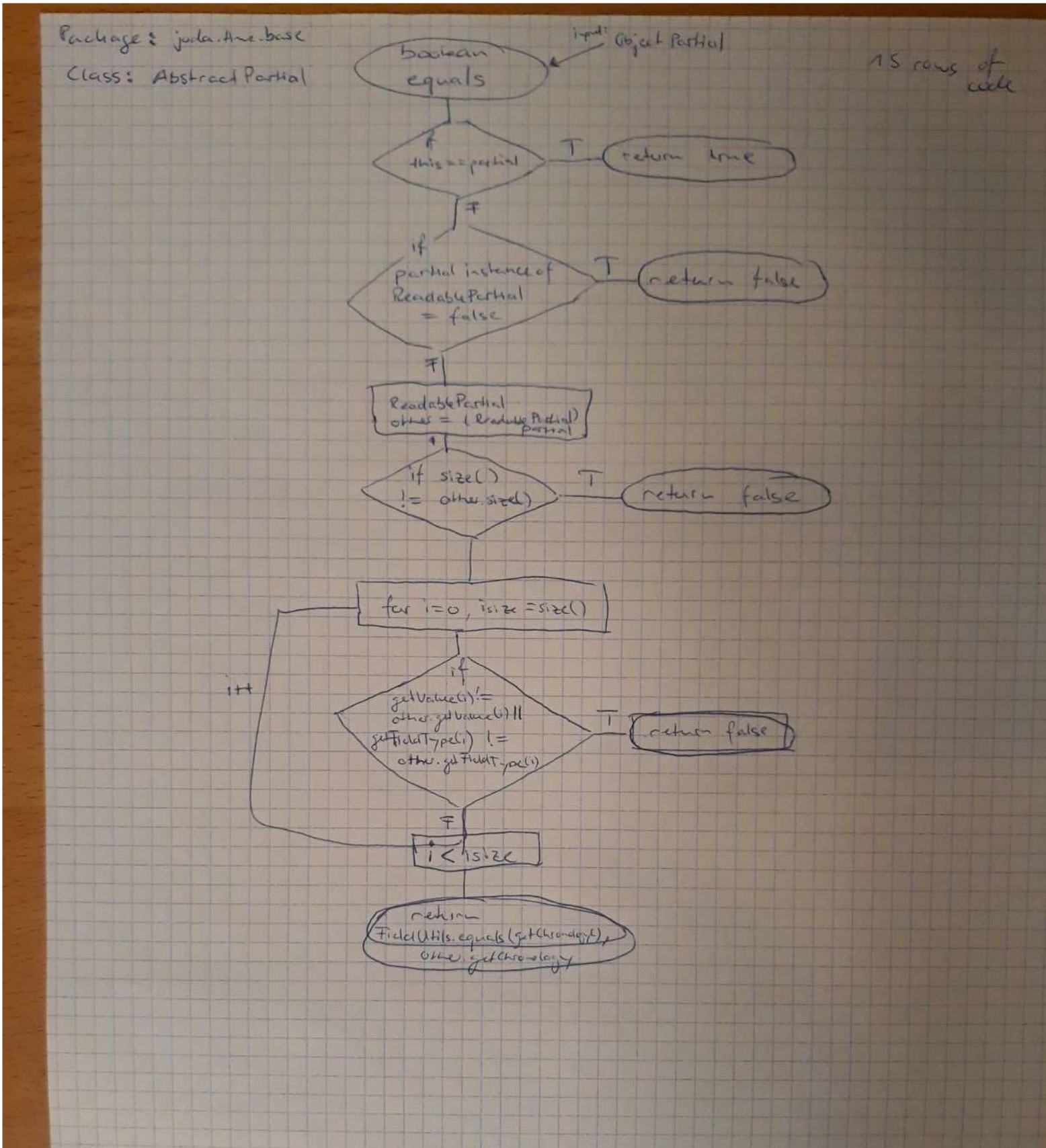
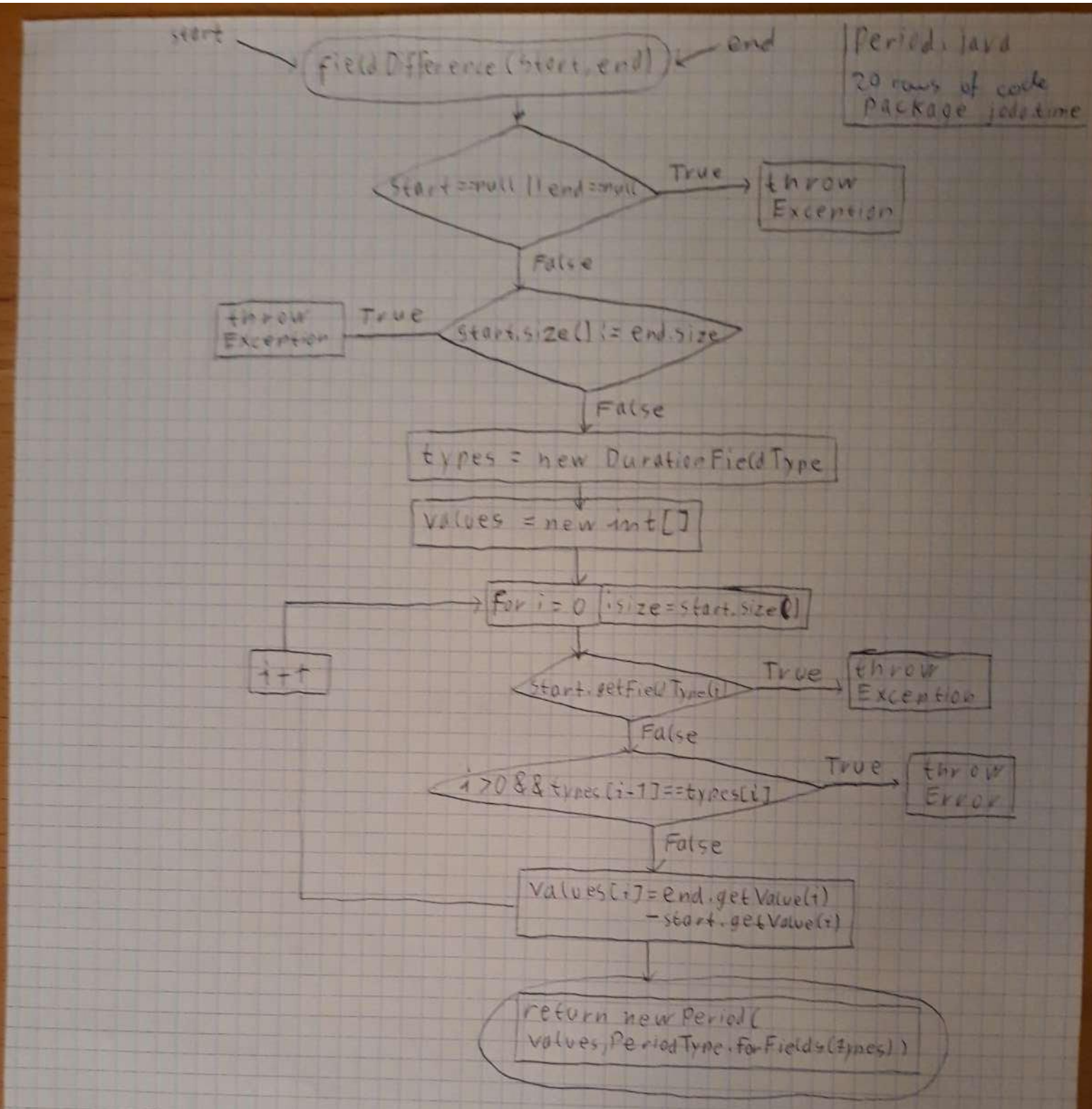
Assignment 1, solution by Nadja Näf, Vincent Müller und Yves Boutellier

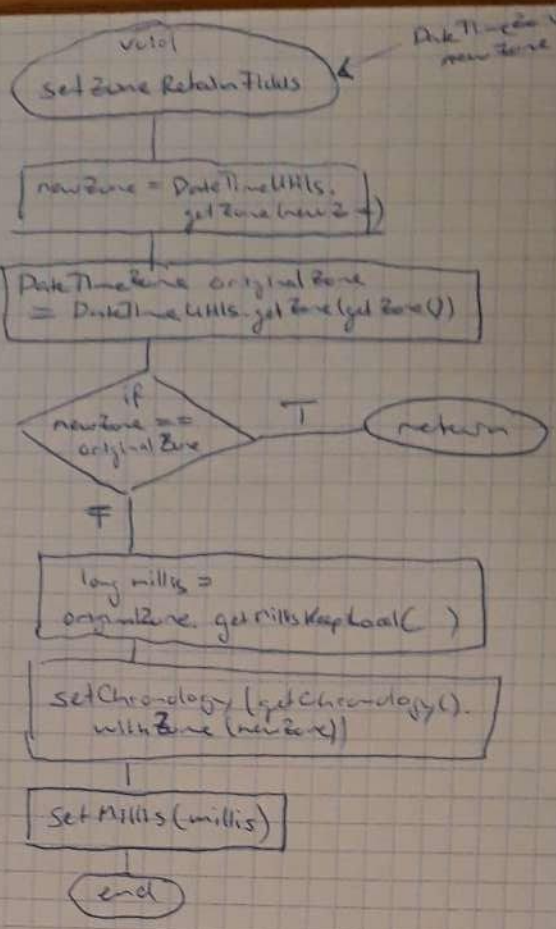
Exercise 2




start
fieldDifference(start, end)
end
Period.java
20 rows of code
package joda.time
start==null || end==null
True
throw Exception
False
throw Exception
True
start.size() != end.size
False
types = new DurationFieldType
values = new int[]
for i = 0
i++
size = start.size()
start.getFieldType(i)
True
throw Exception
False
i > 0 && types[i-1]==types[i]
True
throw Error
False
values[i] = end.getValue(i)
- start.getValue(i)
return new Period(
values, PeriodType.forFields(types))

Package: joda.time

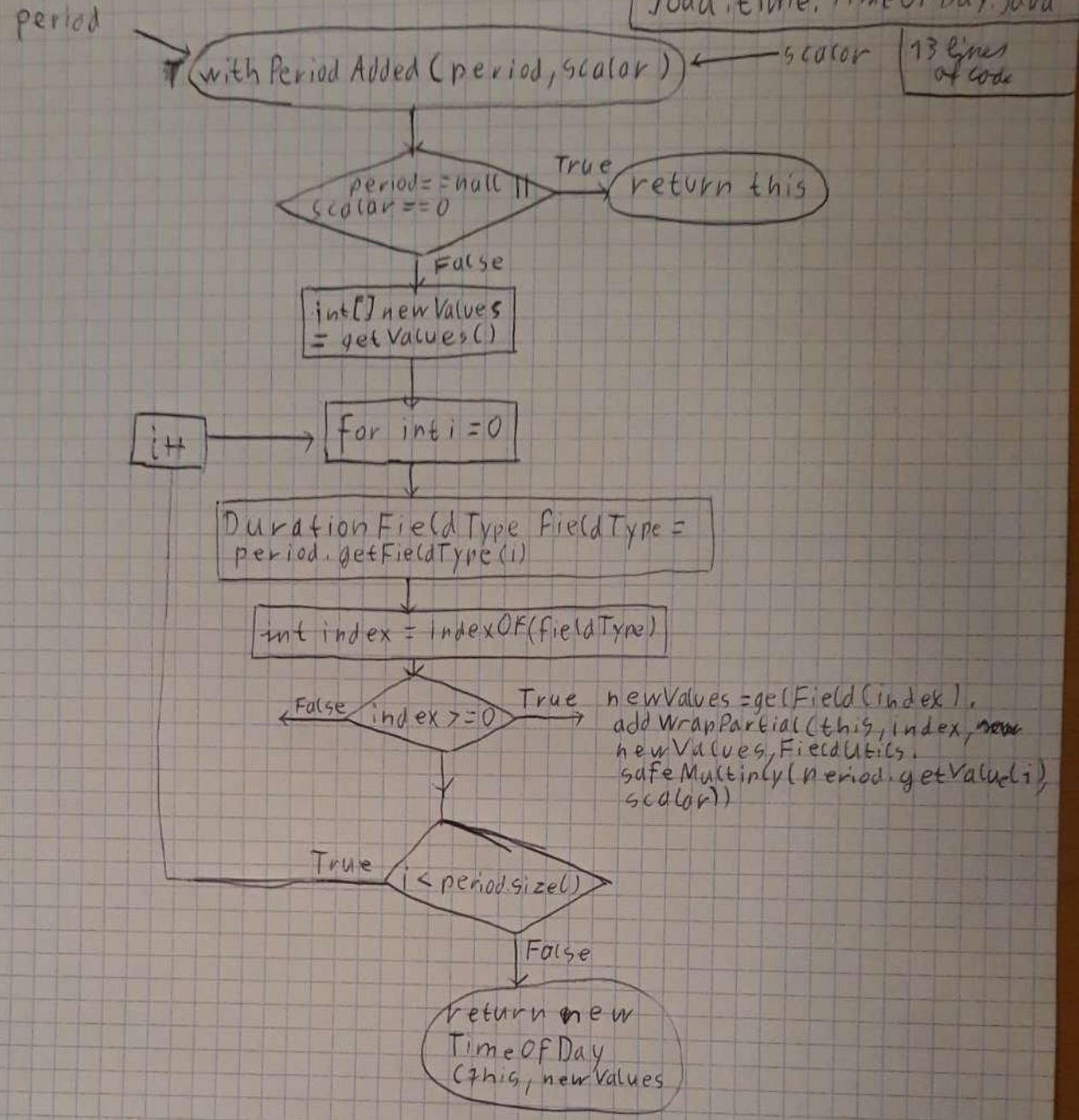
Class: MutableDateTime

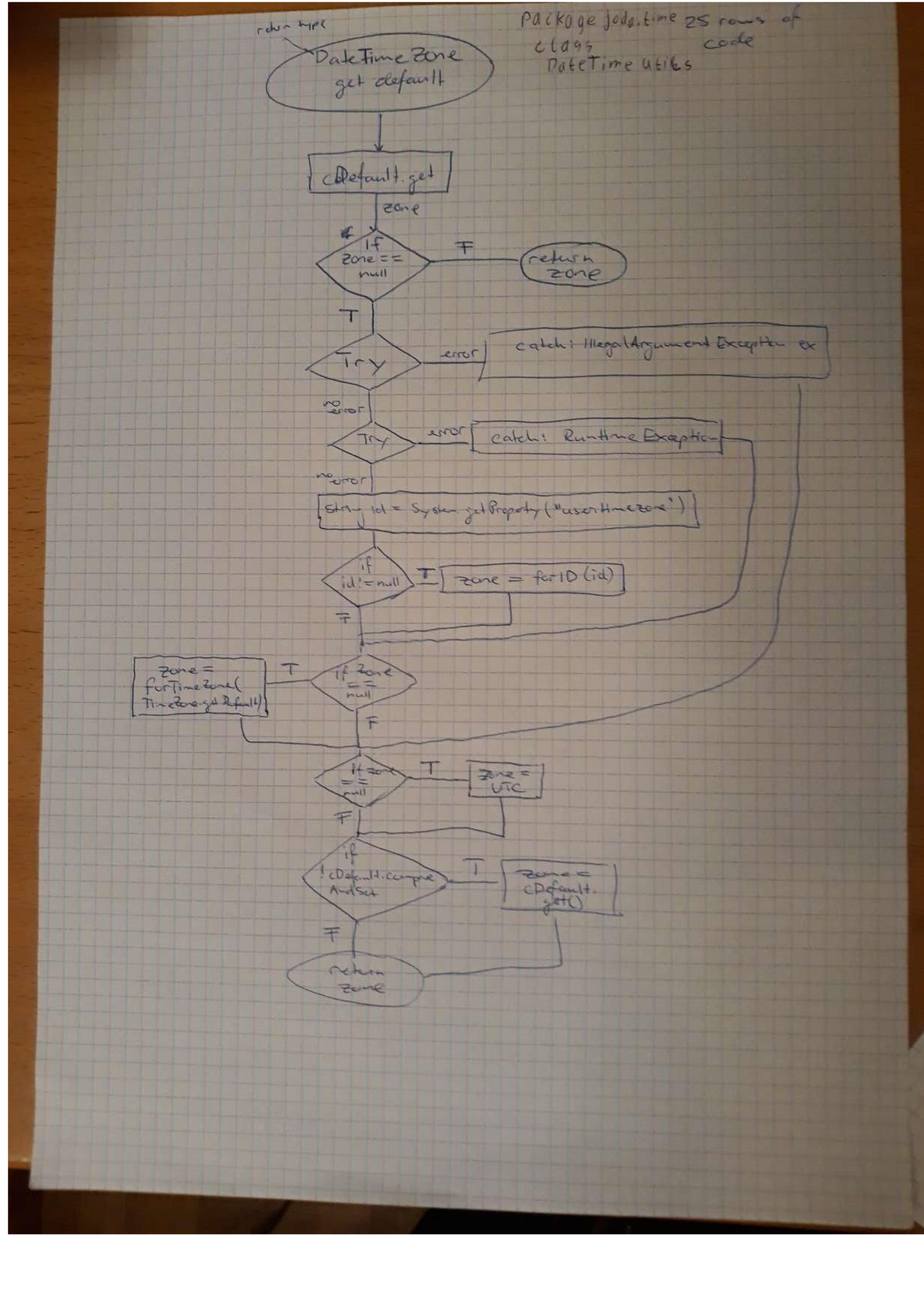


10 rows of code

joda.time.TimeOfDay.java

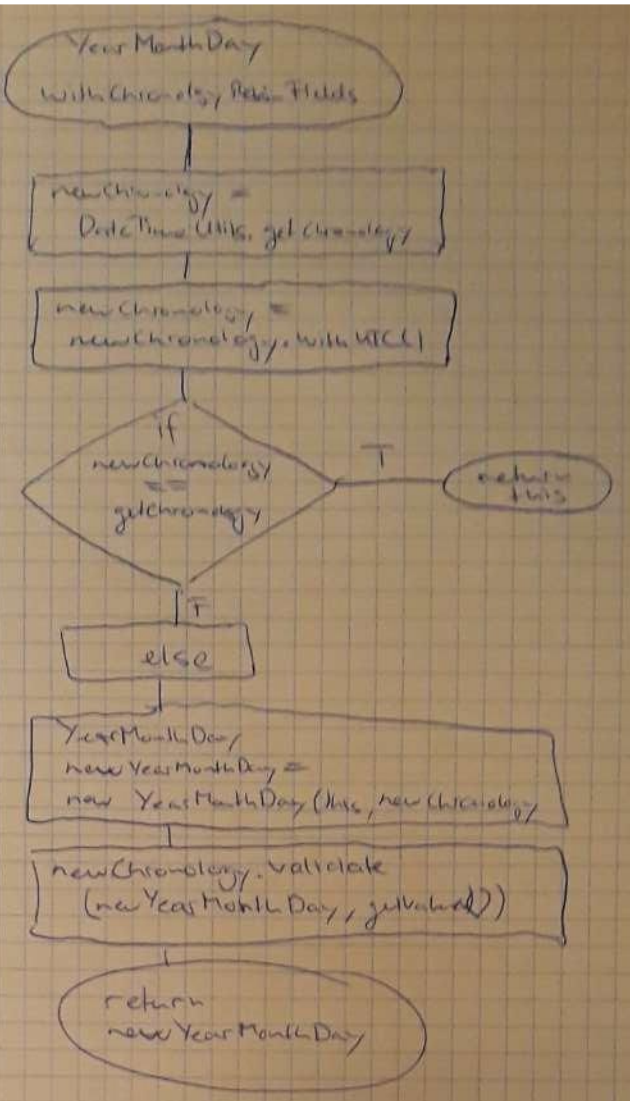
13 lines of code




return type
DateTimeZone getDefault
Package joda.time 25 rows of code
class DateTimeUtils
cDefault.get
zone
if zone == null
F
return zone
T
Try
error
catch: IllegalArgumentException ex
no error
Try
error
catch: RuntimeException
no error
String id = System.getProperty("user.timezone")
if id != null
T
zone = forID(id)
F
zone = forTimeZone(TimeZone.getDefault())
T
if zone == null
F
if zone == null
T
zone = UTC
F
if !cDefault.compareAndSet
T
zone = cDefault.get()
F
return zone

Package: ~~java.time~~ java.time

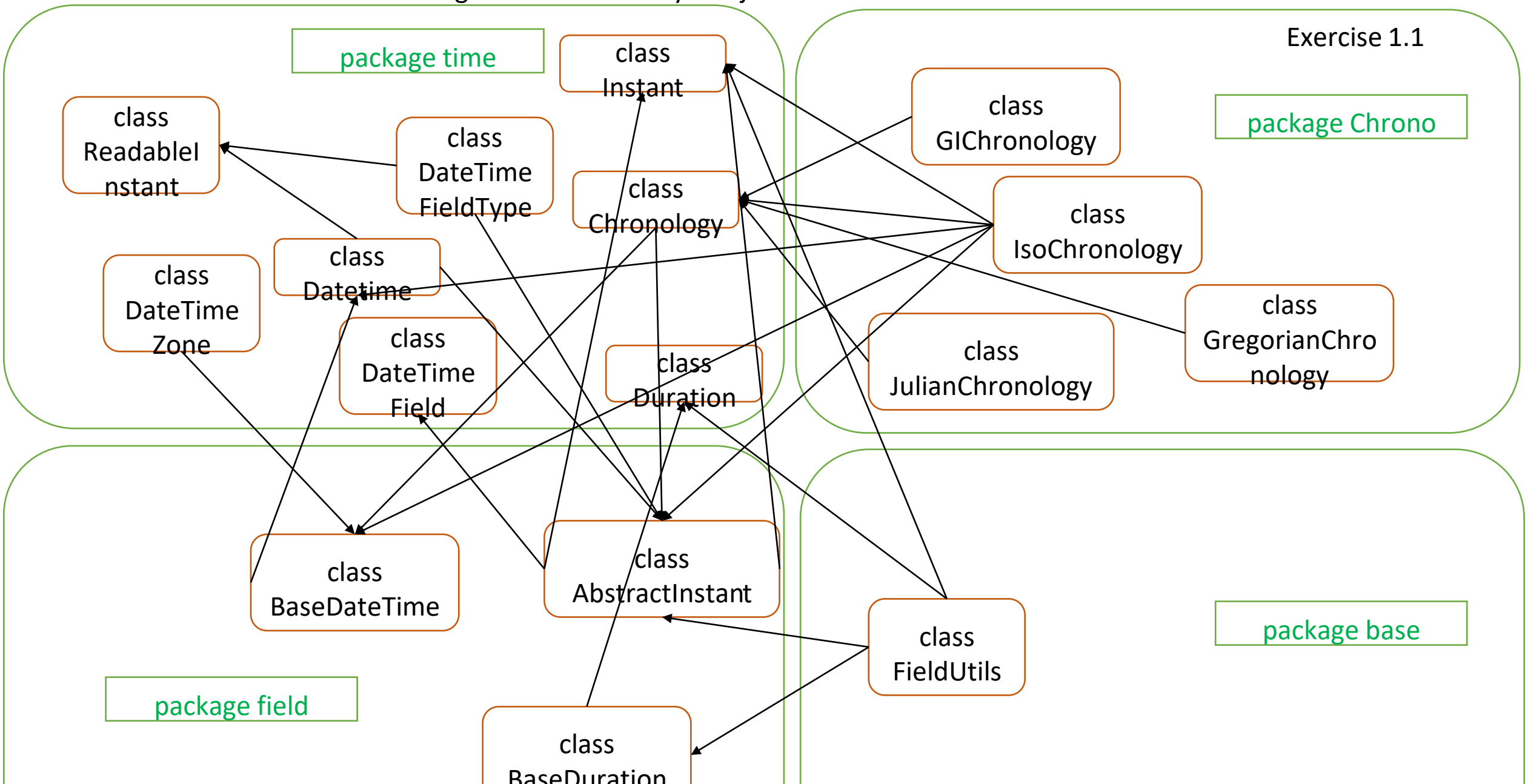
Class: YearMonthDay



8 lines of code

Assignment Solution by Nadja Näf Vincent Müller und Yves Boutellier

Exercise 1.1



Assignment Solution by Nadja Näf Vincent Müller und Yves Boutellier
Exercise 1.1

For the map of the architecture, we decided to look at the imports that each class makes. We started at the package joda time, because from the documentation we have inferred, that here are the most classes defined, the user interacts with and it is in general the most important package.

We also looked at the package base, because there are many of the important backend classes of the library definded. Many connections inside the time package are made through classes in the package base. This is another reason we decided to include this package.

And we looked at the package chrono. There the calendar system is implemented, with the IsoChronology as default calendar but also other calendars and the possibility to plug in own calendar system. We found these features very important since it determines in what sizes we can talk about time, week, day or hour for example. So we decided to include this package as well.

The last package we decided to include in our analysis is the package field. This package is used for displaying the date and time information in different formats. For example: The DayOfYear Method shows the days passed at that year.

Many of the important User interaction is done via this package. Therefore, we decided to also include this package.

From the documentation, we also inferred which classes are the most important ones in the package joda.time. Otherwise, because there are so many classes in this package and some of them are significantly less important then others. Then we looked, which classes are importet from joda.time made to the other selected packages: base, chrono, field. And we also regarded the connections from these other connections back to joda.time. This way we

Assignment Solution by Nadja Näf Vincent Müller und Yves Boutellier

could figure out the connections between important members within joda.time going via these other packages, most importantly base, for this purpose.

Exercise 1.2