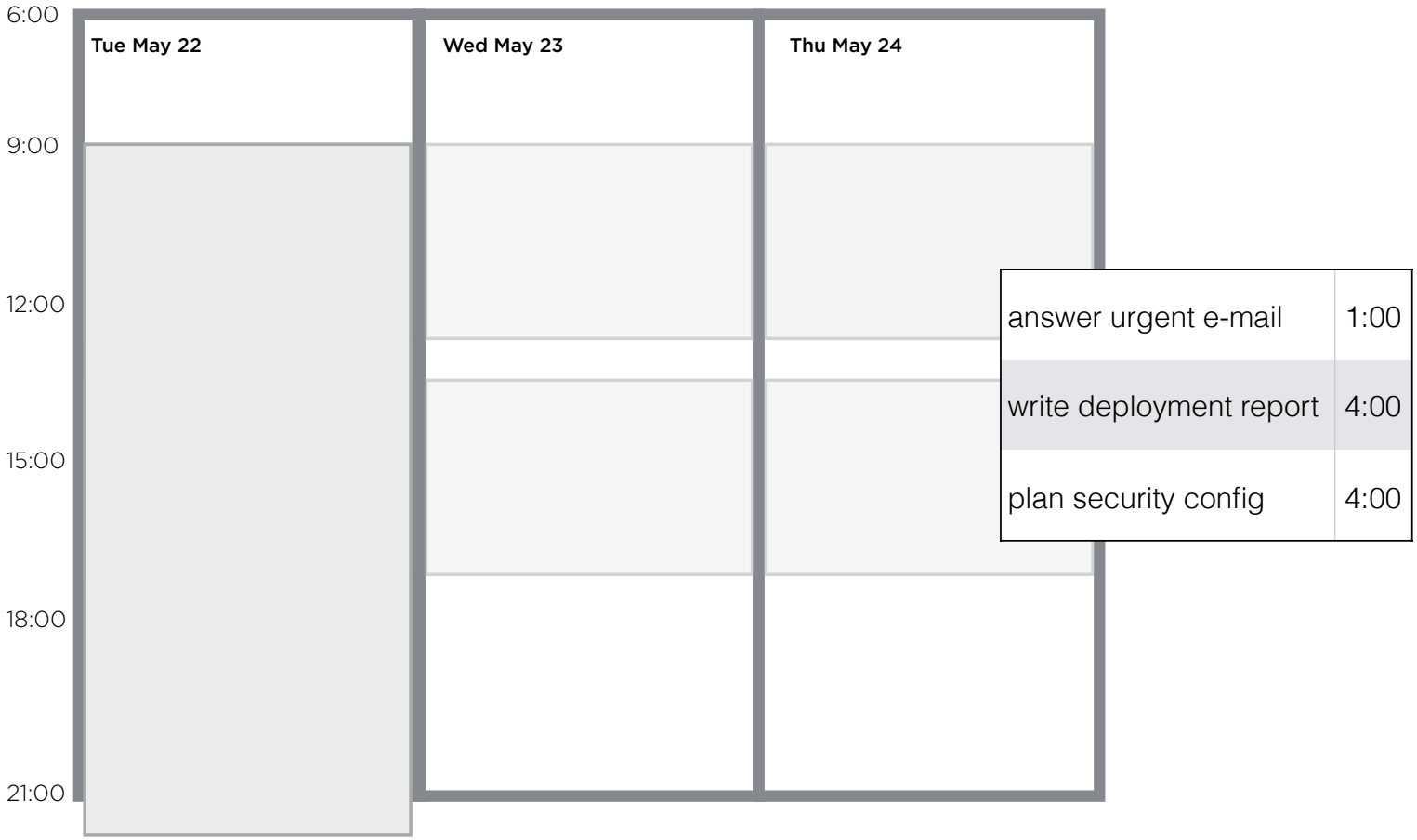
In the demo for this module, we'll see

- a demo class with a main method that creates a `WorkPeriod` containing three tasks
- **how to write the method that splits the `WorkPeriod`**

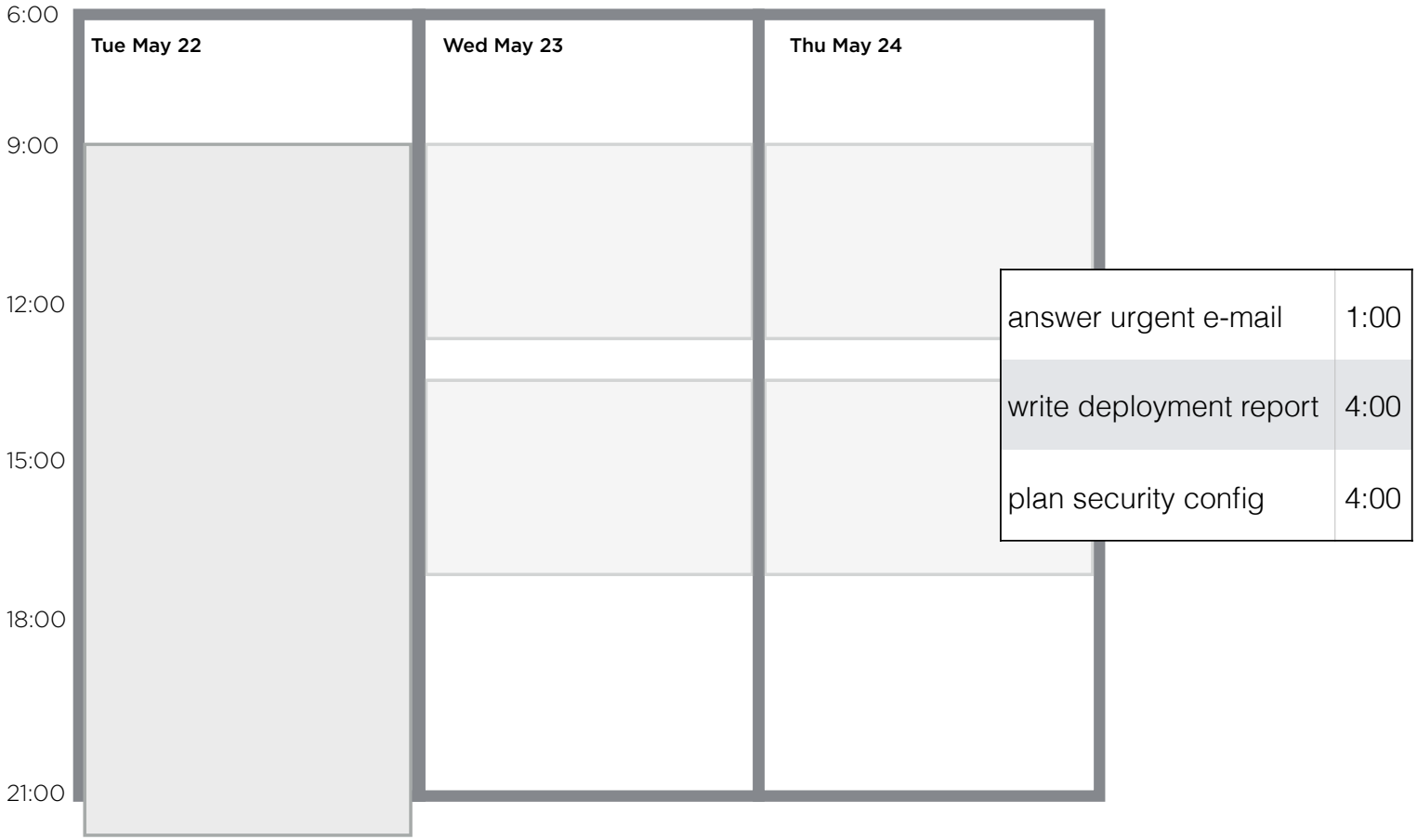
A Task Scheduler



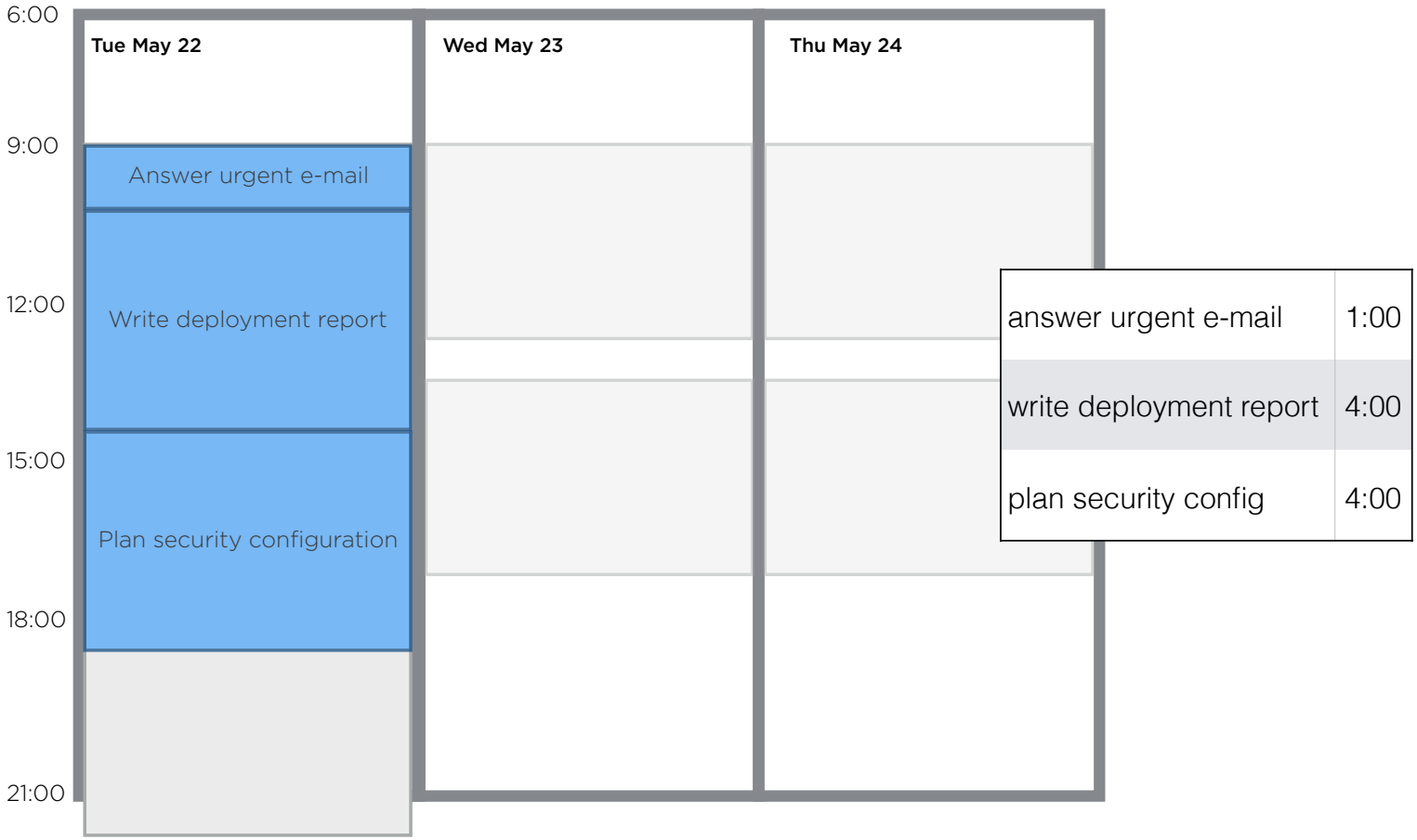
A Task Scheduler



A Task Scheduler

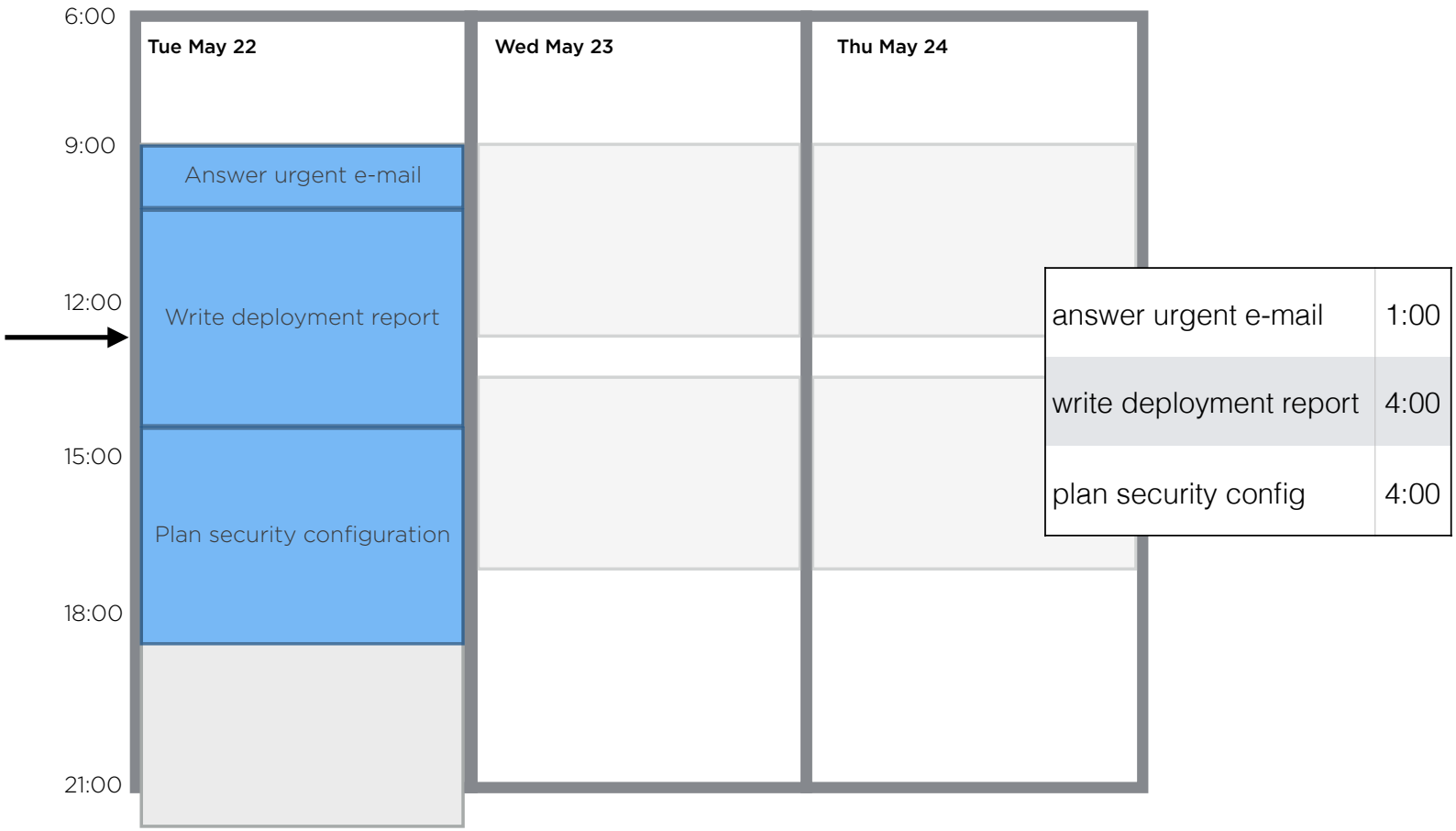


A Task Scheduler

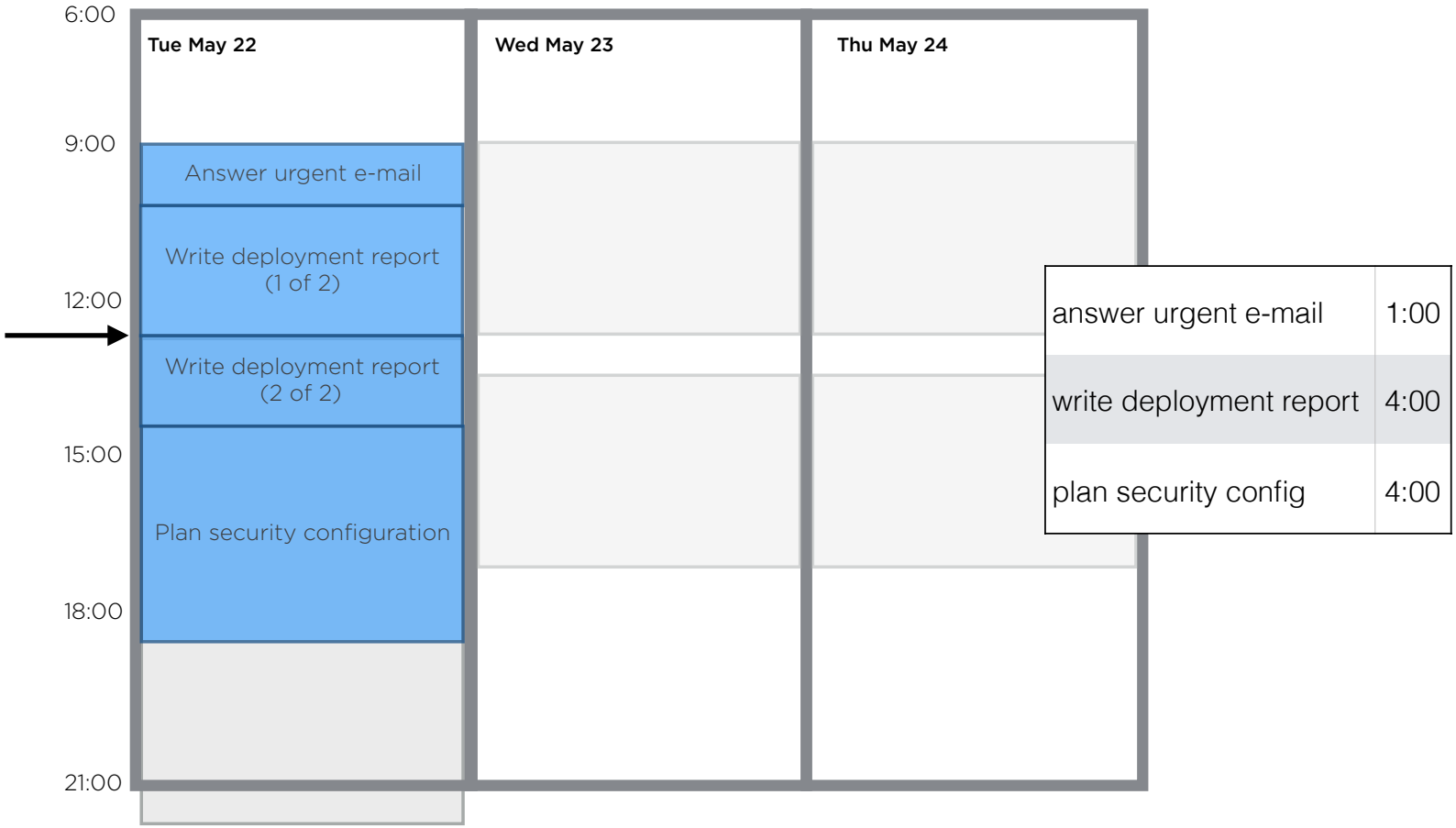


answer urgent e-mail	1:00
write deployment report	4:00
plan security config	4:00

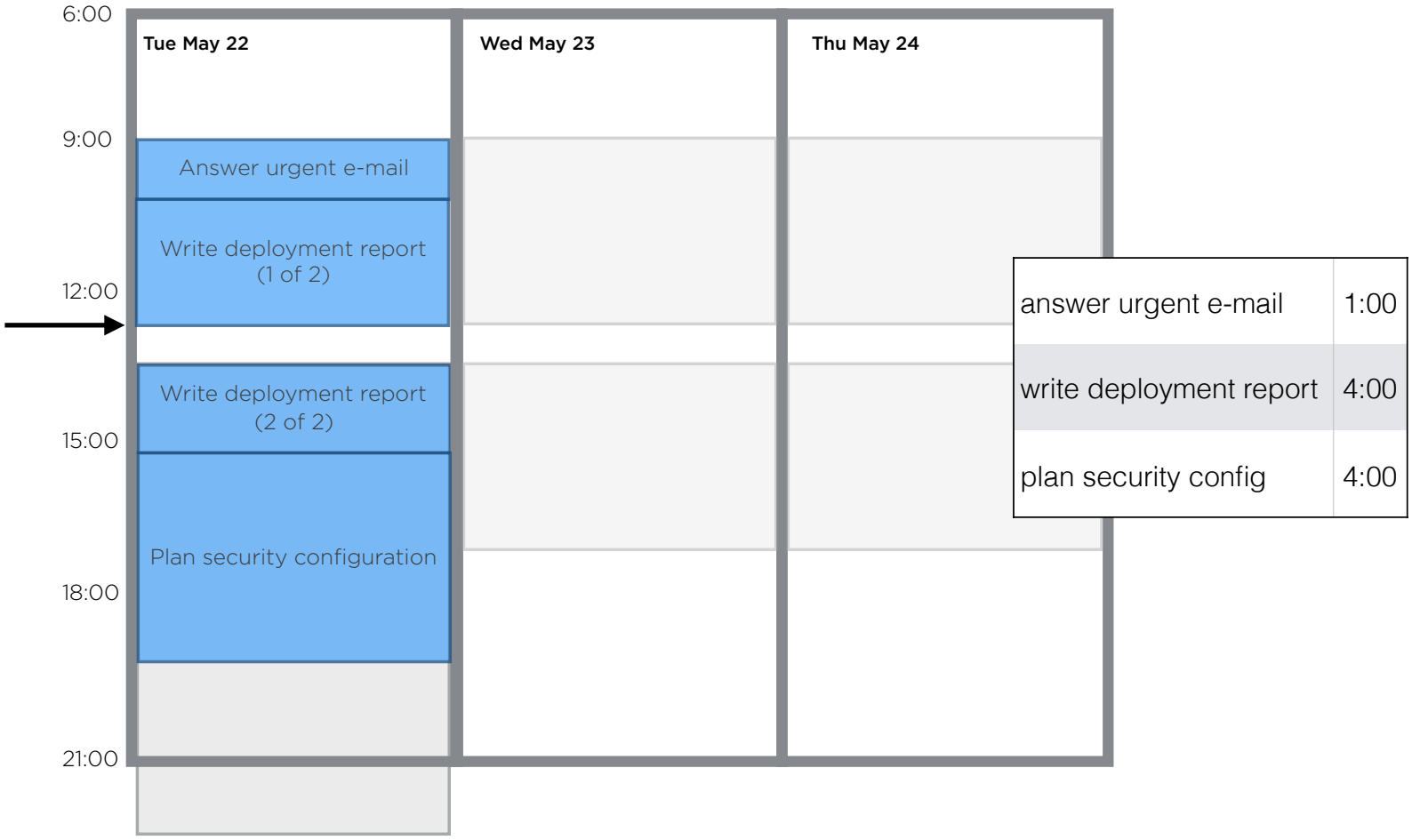
A Task Scheduler



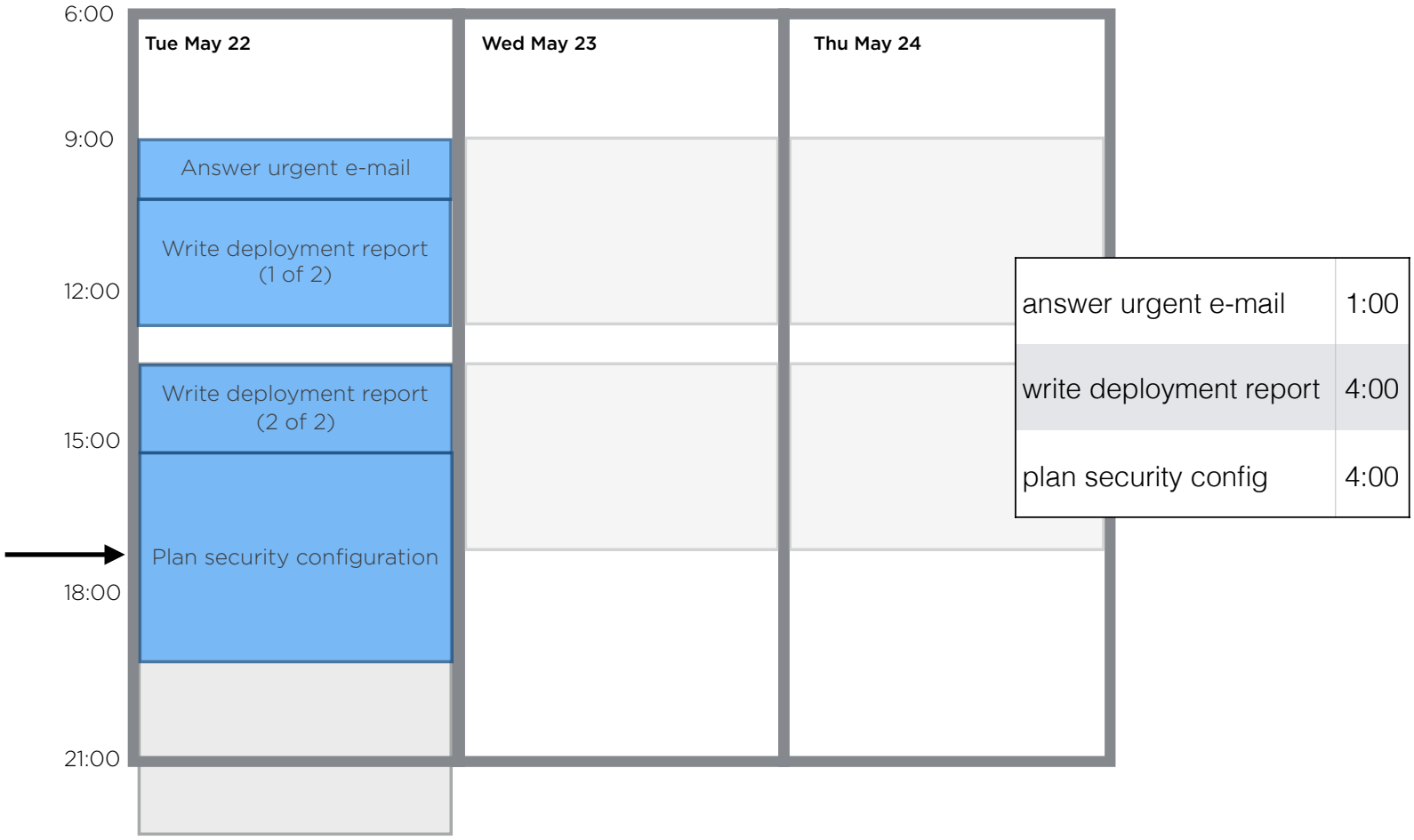
A Task Scheduler



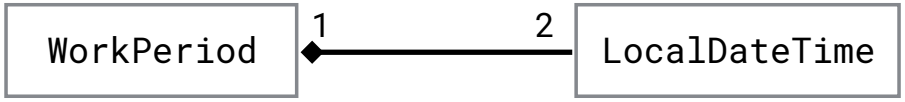
A Task Scheduler



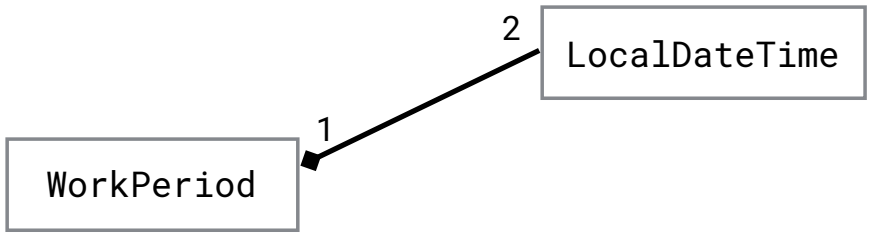
A Task Scheduler



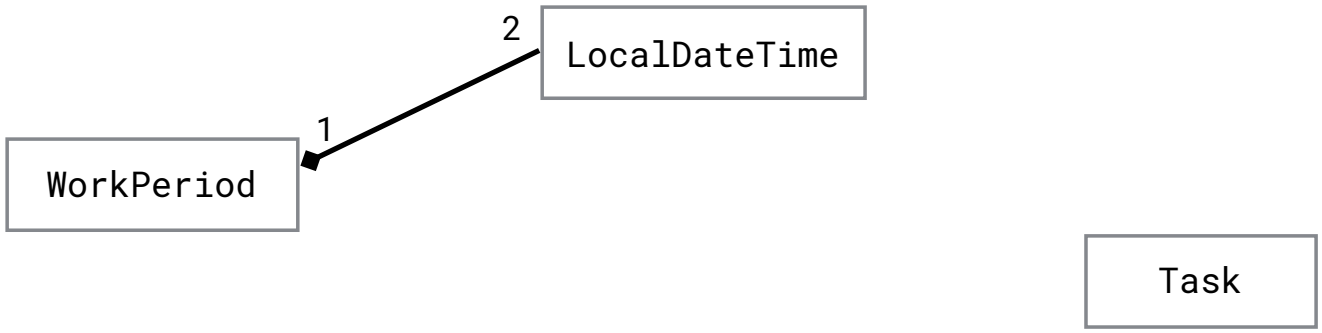
What is a `WorkPeriod`?



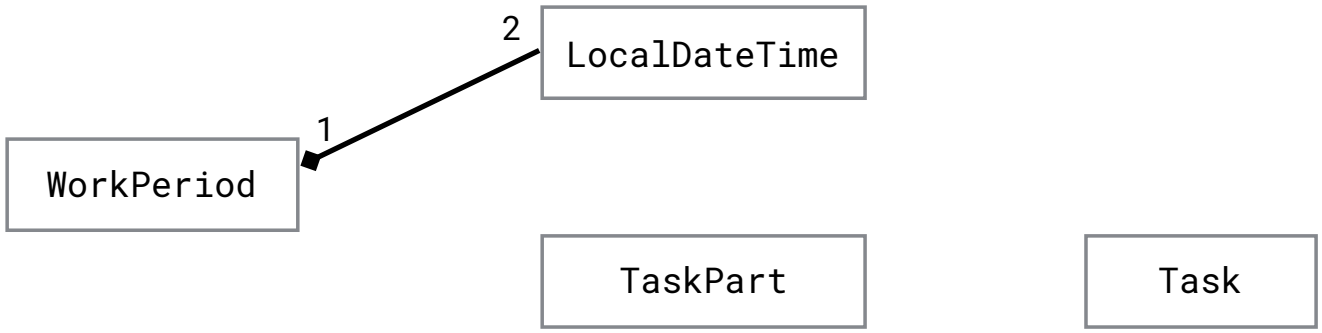
What is a `WorkPeriod`?



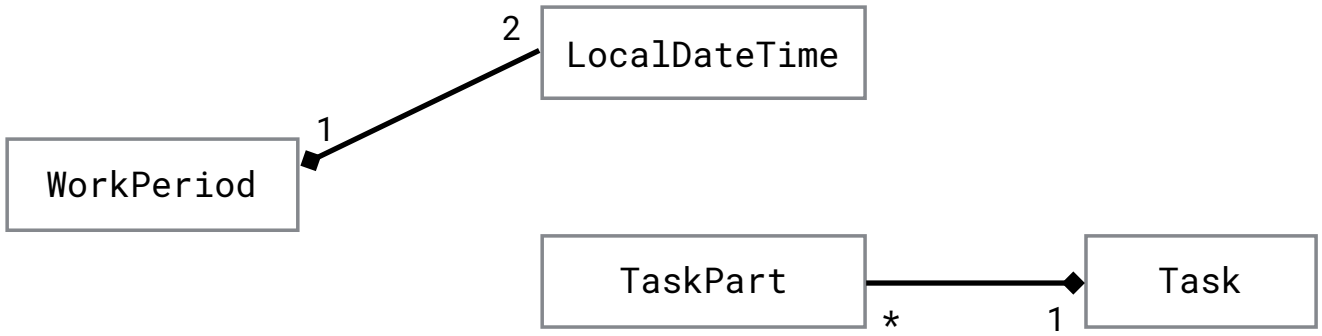
What is a `WorkPeriod`?



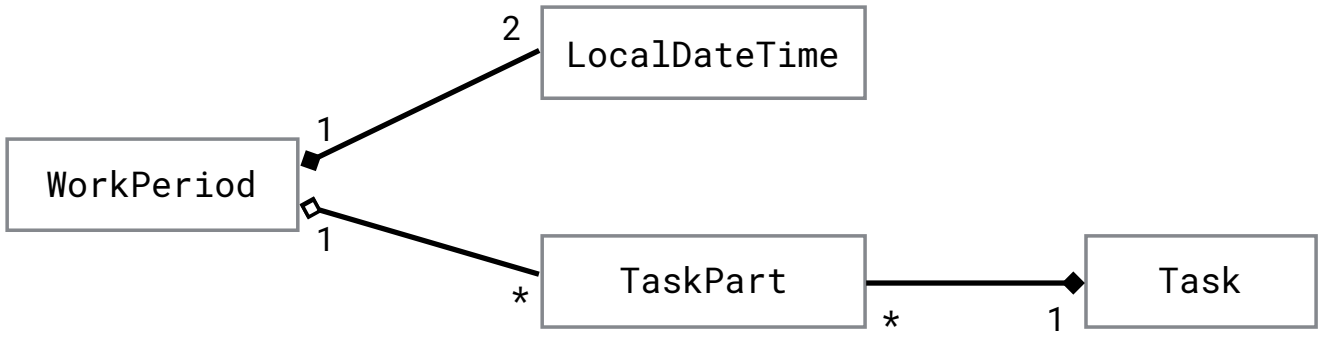
What is a `WorkPeriod`?



What is a `WorkPeriod`?



What is a `WorkPeriod`?



Splitting a task-containing period

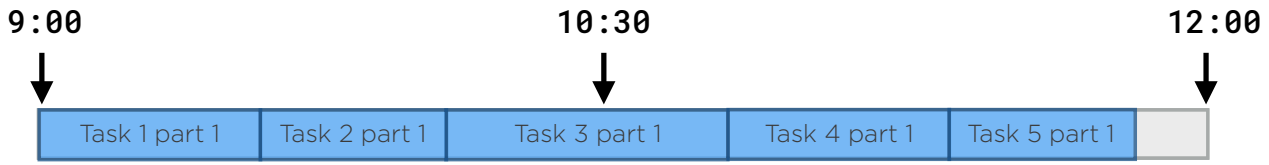
```
LocalDate may22 = LocalDate.of(2018,Month.MAY,22);
LocalDateTime startTime = LocalDateTime.of(may22,LocalTime.of(9,0));
WorkPeriod p1 = new WorkPeriod(startTime,startTime.plus(Duration.ofHours(3));
```

Splitting a task-containing period

```
LocalDate may22 = LocalDate.of(2018,Month.MAY,22);
LocalDateTime startTime = LocalDateTime.of(may22,LocalTime.of(9,0));
WorkPeriod p1 = new WorkPeriod(startTime,startTime.plus(Duration.ofHours(3));

p1.addTaskPart(…); p1.addTaskPart(…); ...
```

Splitting a task-containing period

```
LocalDate may22 = LocalDate.of(2018,Month.MAY,22);
LocalDateTime startTime = LocalDateTime.of(may22,LocalTime.of(9,0));
WorkPeriod p1 = new WorkPeriod(startTime,startTime.plus(Duration.ofHours(3));

p1.addTaskPart(…); p1.addTaskPart(…); ...

LocalDateTime splitTime = startTime.plus(Duration.ofMinutes(90));
```



Using NavigableMap

```
NavigableMap<LocalDateTime,TaskPart> timeToTaskPart;
```

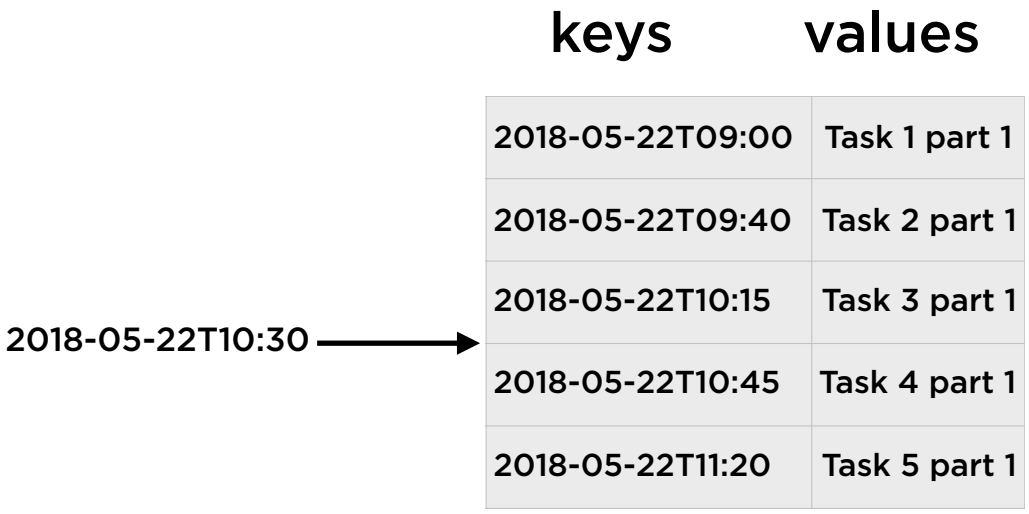
Using NavigableMap

```
NavigableMap<LocalDateTime,TaskPart> timeToTaskPart;
```

keys	values
2018-05-22T09:00	Task 1 part 1
2018-05-22T09:40	Task 2 part 1
2018-05-22T10:15	Task 3 part 1
2018-05-22T10:45	Task 4 part 1
2018-05-22T11:20	Task 5 part 1

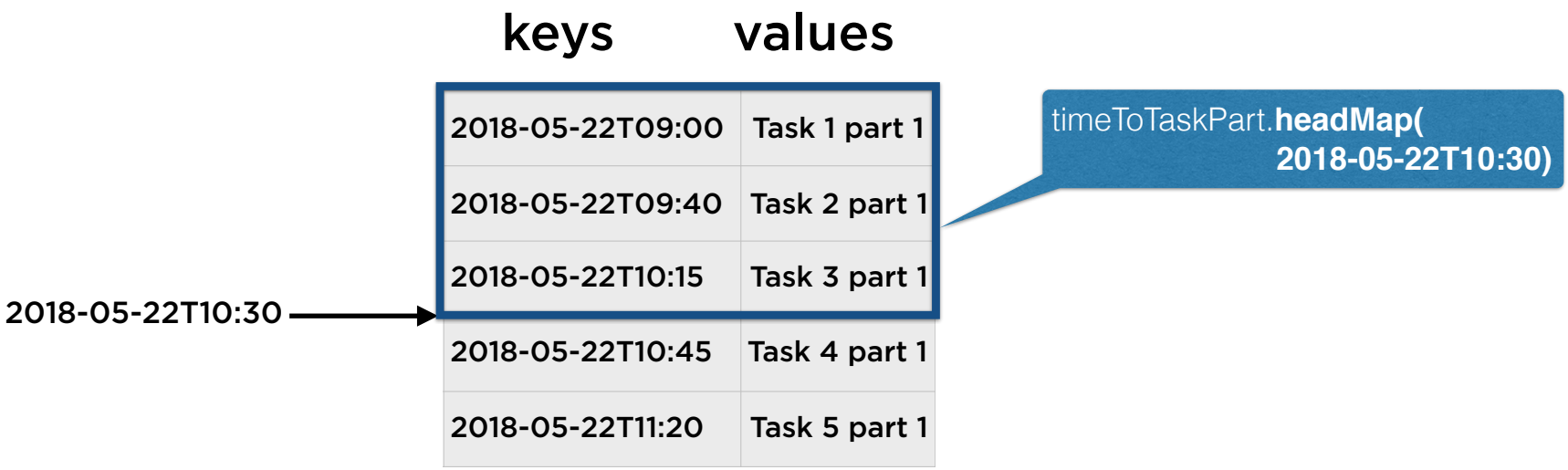
Using NavigableMap

```
NavigableMap<LocalDateTime,TaskPart> timeToTaskPart;
```



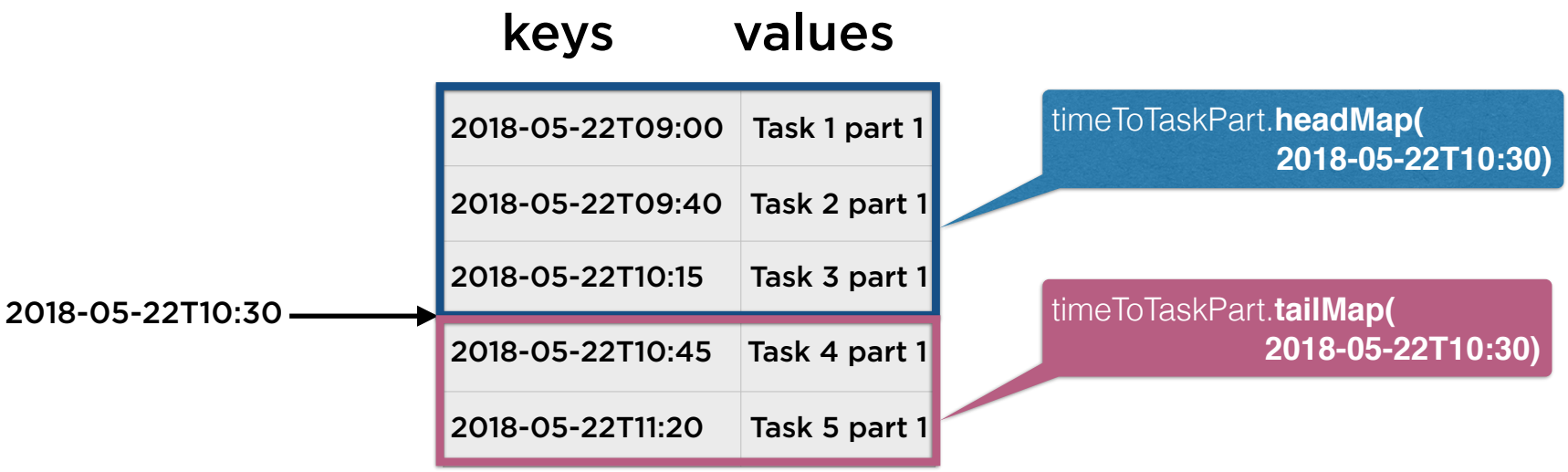
Using NavigableMap

```
NavigableMap<LocalDateTime,TaskPart> timeToTaskPart;
```



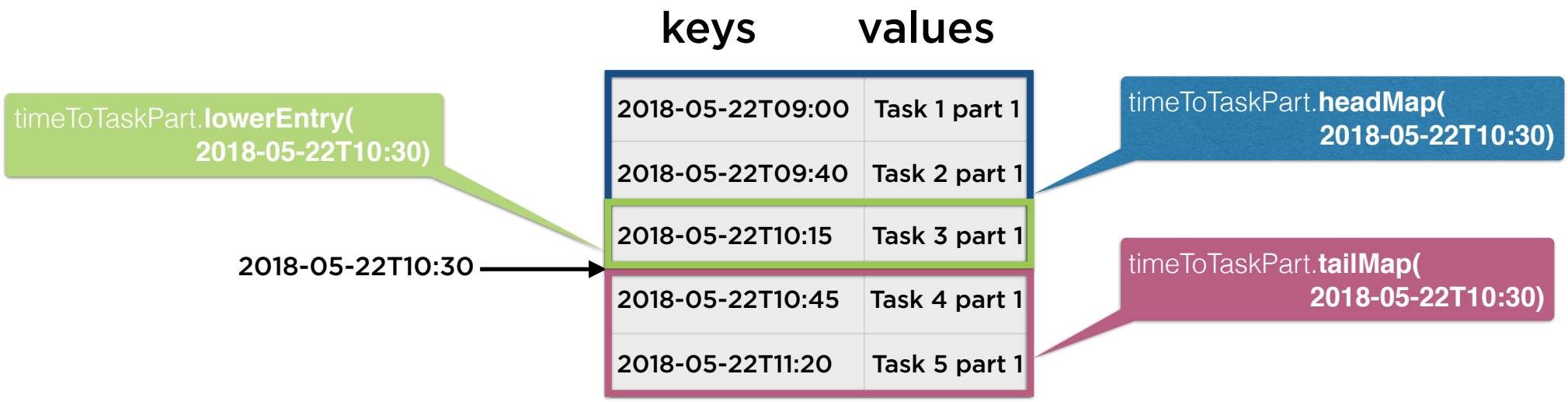
Using NavigableMap

```
NavigableMap<LocalDateTime,TaskPart> timeToTaskPart;
```



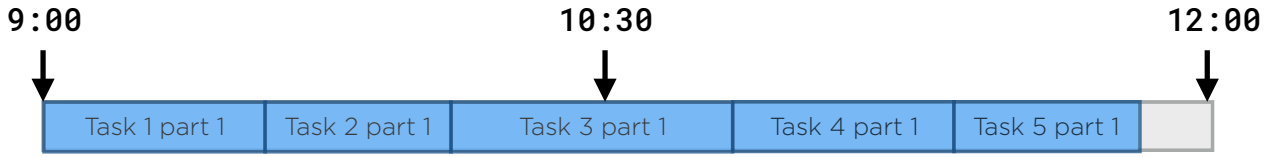
Using NavigableMap

```
NavigableMap<LocalDateTime,TaskPart> timeToTaskPart;
```



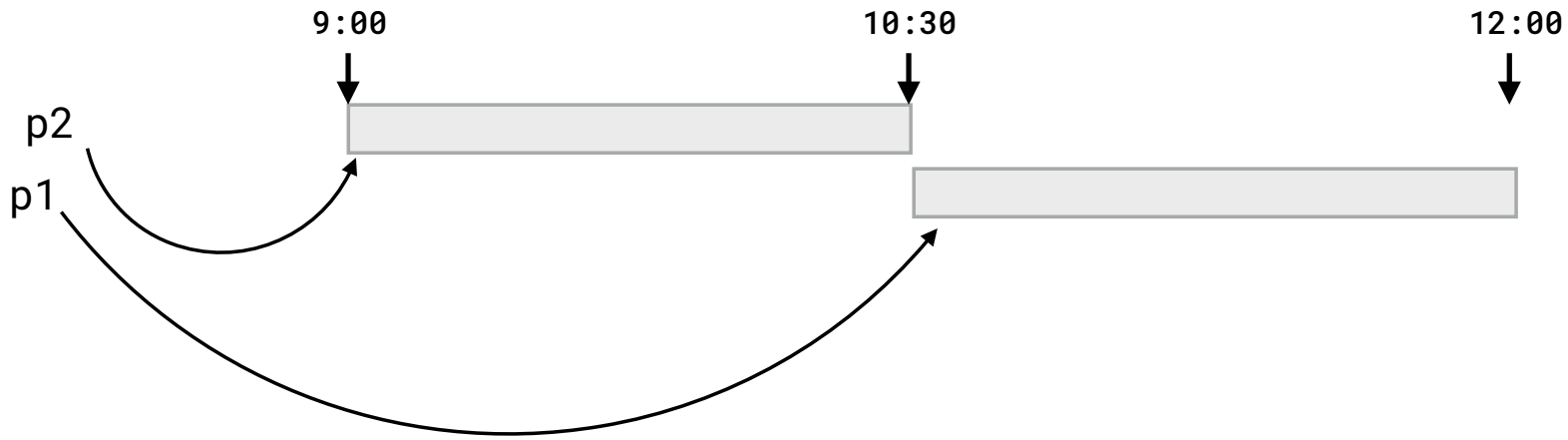
Splitting a task-containing period

```
WorkPeriod p2 = p1.split(splitTime);
```



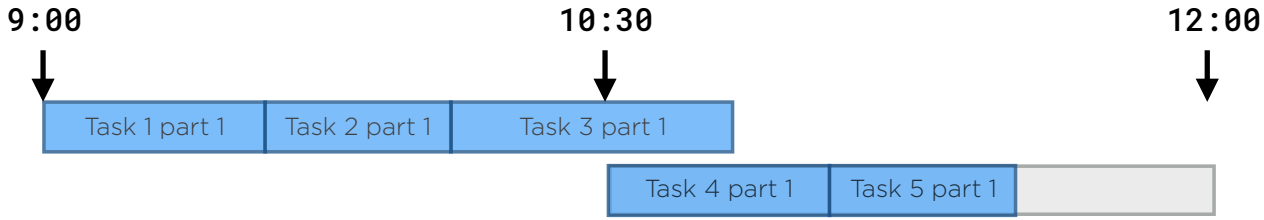
Splitting a task-containing period

```
WorkPeriod p2 = p1.split(splitTime);
```



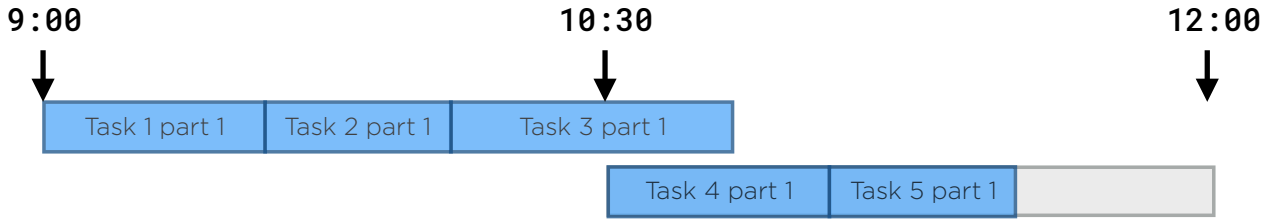
Splitting a task-containing period

```
WorkPeriod p2 = p1.split(splitTime);
```



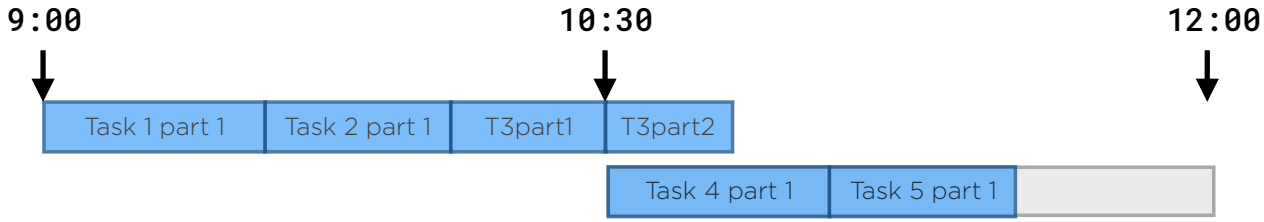
Splitting a task-containing period

```
WorkPeriod p2 = p1.split(splitTime);
```



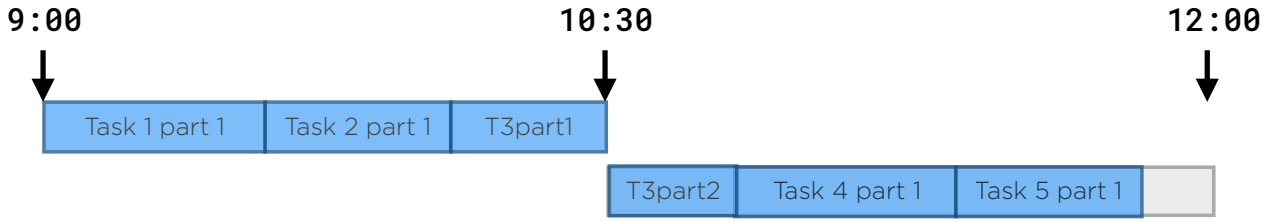
Splitting a task-containing period

```
WorkPeriod p2 = p1.split(splitTime);
```



Splitting a task-containing period

```
WorkPeriod p2 = p1.split(splitTime);
```



Summary

Summary

Summary

Core classes

Summary

Core classes

- `LocalDateTime, LocalDate, LocalTime, Duration`

Summary

Core classes

- `LocalDateTime, LocalDate, LocalTime, Duration`

Core class methods

Summary

Core classes

- `LocalDateTime`, `LocalDate`, `LocalTime`, `Duration`

Core class methods

- **Creation, field access, adjustment, comparison, conversion**

Summary

Core classes

- `LocalDateTime`, `LocalDate`, `LocalTime`, `Duration`

Core class methods

- Creation, field access, adjustment, comparison, conversion

Illustrated in operations on `WorkPeriod`

Summary

Core classes

- `LocalDateTime`, `LocalDate`, `LocalTime`, `Duration`

Core class methods

- Creation, field access, adjustment, comparison, conversion

Illustrated in operations on `WorkPeriod`

- **Creation and splitting**