

Java Database Connectivity Driver for MobilityDB

Tatiana del Pilar Millan Poveda

Supervisor: Prof. Esteban Zimányi

September 1st, 2022



Introduction

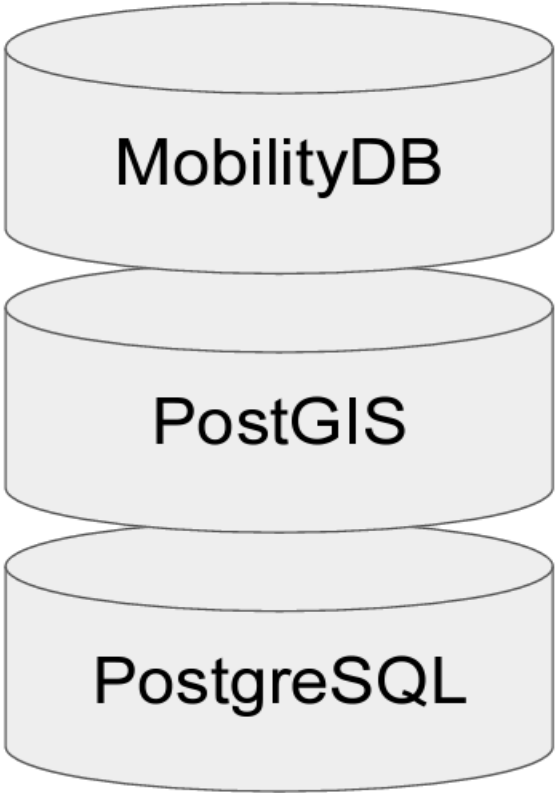
- MobilityDB Management system for moving objects.
- Stores and retrieve information efficiently.
- Python adapter for MobilityDB.
- Java popular programming language.

OBJECTIVE: Create a Java Database Connectivity Driver for MobilityDB with feature parity with the Python Adapter.

MobilityDB Architecture

MobilityDB: Architecture





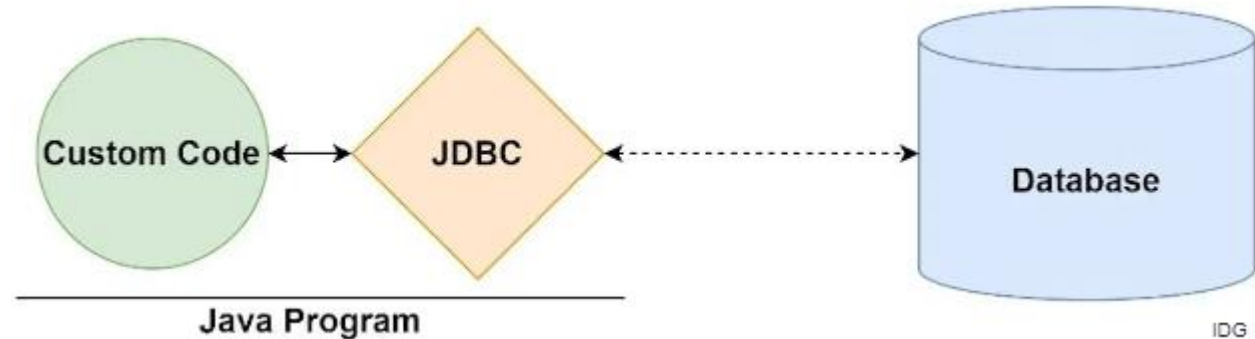
tgeompoint, tgeogpoint,
tint, tfloat, ttext, tbool

geometry, geography

numeric, monetary, character,
data/time, boolean, enum,
arrays, range,
XML, JSON, ...

Java Data Base Connectivity

- Acts as an intermediary between the Java code and the database
- Java API
- It is used for:
 - Connecting to a database
 - Querying and updating database
 - Handling the result sets



PostgreSQL JDBC

- Connects Java code with PostgreSQL
- “*addDataType*” method for creation of custom data type

```
((org.postgresql.PGConnection)myconn).addDataType("mytype","my.class.name");
```

- New classes should inherit from *PGObject*



PostGIS JDBC

- For accessing geometry objects in PostgreSQL database
- Register the type to the connection:

```
((org.postgresql.PGConnection)conn)  
    .addDataType("geometry", Class.forName("org.postgis.PGgeometry"));
```

- Retrieve the Geometry

```
PGgeometry geom = (PGgeometry)r.getObject(1);
```

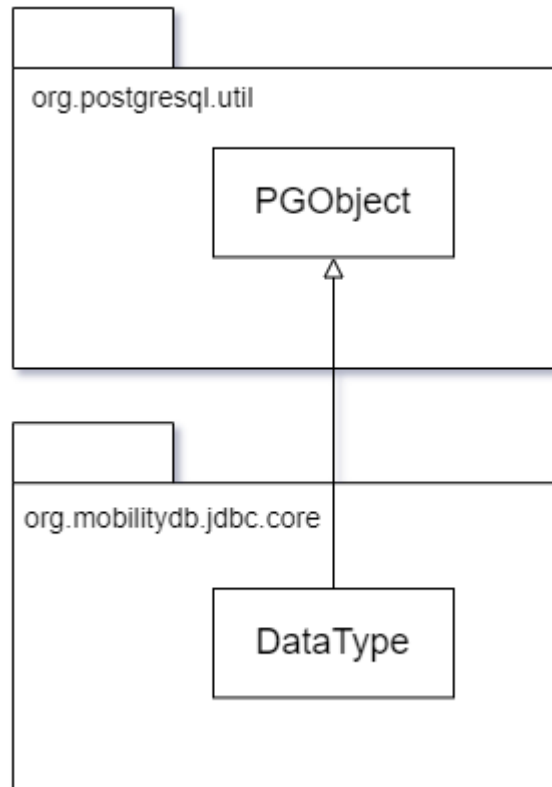




Solution

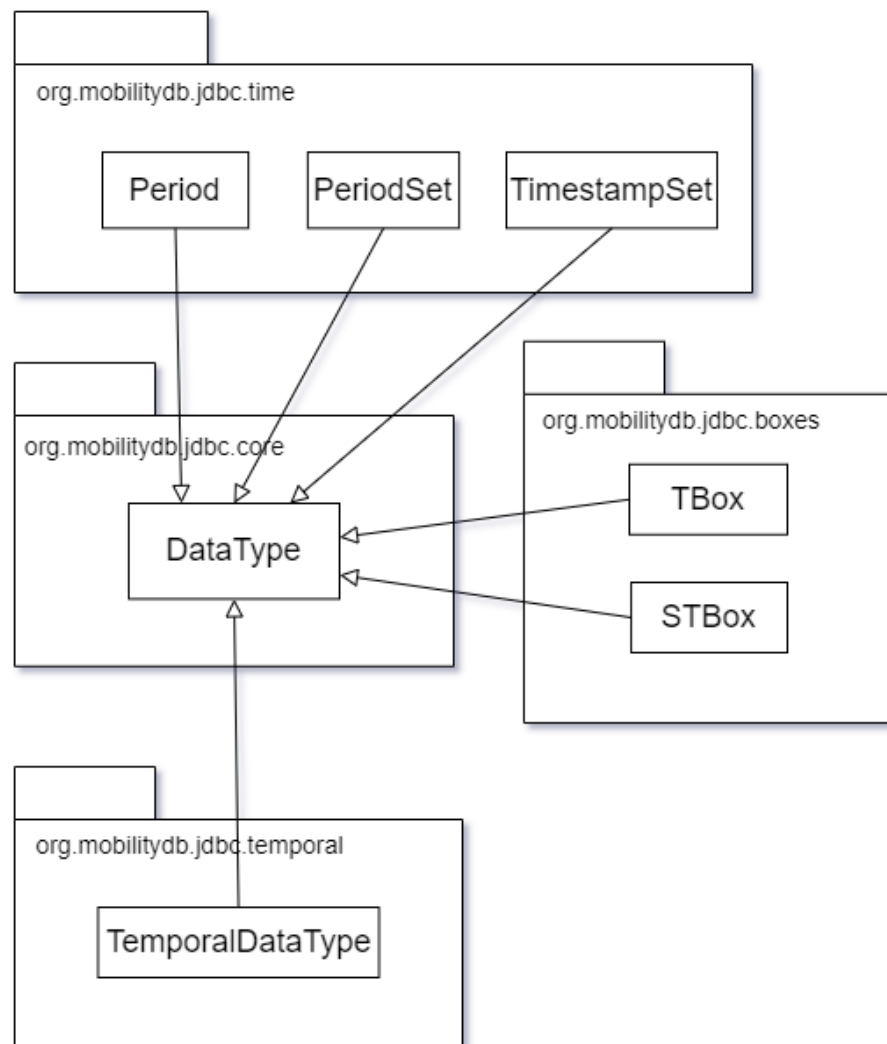


Base Class *DataType*



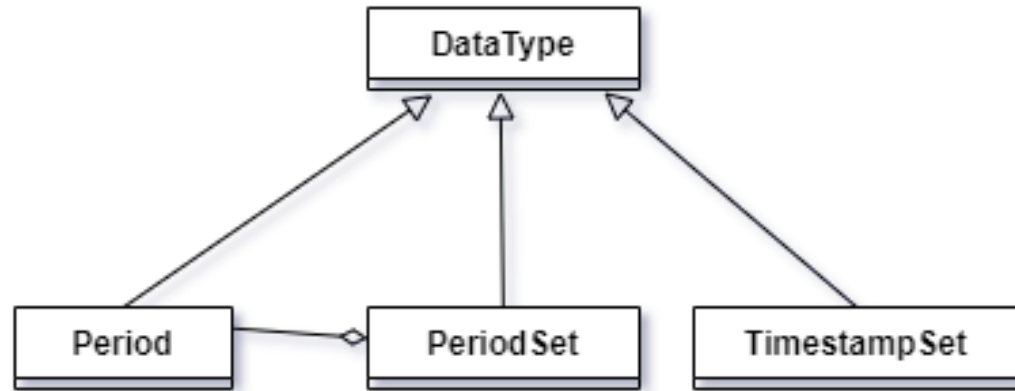
- To add MobilityDB data types to PostgreSQL JDBC, they must extend `PGObject`.
- `DataType` class inherits from `PGObject` and is the base class for all types.

Type Inheritance



- MobilityDB types inherit from `DataType` class
- They need to override the methods *getValue* and *setValue*

Time Types

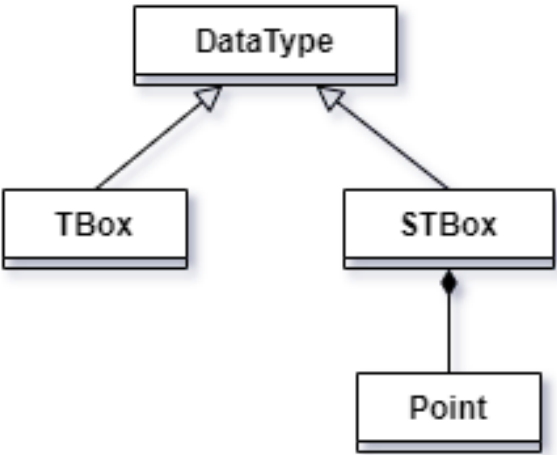


- Time types are:
 - ***TimestampSet***
 - ***Period***
 - ***PeriodSet***
- Example of usage for a *PeriodSet*

```

//Constructor PeriodSet(String value)
PeriodSet periodSet = new PeriodSet (
    "[2019-09-08 00:00:00+01, 2019-09-10 00:00:00+01],
    [2019-09-11 00:00:00+01, 2019-09-12 00:00:00+01)]");
Period period = periodSet.period();
// The value for period is '[2019-09-08 00:00:00+01,
    2019-09-12 00:00:00+01)']
    
```

Box Types



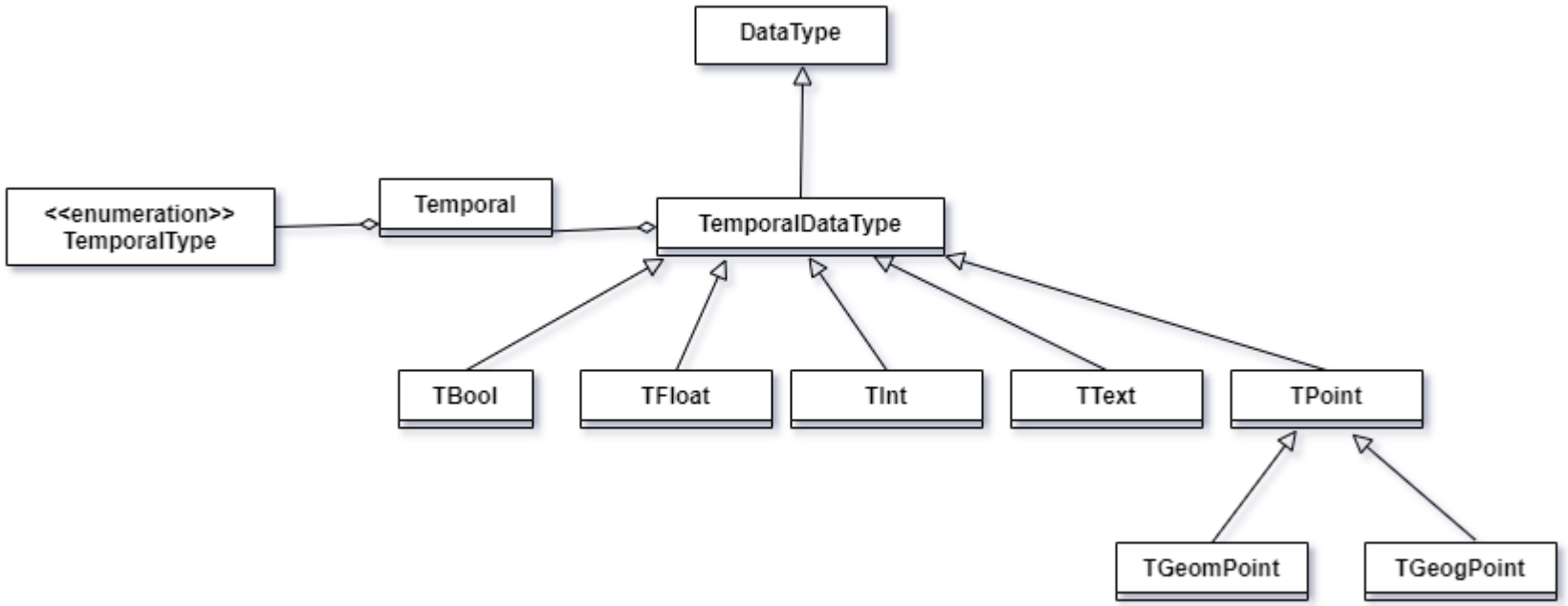
- Box types are:
 - *TBox*
 - *STBox*
- Example of usage for a *STBox*

```

//Constructor STBox ( Point pMin , OffsetDateTime
tMin, Point pMax ,
OffsetDateTime tMax , boolean isGeodetic , int srid)
OffsetDateTime tMin = OffsetDateTime.now();
OffsetDateTime tMax = OffsetDateTime.now()
    .plusDays(1);
STBox stBox = new STBox (
    new Point (1.0 ,2.0 ,3.0), tMin,
    new Point (1.0 ,2.0 ,3.0), tMax , true, 5676);
    
```

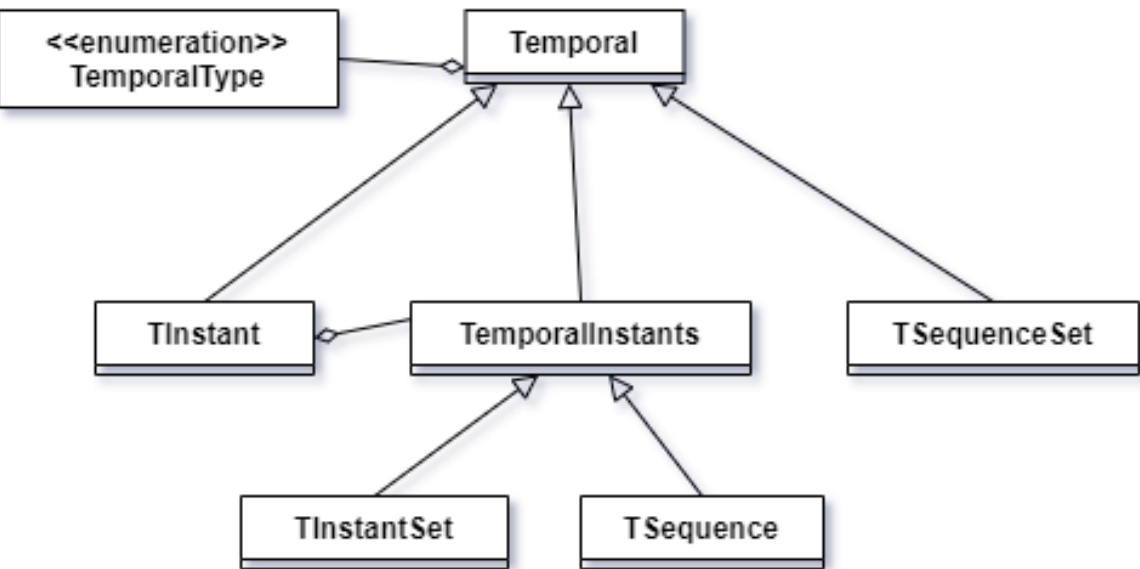


Temporal Types Abstraction



- Temporal Types need a different abstraction, because each type has subtypes.
- Every temporal type will inherit from TemporalType.

Temporal Subtypes

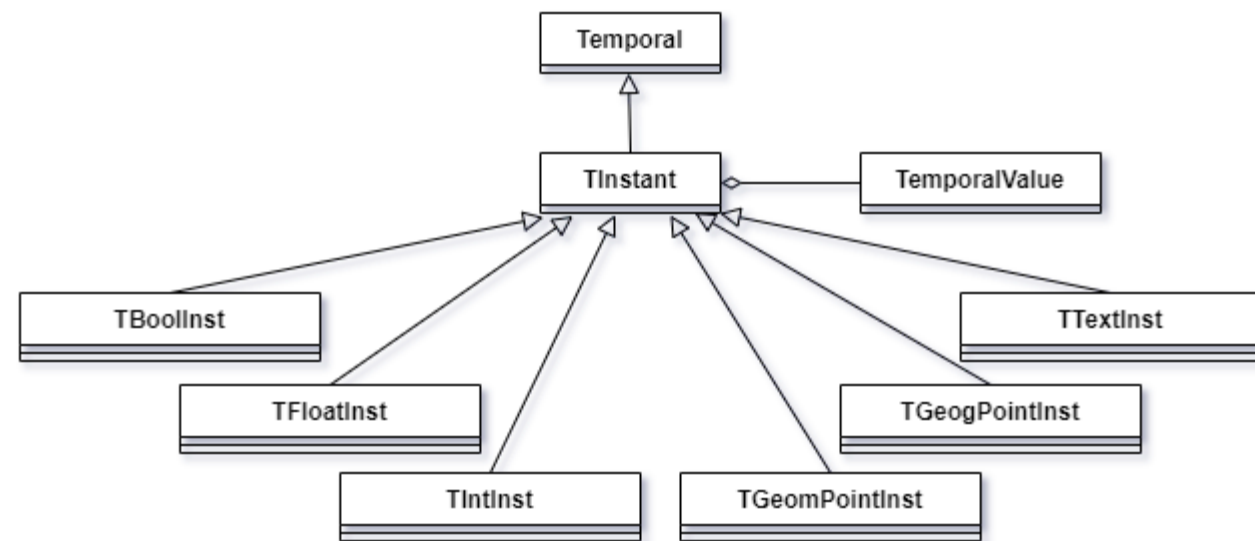


The subtypes are:

- **TInstant**
- **TInstantSet**
- **Tsequence**
- **TsequenceSet**

Temporal Instant

- Represent a single instant.
- Could be of type: *TBoolInst*, *TFloatInst*, *TIntInst*, *TGeomPointInst*, *TGeogPointInst* or *TTextInst*.
- Example of usage for a *TFloatInst*

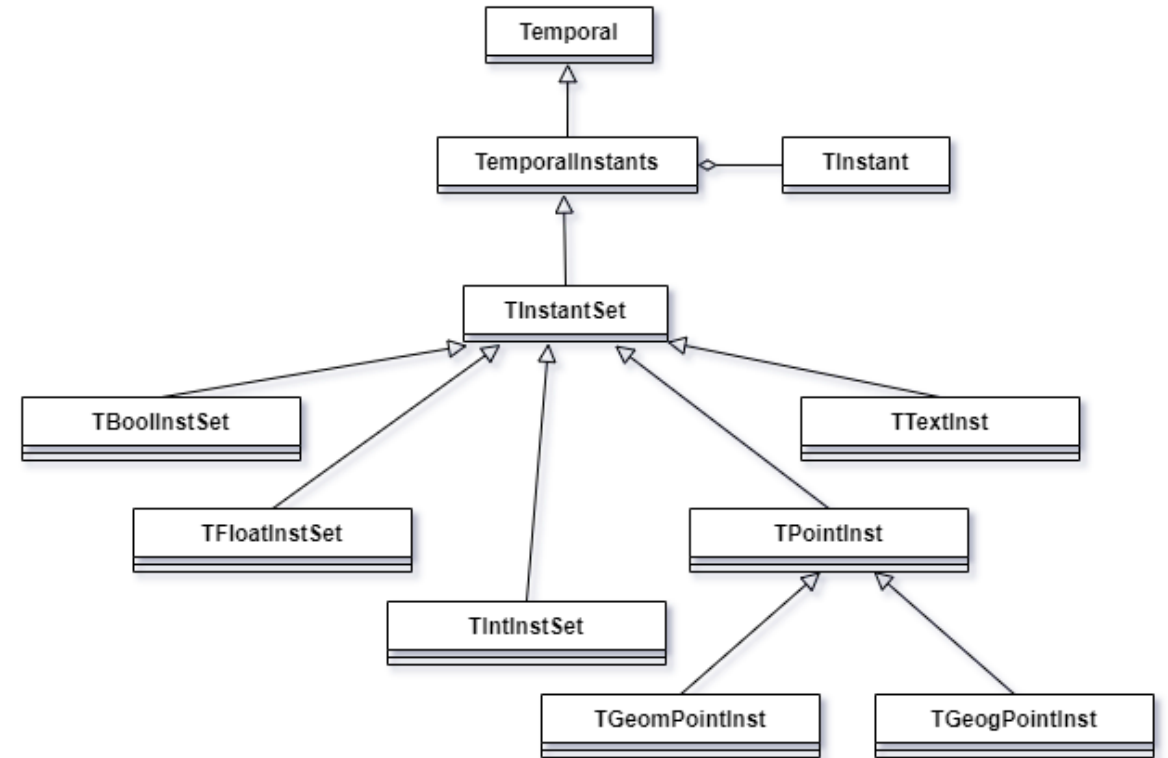


```

TFloat tFloat = new TFloat("10.5@2019-09-08 06:04:32+02"); //creates a Tfloat
tFloat.getTemporalType(); // returns TEMPORAL_INSTANT
tFloat.getTemporal(); // returns the temporal of type TFloatInst
tFloat.getTemporal().shift(Duration.ofDays(2)); // the value will be '10.5@2019-09-10 06:04:32+02'
    
```

Temporal Instant Set

- Represents a set of instants
- Could be of type: *TBoolInstSet*, *TFloatInstSet*, *TIntInstSet*, *TGeomPointInstSet*, *TGeogPointInstSet* or *TTextInstSet*.
- Example of usage for a *TGeomPointInstSet*

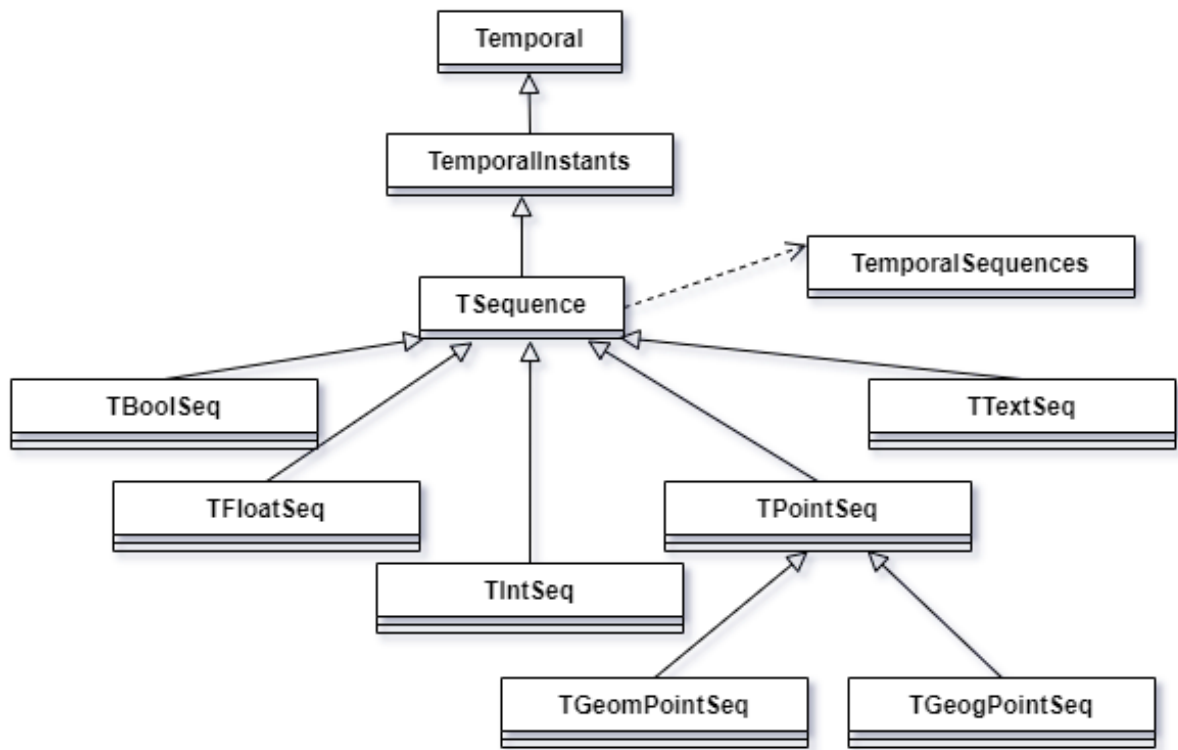


```

TGeomPoint tGeomPoint = new TGeomPoint("{Point(0 0)@2001-01-01 08:00:00+02, " +
    "Point(1 1)@2001-01-03 08:00:00+02}"); //creates a TGeomPoint
tGeomPoint.getTemporalType(); //returns TEMPORAL_INSTANT_SET
tGeomPoint.getTemporal(); //returns a temporal of type TGeomPointInstSet
tGeomPoint.getTemporal().endTimestamp(); //the value will be '2001-01-03 08:00:00+02'
  
```

Temporal Sequence

- Represents trajectories, continues movement.
- Could be of type: *TBoolSeq*, *TFloatSeq*, *TIntSeq*, *TGeomPointSeq*, *TGeogPointSeq* or *TTextSeq*
- Example of usage for a *TIntSeq*

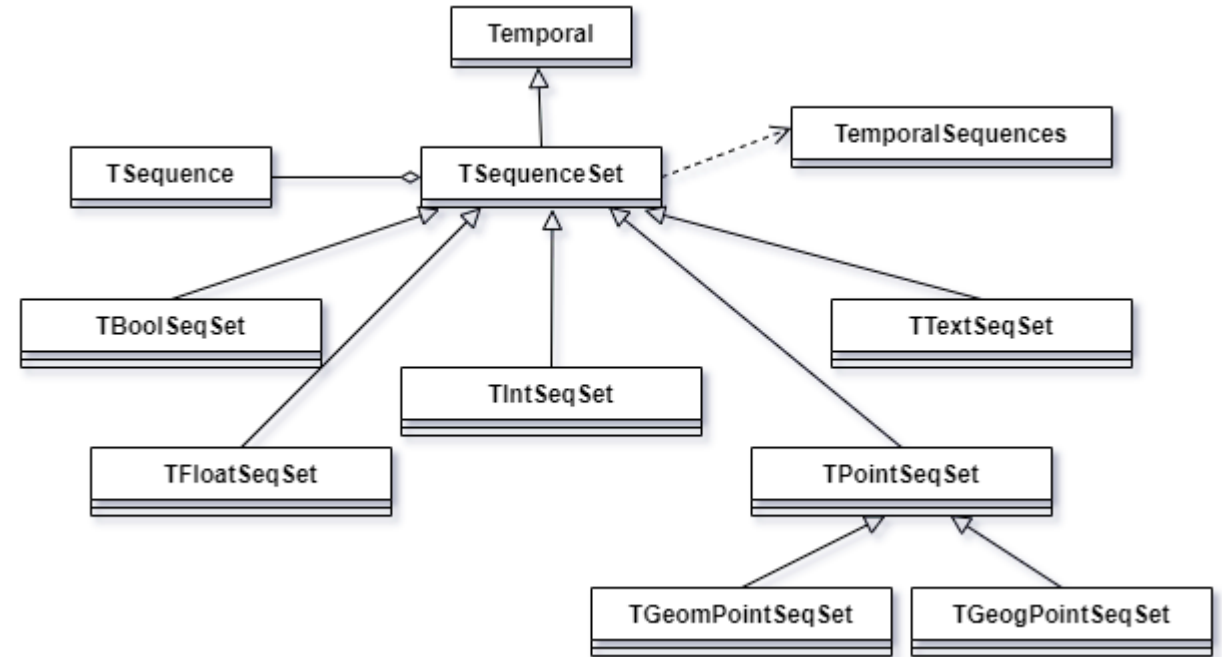


```

TInt tInt = new TInt ("(1@2001-01-01 08:00:00+02, 2@2001-01-03 08:00:00+02]"); //Creates a TInt
tInt.getTemporalType(); //returns TEMPORAL_SEQUENCE
tInt.getTemporal(); //returns a temporal of type TIntSeq
tInt.getTemporal().instantN(1); //the value will be '2@2001-01-03 08:00:00+02'
  
```


Temporal Sequence Set

- Introduce temporal gaps in the movement
- Could be of type: *TBoolSeqSet*, *TFloatSeqSet*, *TIntSeqSet*, *TGeomPointSeqSet*, *TGeogPointSeqSet* or *TTextSeqSet*
- Example of usage for a *TTextSeqSet*



```

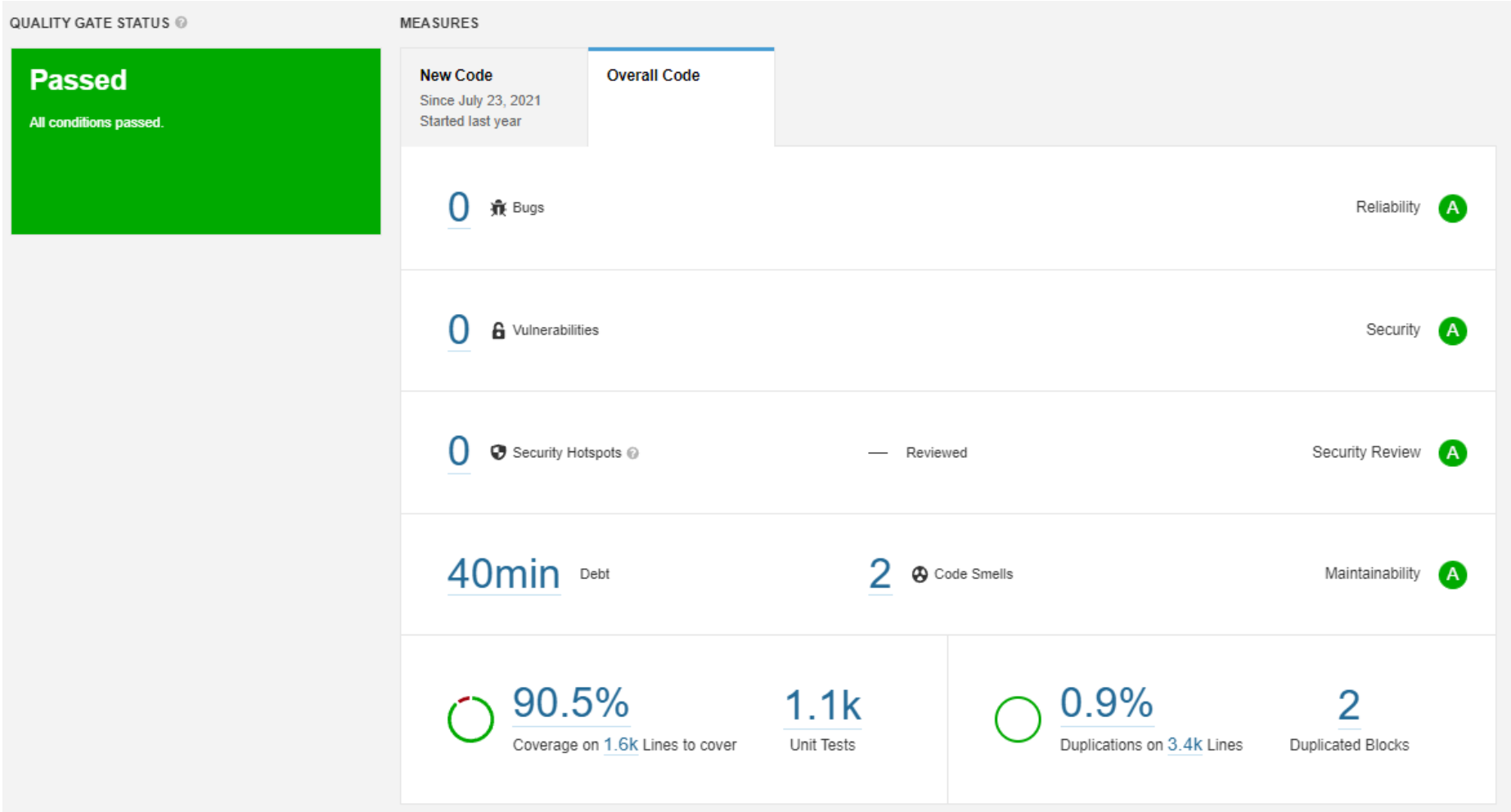
TText tText = new TText ("{(abcd@2001-01-01 08:00:00+02, efgh@2001 -01 -03 08:00:00+02]," +
    "[ijkl@2001-01-04 08:00:00+02, mnop@2001-01-05 08:00:00+02, qrst@2001-01-06 08:00:00+02]}");
//creates a TText
tText.getTemporalType(); //returns TEMPORAL_SEQUENCE_SET
tText.getTemporal(); //returns a temporal of type TTextSeqSet
tText.getTemporal().numSequences(); //the value will be '2'
  
```

Code Quality - SonarQube

- SonarQube is a static code analyzer
- Assures code quality and maintainability.
- Needs 80% code coverage



SonarQube Current Status



SonarQube Common Code Smells

mobilitydb-jdbc

☆ master

+

Last analysis had 1 warning

August 28, 2022 at 6:05 PM

Version unspecified

🏠

Overview

Issues

Security Hotspots

Measures

Code

Activity

Project Settings

Project Information

My Issues

All

Filters

Clear All Filters

Type

CODE SMELL

Clear

Bug

0

Vulnerability

0

Code Smell

6

Ctrl + click to add to selection

Severity

Blocker

1

Minor

2

Critical

1

Info

0

Major

2

Scope

Resolution

Status

Security Category

Creation Date

NEW CODE

Clear

Language

Bulk Change

1 / 6 issues

1h 23min effort

src/.../java/com/mobilitydb/jdbc/tbool/TBool.java

Remove this unused import 'java.util.List'. Why is this an issue?

10 minutes ago

L11

🔗

⌵

Code Smell

Minor

Open

Not assigned

2min effort

Comment

unused

Override the "equals" method in this class. Why is this an issue?

10 minutes ago

L18

🔗

⌵

Code Smell

Minor

Open

Not assigned

30min effort

Comment

suspicious

"value" is the name of a field in "PGObject". Why is this an issue?

10 minutes ago

L19

🔗

⌵

Code Smell

Blocker

Open

Not assigned

5min effort

Comment

confusing

Refactor this method to reduce its Cognitive Complexity from 16 to the 15 allowed. Why is this an issue?

29 seconds ago

L33

🔗

⌵

Code Smell

Critical

Open

Not assigned

6min effort

Comment

brain-overload

src/.../com/mobilitydb/jdbc/tpoint/TPointInstSet.java

1 duplicated blocks of code must be removed. Why is this an issue?

3 months ago

🔗

⌵

Code Smell

Major

Open

Not assigned

20min effort

Comment

pitfall

src/.../com/mobilitydb/jdbc/tpoint/TPointSeq.java

1 duplicated blocks of code must be removed. Why is this an issue?

3 months ago

🔗

⌵

Code Smell

Major

Open

Not assigned

20min effort

Comment

pitfall

6 of 6 shown

Unit Tests

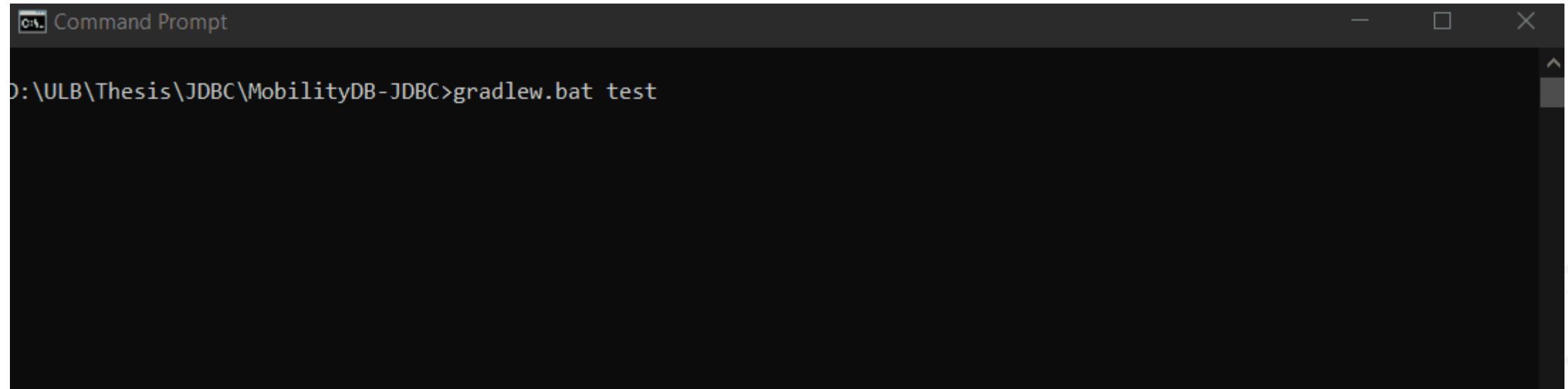
- In charge of checking that each method works as expected.
- Allows to make refactors in the code without affecting existing functionality

```
@Test
void testConstructor() throws SQLException {
    String value = "true@2019-09-08 06:04:32+02";
    ZoneOffset tz = ZoneOffset.of("+02:00");
    OffsetDateTime otherDate = OffsetDateTime
        .of(2019,9, 8, 6, 4, 32, 0, tz);

    TBoolInst tBoolInst = new TBoolInst(value);
    TBoolInst other = new TBoolInst(true, otherDate);

    assertEquals(tBoolInst.getTemporalValue(),
        other.getTemporalValue());
}
```

Running Unit Tests – Command Line



```
Command Prompt
D:\ULB\Thesis\JDBC\MobilityDB-JDBC>gradlew.bat test
```

Unit Tests Report

Test Summary

1127

tests

0

failures

0

ignored

1.941s

duration

100%

successful

Packages

Classes

Package	Tests	Failures	Ignored	Duration	Success rate
com.mobilitydb.jdbc.unit	1	0	0	0.865s	100%
com.mobilitydb.jdbc.unit.boxes	86	0	0	0.168s	100%
com.mobilitydb.jdbc.unit.tbool	156	0	0	0.147s	100%
com.mobilitydb.jdbc.unit.tfloat	152	0	0	0.334s	100%
com.mobilitydb.jdbc.unit.time	115	0	0	0.087s	100%
com.mobilitydb.jdbc.unit.tint	150	0	0	0.101s	100%
com.mobilitydb.jdbc.unit.tpoint	4	0	0	0.004s	100%
com.mobilitydb.jdbc.unit.tpoint.helpers	10	0	0	0.016s	100%
com.mobilitydb.jdbc.unit.tpoint.tgeog	151	0	0	0.091s	100%
com.mobilitydb.jdbc.unit.tpoint.tgeom	151	0	0	0.074s	100%
com.mobilitydb.jdbc.unit.ttext	151	0	0	0.054s	100%

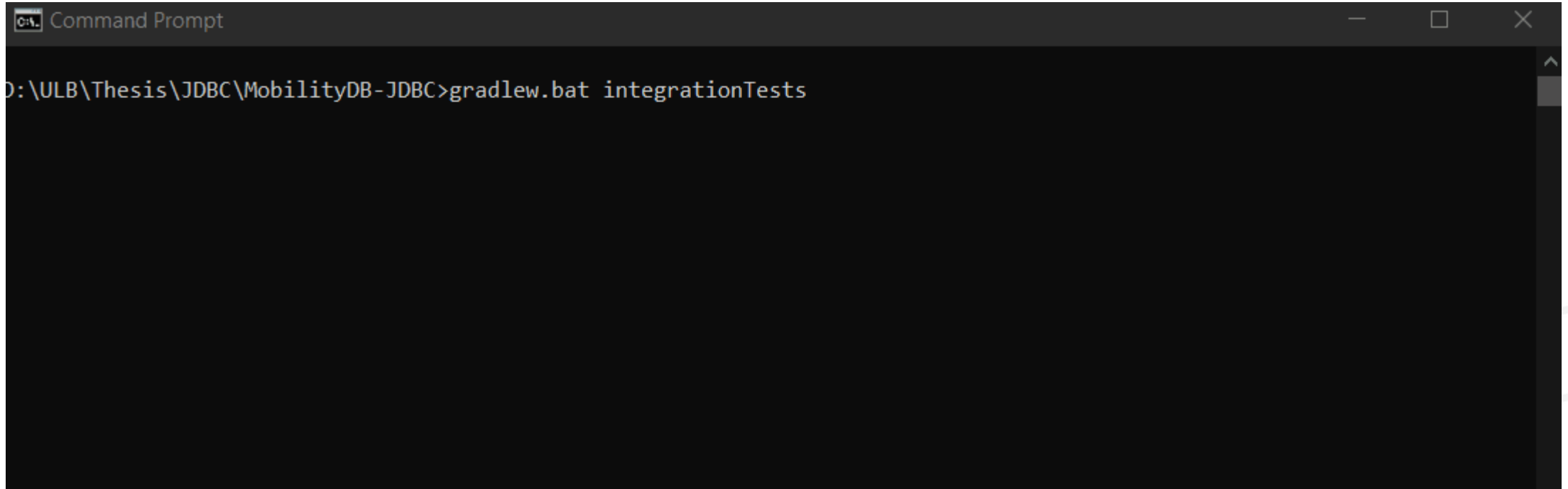
1127 unit tests running successfully.

Integration Tests

Check the JDBC driver is working as expected while sending and retrieving data from the database.

```
@Test
void testTIntStringConstructor() throws SQLException {
    TInt tInt = new TInt("10@2021-04-08 05:04:45+01");
    PreparedStatement insertStatement = con.prepareStatement(
        "INSERT INTO tbl_tint (temporalType) VALUES (?);");
    insertStatement.setObject(1, tInt);
    ...
    PreparedStatement readStatement = con.prepareStatement(
        "SELECT temporalType FROM tbl_tint WHERE temporalType=?;");
    readStatement.setObject(1, tInt);
    ...
    if (rs.next()) {
        TInt retrievedTInt = (TInt) rs.getObject(1);
        assertEquals(tInt.getTemporal(), retrievedTInt.getTemporal());
    }
    ...
}
```


Running Integration Tests – Command Line



```
Command Prompt
D:\ULB\Thesis\JDBC\MobilityDB-JDBC>gradlew.bat integrationTests
```

Integration Tests Report

Test Summary

145
tests

0
failures

0
ignored

7.528s
duration

100%
successful

Packages

Classes

Package	Tests	Failures	Ignored	Duration	Success rate
com.mobilitydb.jdbc.integration.bboxes	13	0	0	0.710s	100%
com.mobilitydb.jdbc.integration.tbool	13	0	0	0.717s	100%
com.mobilitydb.jdbc.integration.tfloat	15	0	0	0.846s	100%
com.mobilitydb.jdbc.integration.time	8	0	0	0.441s	100%
com.mobilitydb.jdbc.integration.tint	13	0	0	0.664s	100%
com.mobilitydb.jdbc.integration.tpoint.tgeog	35	0	0	1.828s	100%
com.mobilitydb.jdbc.integration.tpoint.tgeom	35	0	0	1.648s	100%
com.mobilitydb.jdbc.integration.ttext	13	0	0	0.674s	100%

145
integration
tests running
successfully

Javadoc

```
/**
 * Helper to format dateTime values
 */
public final class DateTimeFormatHelper {
...
    /**
     * Formats a string into an OffsetDateTime
     * @param value String
     * @return OffsetDateTime
     */
    public static OffsetDateTime getDateTimeFormat(String value) {
        DateTimeFormatter format = DateTimeFormatter
            .ofPattern(FORMAT);
        return OffsetDateTime.parse(value.trim(), format);
    }
...
}
```

- Documentation generator
- Uses comments in code to generate HTML documentation
- Standard for Java

Javadoc generated HTML page

OVERVIEW

PACKAGE

CLASS

TREE

DEPRECATED

INDEX

HELP

ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

Package

com.mobilitydb.jdbc.core

Class

DateTimeFormatHelper

java.lang.Object

com.mobilitydb.jdbc.core.DateTimeFormatHelper

public final class

DateTimeFormatHelper

extends java.lang.Object

Helper to format date

Time values

Method Summary

All Methods

Static Methods

Concrete Methods

Modifier and Type	Method	Description
static java.time.OffsetDateTime	getDateTimeFormat(java.lang.String value)	Formats a string into an OffsetDateTime
static java.lang.String	getStringFormat(java.time.OffsetDateTime value)	Formats an OffsetDateTime into a string

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Demo with MobilityDB Workshop

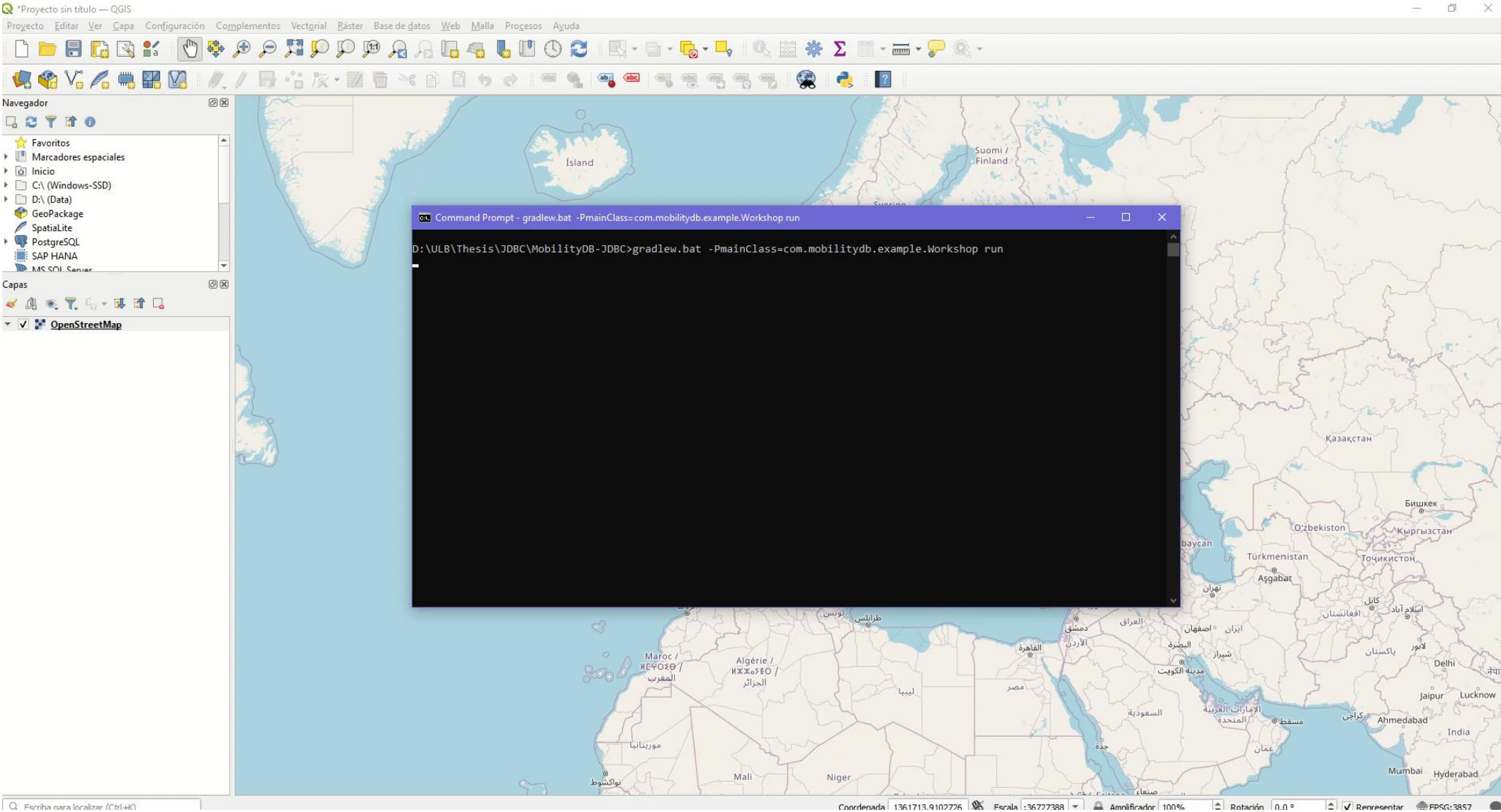


Why the workshop?



- Illustrates usage scenarios of MobilityDB.
<https://github.com/MobilityDB/MobilityDB-workshop>
- Module used: “Managing Ship Trajectories” until *Constructing Trajectories* .
- *Docker image with dependencies and files*

Execution and QGIS



Conclusions

- ***Abstractions*** were helpful for extensibility and maintainability.
- ***Unit*** and ***integration tests*** useful for identifying issues.
- ***SonarQube*** assure code maintainability and quality.
- ***Example*** gives a realistic view of the usage of the driver.
- ***Differences*** in ***implementation***, although feature parity.

Future Work

- Add SonarQube to Github Actions
- Publish the JDBC Driver for MobilityDB to Maven Central Repository
- Maintenance and implementation of new features
- Integrate with ORM like Hibernate by extending PostGIS dialect

References

- Université libre de Bruxelles and MobilityDB contributors. *MobilityDB*. <https://github.com/MobilityDB/MobilityDB>. 2022.
- Université libre de Bruxelles and MobilityDB contributors. *MobilityDB Workshop*. <https://github.com/MobilityDB/MobilityDB-workshop>. 2021.
- Université libre de Bruxelles and MobilityDB contributors. *MobilityDB-python*. <https://github.com/MobilityDB/MobilityDB-python>. 2021.
- Esteban Zimanyi. *Welcome to python-mobilitydb's documentation!* 2020. url: <https://docs.mobilitydb.com/python-mobilitydb/master/>.
- Esteban Zimanyi and Mahmoud Sakr. *MobilityDB: Managing Mobility Data in PostgreSQL*. PGConf.Russia. 2020. url: <https://pgconf.ru/en/2020/264545>.
- PostgreSQL Tutorial Website. *What is PostgreSQL?* url: <https://www.postgresqltutorial.com/what-is-postgresql/>.
- Johannes Link Stefan Bechtold Sam Brannen and collaborators. *JUnit 5 User Guide*. url: <https://junit.org/junit5/docs/current/user-guide/#overview>.

References

- *Chapter 5. Using PostGIS: Building Applications*. PostGIS. url: <https://postgis.net/docs/manual-1.5/ch05.html> (visited on 08/08/2022).
- PostGIS Project Steering Committee. *PostGIS - Spatial and Geographic objects for PostgreSQL*. url: <https://postgis.net/>.
- SonarSource S.A. *SonarQube Documentation*. url: <https://docs.sonarqube.org/latest/>.
- *PostgreSQL JDBC Driver*. The PostgreSQL Global Development Group. [url:https://jdbc.postgresql.org/about/about.html](https://jdbc.postgresql.org/about/about.html) (visited on 08/06/2022).
- *PostgreSQL Extensions to the JDBC API*. The PostgreSQL Global Development Group. url: <https://www.postgresql.org/docs/7.4/jdbc-ext.html> (visited on 08/08/2022).
- Official JDBC driver for PostgreSQL. *PostgreSQL JDBC Driver*. <https://github.com/pgjdbc/pgjdbc>. 2022.
- *Javadoc*. Oracle. url: <https://docs.oracle.com/javase/9/javadoc/javadoc.htm> (visited on 08/08/2022).
- The PostgreSQL Global Development Group. *The PostgreSQL JDBC Interface*. [url:https://jdbc.postgresql.org/documentation/head/index.html](https://jdbc.postgresql.org/documentation/head/index.html).