ЗАТВЕРДЖЕНО

1116130.00755-01 12 01-ЛЗ

ПРОГРАМНИЙ ЗАСІБ

Модуль інтеграції даних систем аналізу показників бортових систем

діагностування локомотивів

Текст програми

1116130.00755-01 12 01

Аркушів 45

2014

Анотація

Документ 1116130.00755-01 12 01 «Модуль інтеграції даних систем аналізу показників бортових систем діагностування локомотивів. Текст програми» входить до складу програмної документації.

У даному документі представлені інформація про класи та текст програми. Програма написана на мові Java. Об'єм пам'яті, що займає програма комплексу складає 40 Мб. Конфігурація комп'ютера стандартна.

ЗМІСТ

[1 ІНФОРМАЦІЯ ПРО КЛАСИ 4](#_Toc390819542)

[2 ТЕКСТ ПРОГРАМИ 6](#_Toc390819543)

[2.1 Текст програми DashboardController.java 6](#_Toc390819544)

[2.2 Текст програми DataManagmentController.java 7](#_Toc390819545)

[2.3 Текст програми DownloadDataController.java 8](#_Toc390819546)

[2.4 Текст програми IndexController.java 9](#_Toc390819547)

[2.5 Текст програми LocomotiveController.java 11](#_Toc390819548)

[2.6 Текст програми DataDaoImpl.java 11](#_Toc390819549)

[2.7 Текст програми FileInfoDaoImpl.java 17](#_Toc390819550)

[2.8 Текст програми FileStructureInfoDaoImpl.java 18](#_Toc390819551)

[2.9 Текст програми LocomotiveDaoImpl.java 19](#_Toc390819552)

[2.10 Текст програми RecordStructureInfoDaoImpl.java 20](#_Toc390819553)

[2.11 Текст програми SensorDaoImpl.java 20](#_Toc390819554)

[2.12 Текст програми UserDaoImpl.java 22](#_Toc390819555)

[2.13 Текст програми FileInfo.java 23](#_Toc390819556)

[2.14 Текст програми FileStructureInfo.java 23](#_Toc390819557)

[2.15 Текст програми LocoDataEntity.java 24](#_Toc390819558)

[2.16 Текст програми LocoEntity.java 35](#_Toc390819559)

[2.17 Текст програми RecordStructureInfo.java 35](#_Toc390819560)

[2.18 Текст програми SensorEntity.java 36](#_Toc390819561)

[2.19 Текст програми UserEntity.java 37](#_Toc390819562)

[2.20 Текст програми CustomException.java 38](#_Toc390819563)

[2.21 Текст програми DataAccessException.java 38](#_Toc390819564)

[2.22 Текст програми ValidationException.java 38](#_Toc390819565)

[2.23 Текст програми DataLoadingServiceImpl.java 38](#_Toc390819566)

[2.24 Текст програми FileCheckInServiceImpl.java 40](#_Toc390819567)

[2.25 Текст програми LocomotiveServiceImpl.java 41](#_Toc390819568)

[2.26 Текст програми SensorServiceImpl.java 42](#_Toc390819569)

[2.27 Текст програми StatisticServiceImpl.java 43](#_Toc390819570)

# ІНФОРМАЦІЯ ПРО КЛАСИ

До прорами входять наступні класи:

* DashboardController – клас, який слугує контролером для головної сторінки;
* DataManagmentController – клас, який слугує контролером для керування даними;
* DownloadDataController – клас, який слугує контролером для сторінки завантаженя даних;
* IndexController – клас, який слугує контролером для головної сторінки;
* LocomotiveController – клас, який слугує контролером для сторінки керуваня локомотивами;
* DataDaoImpl – клас, який слугує для зберігання статистичних даних у базі даних;
* FileInfoDaoImpl – клас, який слугує для зберігання інформації про файли з яких проводилося завантаження статистичних даних;
* FileStructureInfoDaoImpl – клас, який слугує зберігання структури файлу для отримання статистичних даних;
* LocomotiveDaoImpl – клас, який слугує для зберіганнґ інформації про локомотиви в базі даних;
* RecordStructureInfoDaoImpl – клас, який слугує для зберігання структури запису в файлі із статистичними даними;
* SensorDaoImpl – клас, який слугує для зберігання інформації про датчики за якими ведеться вимірювання;
* UserDaoImpl – клас, який слугує для зберігання інформації про користувачів системи;
* FileInfo – клас, який являє сутність таблиці у базі даних;
* FileStructureInfo – клас, який являє сутність таблиці у базі даних;
* LocoDataEntity – клас, який являє сутність таблиці у базі даних;
* LocoEntity – клас, який являє сутність таблиці у базі даних;
* RecordStructureInfo – клас, який являє сутність таблиці у базі даних;
* SensorEntity – клас, який являє сутність таблиці у базі даних;
* UserEntity – клас, який являє сутність таблиці у базі даних;
* CustomException – клас, який слугує батьківським класом для системних виключень;
* DataAccessException – клас, який реалізує виключення доступу до даних;
* ValidationException – клас, який реалізує виключення перевірки даних;
* DataLoadingServiceImpl – клас, який реалізує сервісний прошарок для завантаження статистичних данних;
* FileCheckInServiceImpl – клас, який реалізує перевірку завантаження данних з поточного файлу;
* LocomotiveServiceImpl – клас, який реалізує сервісний прошарок для отримання інформації про локомотиви;
* SensorServiceImpl – клас, який реалізує сервісний прошарок для отримання інформації про датчики наявні в системі;
* StatisticServiceImpl – клас, який реалізує сервісний прошарок для отримання статистичної інформації.

# ТЕКСТ ПРОГРАМИ

## Текст програми DashboardController.java

package com.locostatmanager.busines.controllers;

import com.locostatmanager.busines.dao.entities.LocoEntity;

import com.locostatmanager.busines.dao.entities.SensorEntity;

import com.locostatmanager.busines.exceptions.DataAccessException;

import com.locostatmanager.busines.exceptions.ValidationException;

import com.locostatmanager.busines.service.LocomotiveService;

import com.locostatmanager.busines.service.SensorService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import java.util.List;

/\*\*

\*

\* @author dmitry

\*/

@Controller

@RequestMapping(value = "/dashboard.htm")

public class DashboardController {

@Autowired

private LocomotiveService locomotiveService;

@Autowired

private SensorService sensorService;

@RequestMapping(method = RequestMethod.GET)

public String getPage() {

return "dashboard";

}

@ModelAttribute("locomotives")

public List<LocoEntity> getLocomotives() throws ValidationException, DataAccessException {

return locomotiveService.getAll();

}

@ModelAttribute("osnov")

public List<SensorEntity> getOsnov() throws DataAccessException, ValidationException {

return sensorService.getOsnovn();

}

@ModelAttribute("buks")

public List<SensorEntity> getBuks() throws DataAccessException, ValidationException {

return sensorService.getBuks();

}

@ModelAttribute("ptres")

public List<SensorEntity> getPTres() throws DataAccessException, ValidationException {

return sensorService.getPTres();

}

@ModelAttribute("tdvig")

public List<SensorEntity> getTDvig() throws DataAccessException, ValidationException {

return sensorService.getTDvig();

}

}

## Текст програми DataManagmentController.java

package com.locostatmanager.busines.controllers;

import com.locostatmanager.busines.dao.entities.LocoDataEntity;

import com.locostatmanager.busines.dao.entities.SensorEntity;

import com.locostatmanager.busines.exceptions.DataAccessException;

import com.locostatmanager.busines.exceptions.ValidationException;

import com.locostatmanager.busines.message.ResponseError;

import com.locostatmanager.busines.message.ResponseOK;

import com.locostatmanager.busines.service.LocomotiveService;

import com.locostatmanager.busines.service.SensorService;

import com.locostatmanager.busines.service.StatisticService;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.MediaType;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ExceptionHandler;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.ResponseBody;

/\*\*

\*

\* @author dmitry

\*/

@Controller

@RequestMapping(value = "/statistic")

public class DataManagmentController {

@Autowired

private StatisticService statisticService;

@Autowired

private LocomotiveService locomotiveService;

@Autowired

private SensorService sensorService;

@RequestMapping(value = "/sensor", method = RequestMethod.GET, produces = MediaType.APPLICATION\_JSON\_VALUE)

@ResponseBody

public SensorEntity getSensor() throws DataAccessException, ValidationException {

SensorEntity entity = sensorService.getByName("NAPR\_ACCUM\_BATR\_SEC\_A");

return entity;

}

@RequestMapping(value = "/locoStat.json", method = RequestMethod.GET, produces = MediaType.APPLICATION\_JSON\_VALUE)

@ResponseBody

public List<LocoDataEntity> getLocodata(@RequestParam String sdate, @RequestParam String edate, @RequestParam String id) throws DataAccessException, ValidationException {

return statisticService.getBetween(sdate, edate, id);

}

@RequestMapping(value = "/addLoco", method = RequestMethod.GET, produces = MediaType.APPLICATION\_JSON\_VALUE)

@ResponseBody

public ResponseOK addLocomotive(@RequestParam String id, @RequestParam String title) throws ValidationException, DataAccessException {

locomotiveService.add(id, title);

return new ResponseOK();

}

@RequestMapping(value = "/deleteLoco", method = RequestMethod.GET, produces = MediaType.APPLICATION\_JSON\_VALUE)

@ResponseBody

public ResponseOK deleteLocomotive(@RequestParam String id) throws ValidationException, DataAccessException {

locomotiveService.delete(id);

return new ResponseOK();

}

@ExceptionHandler({ValidationException.class, DataAccessException.class, Exception.class})

@ResponseBody

public ResponseError onException(Exception e) {

return new ResponseError(e.getMessage());

}

}

## Текст програми DownloadDataController.java

package com.locostatmanager.busines.controllers;

import com.locostatmanager.busines.dao.FileStructureInfoDao;

import com.locostatmanager.busines.dao.entities.FileStructureInfo;

import com.locostatmanager.busines.dao.entities.LocoEntity;

import com.locostatmanager.busines.exceptions.DataAccessException;

import com.locostatmanager.busines.exceptions.ValidationException;

import com.locostatmanager.busines.service.DataLoadingService;

import com.locostatmanager.busines.service.FileCheckInService;

import com.locostatmanager.busines.service.LocomotiveService;

import java.io.IOException;

import java.util.List;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.ExceptionHandler;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.multipart.MultipartFile;

import org.springframework.web.multipart.MultipartHttpServletRequest;

import org.springframework.web.servlet.ModelAndView;

/\*\*

\*

\* @author dmitry

\*/

@Controller()

@RequestMapping(value = "/downloadData.htm")

public class DownloadDataController {

@Autowired

private LocomotiveService locomotiveService;

@Autowired

private FileStructureInfoDao fileStructureInfoDao;

@Autowired

private DataLoadingService dataLoadingService;

@Autowired

private FileCheckInService checkInService;

@RequestMapping(method = RequestMethod.GET)

public String getPage(Model model) {

return "downloadData";

}

@RequestMapping(method = RequestMethod.POST)

public String uploadFiles(

MultipartHttpServletRequest request,

@RequestParam String fileType,

@RequestParam String locoId,

Model model) throws Exception {

Map<String, MultipartFile> fileMap = request.getFileMap();

for (MultipartFile file : fileMap.values()) {

if (!file.getOriginalFilename().contains(".dat")) {

throw new ValidationException("Файл має невірне розширення або покодженно!");

}

checkInService.checkIn(file.getOriginalFilename(), (int) file.getSize());

dataLoadingService.loadData(file.getBytes(), fileType);

}

model.addAttribute("success", "We will rock you!");// сообщение пока не используется, нужно для красивой зеленой строчки

return "downloadData";

}

@ModelAttribute("locomotives")

public List<LocoEntity> getLocomotives() throws ValidationException, DataAccessException {

return locomotiveService.getAll();

}

@ModelAttribute("fileTypes")

public List<FileStructureInfo> getFileNames() throws DataAccessException {

return fileStructureInfoDao.getAll();

}

// @ExceptionHandler(Exception.class)

// public ModelAndView onException(Exception e) {

//

// ModelAndView mv = new ModelAndView("downloadData");

// mv.addObject("errMsg", e.getMessage());

// return mv;

// }

}

## Текст програми IndexController.java

package com.locostatmanager.busines.controllers;

import com.locostatmanager.busines.exceptions.DataAccessException;

import com.locostatmanager.busines.exceptions.ValidationException;

import com.locostatmanager.busines.message.LocomotiveStatistic;

import com.locostatmanager.busines.service.LocomotiveService;

import com.locostatmanager.busines.service.SensorService;

import com.locostatmanager.busines.service.StatisticService;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import org.codehaus.jackson.JsonGenerationException;

import org.codehaus.jackson.map.ObjectMapper;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

/\*\*

\*

\* @author dmitry

\*/

@Controller

@RequestMapping(value = {"/", "/index.htm"})

public class IndexController {

@Autowired

private LocomotiveService locomotiveService;

@Autowired

private SensorService sensorService;

@Autowired

private StatisticService statisticService;

@RequestMapping(method = RequestMethod.GET)

public String getPage(Model model) {

return "index";

}

@ModelAttribute("sensorCount")

public String getSensorCount() throws ValidationException, DataAccessException {

return sensorService.getCount();

}

@ModelAttribute("locoCount")

public String getLocoCount() throws ValidationException, DataAccessException {

return locomotiveService.getCount();

}

@ModelAttribute("lastUpdate")

public String getLustUpdate() throws ValidationException, DataAccessException {

return new SimpleDateFormat("dd.MM.yyyy").format(new Date());

}

@ModelAttribute("dbSize")

public String getDbSize() throws ValidationException, DataAccessException {

return "176.7MB";

}

@ModelAttribute("locoRatio")

public List<LocomotiveStatistic> getLocoRatio() throws ValidationException, DataAccessException, IOException {

return statisticService.getLocomotivesRatio();

}

@ModelAttribute("locoPercentage")

public List<LocomotiveStatistic> getLocoPercentage() throws ValidationException, DataAccessException {

return statisticService.getLocomotivesPercentage();

}

}

## Текст програми LocomotiveController.java

package com.locostatmanager.busines.controllers;

import com.locostatmanager.busines.dao.entities.LocoEntity;

import com.locostatmanager.busines.exceptions.DataAccessException;

import com.locostatmanager.busines.exceptions.ValidationException;

import com.locostatmanager.busines.service.LocomotiveService;

import com.locostatmanager.busines.service.StatisticService;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

/\*\*

\*

\* @author dmitry

\*/

@Controller

@RequestMapping(value = "/trains.htm")

public class LocomotiveController {

@Autowired

private LocomotiveService locomotiveService;

@RequestMapping(value = "/", method = RequestMethod.GET)

public String getPage(Model model) {

return "trains";

}

@ModelAttribute("locomotives")

public List<LocoEntity> getLocomotives() throws ValidationException, DataAccessException {

return locomotiveService.getAll();

}

}

## Текст програми DataDaoImpl.java

package com.locostatmanager.busines.dao;

import com.locostatmanager.busines.dao.entities.LocoDataEntity;

import com.locostatmanager.busines.dao.entities.LocoEntity;

import com.locostatmanager.busines.exceptions.DataAccessException;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Timestamp;

import java.util.List;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.jdbc.core.support.JdbcDaoSupport;

/\*\*

\*

\* @author dmitry

\*/

public class DataDaoImpl extends JdbcDaoSupport implements DataDao {

@Override

public void add(LocoDataEntity entity) throws DataAccessException {

// try {

getJdbcTemplate().update("INSERT INTO LOCO\_DATA "

+ "(ID\_LOCO,"

+ "RECORD\_TIME\_LOCO\_DATA,"

+ "NAPR\_ACCUM\_BATR\_SEC\_A,"

+ "NAPR\_ACCUM\_BATR\_SEC\_B,"

+ "TOK\_ACCUM\_BATR\_SEC\_A,"

+ "TOK\_ACCUM\_BATR\_SEC\_B,"

+ "TOK\_VOZ\_TED\_SEC\_A,"

+ "TOK\_VOZ\_TED\_SEC\_B,"

+ "TOK\_YAK\_TED12\_SEC\_A,"

+ "TOK\_YAK\_TED12\_SEC\_B,"

+ "TOK\_YAK\_TED34\_SEC\_A,"

+ "TOK\_YAK\_TED34\_SEC\_B,"

+ "NAPR\_CONT\_SET\_SEC\_A,"

+ "NAPR\_CONT\_SET\_SEC\_B,"

+ "TEMPR\_BUKS1\_STOR\_MASH\_SEC\_A,"

+ "TEMPR\_BUKS1\_STOR\_POM\_MASH\_SEC\_A,"

+ "TEMPR\_BUKS2\_STOR\_MASH\_SEC\_A,"

+ "TEMPR\_BUKS2\_STOR\_POM\_MASH\_SEC\_A,"

+ "TEMPR\_BUKS3\_STOR\_MASH\_SEC\_A,"

+ "TEMPR\_BUKS3\_STOR\_POM\_MASH\_SEC\_A,"

+ "TEMPR\_BUKS4\_STOR\_MASH\_SEC\_A,"

+ "TEMPR\_BUKS4\_STOR\_POM\_MASH\_SEC\_A,"

+ "TEMPR\_BUKS4\_STOR\_POM\_MASH\_SEC\_B,"

+ "TEMPR\_BUKS4\_STOR\_MASH\_SEC\_B,"

+ "TEMPR\_BUKS3\_STOR\_POM\_MASH\_SEC\_B,"

+ "TEMPR\_BUKS3\_STOR\_MASH\_SEC\_B,"

+ "TEMPR\_BUKS2\_STOR\_POM\_MASH\_SEC\_B,"

+ "TEMPR\_BUKS2\_STOR\_MASH\_SEC\_B,"

+ "TEMPR\_BUKS1\_STOR\_POM\_MASH\_SEC\_B,"

+ "TEMPR\_BUKS1\_STOR\_MASH\_SEC\_B,"

+ "TEMPR\_VHOD\_TED\_SEC\_A1,"

+ "TEMPR\_VHOD\_TED\_SEC\_A2,"

+ "TEMPR\_VHOD\_TED\_SEC\_B1,"

+ "TEMPR\_VHOD\_TED\_SEC\_B2,"

+ "TEMPR\_VIHOD\_TED1\_SEC\_A,"

+ "TEMPR\_VIHOD\_TED2\_SEC\_A,"

+ "TEMPR\_VIHOD\_TED3\_SEC\_A,"

+ "TEMPR\_VIHOD\_TED4\_SEC\_A,"

+ "TEMPR\_VIHOD\_TED1\_SEC\_B,"

+ "TEMPR\_VIHOD\_TED2\_SEC\_B,"

+ "TEMPR\_VIHOD\_TED3\_SEC\_B,"

+ "TEMPR\_VIHOD\_TED4\_SEC\_B,"

+ "TEMPR\_BPTR1\_SEC\_A,"

+ "TEMPR\_BPTR2\_SEC\_A,"

+ "TEMPR\_BPTR1\_SEC\_B,"

+ "TEMPR\_BPTR2\_SEC\_B) "

+ "VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?"

+ ")",

new Object[]{

entity.getIdLoco(),

entity.getRecordTimeLocoData(),

entity.getNaprAccumBatrSecA(),

entity.getNaprAccumBatrSecB(),

entity.getTokAccumBatrSecA(),

entity.getTokAccumBatrSecB(),

entity.getTokVozTedSecA(),

entity.getTokVozTedSecB(),

entity.getTokYakTed12SecA(),

entity.getTokYakTed12SecB(),

entity.getTokYakTed34SecA(),

entity.getTokYakTed34SecB(),

entity.getNaprContSetSecA(),

entity.getNaprContSetSecB(),

entity.getTemprBuks1StorMashSecA(),

entity.getTemprBuks1StorPomMashSecA(),

entity.getTemprBuks2StorMashSecA(),

entity.getTemprBuks2StorPomMashSecA(),

entity.getTemprBuks3StorMashSecA(),

entity.getTemprBuks3StorPomMashSecA(),

entity.getTemprBuks4StorMashSecA(),

entity.getTemprBuks4StorPomMashSecA(),

entity.getTemprBuks4StorPomMashSecB(),

entity.getTemprBuks4StorMashSecB(),

entity.getTemprBuks3StorPomMashSecB(),

entity.getTemprBuks3StorMashSecB(),

entity.getTemprBuks2StorPomMashSecB(),

entity.getTemprBuks2StorMashSecB(),

entity.getTemprBuks1StorPomMashSecB(),

entity.getTemprBuks1StorMashSecB(),

entity.getTemprVhodTedSecA1(),

entity.getTemprVhodTedSecA2(),

entity.getTemprVhodTedSecB1(),

entity.getTemprVhodTedSecB2(),

entity.getTemprVihodTed1SecA(),

entity.getTemprVihodTed2SecA(),

entity.getTemprVihodTed3SecA(),

entity.getTemprVihodTed4SecA(),

entity.getTemprVihodTed1SecB(),

entity.getTemprVihodTed2SecB(),

entity.getTemprVihodTed3SecB(),

entity.getTemprVihodTed4SecB(),

entity.getTemprBptr1SecA(),

entity.getTemprBptr2SecA(),

entity.getTemprBptr1SecB(),

entity.getTemprBptr2SecB()}

);

// } catch (org.springframework.dao.DataAccessException e) {

// throw new DataAccessException(e);

// }

}

@Override

public List<LocoDataEntity> getAll() throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM LOCO\_DATA", new LocoDataRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public List<LocoDataEntity> getAfter(Timestamp date) throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM LOCO\_DATA WHERE RECORD\_TIME\_LOCO\_DATA > ?",

new Object[]{date},

new LocoDataRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public List<LocoDataEntity> getBefore(Timestamp date) throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM LOCO\_DATA WHERE RECORD\_TIME\_LOCO\_DATA < ?",

new Object[]{date},

new LocoDataRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public List<LocoDataEntity> getBetween(Timestamp startDate, Timestamp endDate) throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM LOCO\_DATA WHERE RECORD\_TIME\_LOCO\_DATA > ? AND RECORD\_TIME\_LOCO\_DATA < ?",

new Object[]{startDate, endDate},

new LocoDataRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public List<LocoDataEntity> getAfter(Timestamp date, LocoEntity entity) throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM LOCO\_DATA WHERE RECORD\_TIME\_LOCO\_DATA > ? AND ID\_LOCO = ?",

new Object[]{date, entity.getIdLoco()},

new LocoDataRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public List<LocoDataEntity> getBefore(Timestamp date, LocoEntity entity) throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM LOCO\_DATA WHERE RECORD\_TIME\_LOCO\_DATA < ? AND ID\_LOCO = ?",

new Object[]{date, entity.getIdLoco()},

new LocoDataRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public List<LocoDataEntity> getBetween(Timestamp startDate, Timestamp endDate, String idLoco) throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM LOCO\_DATA WHERE RECORD\_TIME\_LOCO\_DATA > ? AND RECORD\_TIME\_LOCO\_DATA < ? AND ID\_LOCO = ?",

new Object[]{startDate, endDate, idLoco},

new LocoDataRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public List<LocoDataEntity> getByLocomotive(LocoEntity entity) throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM LOCO\_DATA WHERE ID\_LOCO = ?",

new Object[]{entity.getIdLoco()},

new LocoDataRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public LocoDataEntity getById(String id) throws DataAccessException {

try {

return getJdbcTemplate().queryForObject("SELECT \* FROM LOCO\_DATA WHERE ID\_LOCO\_DATA = ?",

new Object[]{id},

new LocoDataRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public List<LocoDataEntity> getBetween(String startDate, String endDate, String locomotiveId) throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM LOCO\_DATA WHERE RECORD\_TIME\_LOCO\_DATA > ? AND RECORD\_TIME\_LOCO\_DATA < ? AND ID\_LOCO = ?",

new Object[]{startDate, endDate, locomotiveId},

new LocoDataRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public int getRecordsCount(String LocoId) throws DataAccessException {

try {

return getJdbcTemplate().queryForObject("SELECT COUNT(\*) FROM LOCO\_DATA WHERE ID\_LOCO = ?", Integer.class, new Object[]{LocoId});

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public int getCount() throws DataAccessException {

try {

return getJdbcTemplate().queryForObject("SELECT COUNT(\*) FROM LOCO\_DATA", Integer.class);

} catch (Exception e) {

throw new DataAccessException(e);

}

}

private class LocoDataRowMapper implements RowMapper<LocoDataEntity> {

public LocoDataEntity mapRow(ResultSet rs, int i) throws SQLException {

LocoDataEntity dataEntity = new LocoDataEntity();

dataEntity.setIdLocoData(rs.getLong("ID\_LOCO\_DATA"));

dataEntity.setIdLoco(trim(rs.getString("ID\_LOCO")));

dataEntity.setRecordTimeLocoData(rs.getTimestamp("RECORD\_TIME\_LOCO\_DATA"));

dataEntity.setNaprAccumBatrSecA(rs.getByte("NAPR\_ACCUM\_BATR\_SEC\_A"));

dataEntity.setNaprAccumBatrSecB(rs.getByte("NAPR\_ACCUM\_BATR\_SEC\_B"));

dataEntity.setTokAccumBatrSecA(rs.getByte("TOK\_ACCUM\_BATR\_SEC\_A"));

dataEntity.setTokAccumBatrSecB(rs.getByte("TOK\_ACCUM\_BATR\_SEC\_B"));

dataEntity.setTokVozTedSecA(rs.getShort("TOK\_VOZ\_TED\_SEC\_A"));

dataEntity.setTokVozTedSecB(rs.getShort("TOK\_VOZ\_TED\_SEC\_B"));

dataEntity.setTokYakTed12SecA(rs.getShort("TOK\_YAK\_TED12\_SEC\_A"));

dataEntity.setTokYakTed12SecB(rs.getShort("TOK\_YAK\_TED12\_SEC\_B"));

dataEntity.setTokYakTed34SecA(rs.getShort("TOK\_YAK\_TED34\_SEC\_A"));

dataEntity.setTokYakTed34SecB(rs.getShort("TOK\_YAK\_TED34\_SEC\_B"));

dataEntity.setNaprContSetSecA(rs.getShort("NAPR\_CONT\_SET\_SEC\_A"));

dataEntity.setNaprContSetSecB(rs.getShort("NAPR\_CONT\_SET\_SEC\_B"));

dataEntity.setTemprBuks1StorMashSecA(rs.getByte("TEMPR\_BUKS1\_STOR\_MASH\_SEC\_A"));

dataEntity.setTemprBuks1StorPomMashSecA(rs.getByte("TEMPR\_BUKS1\_STOR\_POM\_MASH\_SEC\_A"));

dataEntity.setTemprBuks2StorMashSecA(rs.getByte("TEMPR\_BUKS2\_STOR\_MASH\_SEC\_A"));

dataEntity.setTemprBuks2StorPomMashSecA(rs.getByte("TEMPR\_BUKS2\_STOR\_POM\_MASH\_SEC\_A"));

dataEntity.setTemprBuks3StorMashSecA(rs.getByte("TEMPR\_BUKS3\_STOR\_MASH\_SEC\_A"));

dataEntity.setTemprBuks3StorPomMashSecA(rs.getByte("TEMPR\_BUKS3\_STOR\_POM\_MASH\_SEC\_A"));

dataEntity.setTemprBuks4StorMashSecA(rs.getByte("TEMPR\_BUKS4\_STOR\_MASH\_SEC\_A"));

dataEntity.setTemprBuks4StorPomMashSecA(rs.getByte("TEMPR\_BUKS4\_STOR\_POM\_MASH\_SEC\_A"));

dataEntity.setTemprBuks4StorPomMashSecB(rs.getByte("TEMPR\_BUKS4\_STOR\_POM\_MASH\_SEC\_B"));

dataEntity.setTemprBuks4StorMashSecB(rs.getByte("TEMPR\_BUKS4\_STOR\_MASH\_SEC\_B"));

dataEntity.setTemprBuks3StorPomMashSecB(rs.getByte("TEMPR\_BUKS3\_STOR\_POM\_MASH\_SEC\_B"));

dataEntity.setTemprBuks3StorMashSecB(rs.getByte("TEMPR\_BUKS3\_STOR\_MASH\_SEC\_B"));

dataEntity.setTemprBuks2StorPomMashSecB(rs.getByte("TEMPR\_BUKS2\_STOR\_POM\_MASH\_SEC\_B"));

dataEntity.setTemprBuks2StorMashSecB(rs.getByte("TEMPR\_BUKS2\_STOR\_MASH\_SEC\_B"));

dataEntity.setTemprBuks1StorPomMashSecB(rs.getByte("TEMPR\_BUKS1\_STOR\_POM\_MASH\_SEC\_B"));

dataEntity.setTemprBuks1StorMashSecB(rs.getByte("TEMPR\_BUKS1\_STOR\_MASH\_SEC\_B"));

dataEntity.setTemprVhodTedSecA1(rs.getByte("TEMPR\_VHOD\_TED\_SEC\_A1"));

dataEntity.setTemprVhodTedSecA2(rs.getByte("TEMPR\_VHOD\_TED\_SEC\_A2"));

dataEntity.setTemprVhodTedSecB1(rs.getByte("TEMPR\_VHOD\_TED\_SEC\_B1"));

dataEntity.setTemprVhodTedSecB2(rs.getByte("TEMPR\_VHOD\_TED\_SEC\_B2"));

dataEntity.setTemprVihodTed1SecA(rs.getByte("TEMPR\_VIHOD\_TED1\_SEC\_A"));

dataEntity.setTemprVihodTed2SecA(rs.getByte("TEMPR\_VIHOD\_TED2\_SEC\_A"));

dataEntity.setTemprVihodTed3SecA(rs.getByte("TEMPR\_VIHOD\_TED3\_SEC\_A"));

dataEntity.setTemprVihodTed4SecA(rs.getByte("TEMPR\_VIHOD\_TED4\_SEC\_A"));

dataEntity.setTemprVihodTed1SecB(rs.getByte("TEMPR\_VIHOD\_TED1\_SEC\_B"));

dataEntity.setTemprVihodTed2SecB(rs.getByte("TEMPR\_VIHOD\_TED2\_SEC\_B"));

dataEntity.setTemprVihodTed3SecB(rs.getByte("TEMPR\_VIHOD\_TED3\_SEC\_B"));

dataEntity.setTemprVihodTed4SecB(rs.getByte("TEMPR\_VIHOD\_TED4\_SEC\_B"));

dataEntity.setTemprBptr1SecA(rs.getShort("TEMPR\_BPTR1\_SEC\_A"));

dataEntity.setTemprBptr2SecA(rs.getShort("TEMPR\_BPTR2\_SEC\_A"));

dataEntity.setTemprBptr1SecB(rs.getShort("TEMPR\_BPTR1\_SEC\_B"));

dataEntity.setTemprBptr2SecB(rs.getShort("TEMPR\_BPTR2\_SEC\_B"));

return dataEntity;

}

}

private String trim(String str){

return str == null ? null : str.trim();

}

}

## Текст програми FileInfoDaoImpl.java

package com.locostatmanager.busines.dao;

import com.locostatmanager.busines.dao.entities.FileInfo;

import com.locostatmanager.busines.exceptions.DataAccessException;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Timestamp;

import java.util.Calendar;

import org.springframework.dao.EmptyResultDataAccessException;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.jdbc.core.support.JdbcDaoSupport;

/\*\*

\*

\* @author dmitry

\*/

public class FileInfoDaoImpl extends JdbcDaoSupport implements FileInfoDao {

public FileInfo get(String fileName, Integer fileSize) throws DataAccessException {

try {

return getJdbcTemplate().queryForObject("SELECT \* FROM FILE\_INFO WHERE FILE\_NAME = ? AND FILE\_SIZE = ?", new Object[]{fileName, fileSize}, new FileInfoMapper());

} catch (EmptyResultDataAccessException e) {

return null;

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public void add(FileInfo fileInfo) throws DataAccessException {

try {

getJdbcTemplate().update("INSERT INTO FILE\_INFO (FILE\_NAME, FILE\_SIZE, LOADING\_DATE) VALUES (?, ?, ?)",

new Object[]{fileInfo.getFileName(), fileInfo.getFileSize(), new Timestamp(Calendar.getInstance().getTimeInMillis())});

} catch (Exception e) {

throw new DataAccessException(e);

}

}

private class FileInfoMapper implements RowMapper<FileInfo> {

public FileInfo mapRow(ResultSet rs, int i) throws SQLException {

FileInfo fileInfo = new FileInfo();

fileInfo.setFileName(rs.getString("FILE\_NAME"));

fileInfo.setFileSize(rs.getInt("FILE\_SIZE"));

fileInfo.setLoadingDate(rs.getTimestamp("LOADING\_DATE"));

return fileInfo;

}

}

}

## Текст програми FileStructureInfoDaoImpl.java

package com.locostatmanager.busines.dao;

import com.locostatmanager.busines.dao.entities.FileStructureInfo;

import com.locostatmanager.busines.exceptions.DataAccessException;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.List;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.jdbc.core.support.JdbcDaoSupport;

import org.springframework.transaction.annotation.Transactional;

/\*\*

\*

\* @author dmitry

\*/

public class FileStructureInfoDaoImpl extends JdbcDaoSupport implements FileStructureInfoDao {

public FileStructureInfo get(String type) {

return getJdbcTemplate().queryForObject("SELECT \* FROM FILE\_STRUCTURE\_INFO WHERE TYPE = ?", new Object[]{type}, new FileStructureInfoMapper());

}

@Transactional

public List<FileStructureInfo> getAll() throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM FILE\_STRUCTURE\_INFO", new FileStructureInfoMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

private class FileStructureInfoMapper implements RowMapper<FileStructureInfo> {

public FileStructureInfo mapRow(ResultSet rs, int i) throws SQLException {

FileStructureInfo fileStructureInfo = new FileStructureInfo();

fileStructureInfo.setId(rs.getLong("ID"));

fileStructureInfo.setRecordLength(rs.getInt("RECORD\_LENGTH"));

fileStructureInfo.setRecordsStartIndex(rs.getInt("RECORD\_START\_INDEX"));

fileStructureInfo.setType(rs.getString("TYPE"));

return fileStructureInfo;

}

}

}

## Текст програми LocomotiveDaoImpl.java

package com.locostatmanager.busines.dao;

import com.locostatmanager.busines.dao.entities.LocoEntity;

import com.locostatmanager.busines.exceptions.DataAccessException;

import java.sql.ResultSet;

import java.sql.SQLException;

import org.springframework.jdbc.core.support.JdbcDaoSupport;

import java.util.List;

import org.springframework.jdbc.core.RowMapper;

/\*\*

\* Created by anatoliy on 23.04.14.

\*/

public class LocomotiveDaoImpl extends JdbcDaoSupport implements LocomotiveDao {

@Override

public void add(LocoEntity entity) throws DataAccessException {

try {

getJdbcTemplate().update("INSERT INTO LOCO (ID\_LOCO, TITLE\_LOCO) VALUES (?,?)",

new Object[]{entity.getIdLoco(), entity.getTitleLoco()});

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public List<LocoEntity> getAll() throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM LOCO", new LocoRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public LocoEntity getById(String id) throws DataAccessException {

try {

return getJdbcTemplate().queryForObject("SELECT \* FROM LOCO WHERE ID\_LOCO = ?", new Object[]{id}, new LocoRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public LocoEntity getByTitle(String title) throws DataAccessException {

try {

return getJdbcTemplate().queryForObject("SELECT \* FROM LOCO WHERE TITLE\_LOCO = ?", new Object[]{title}, new LocoRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public String getCount() throws DataAccessException {

try {

return getJdbcTemplate().queryForObject("SELECT COUNT(\*) FROM LOCO", String.class);

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public void delete(String id) throws DataAccessException {

try {

getJdbcTemplate().update("DELETE FROM LOCO WHERE ID\_LOCO = ?", new Object[]{id});

} catch (Exception e) {

throw new DataAccessException(e);

}

}

private class LocoRowMapper implements RowMapper<LocoEntity> {

public LocoEntity mapRow(ResultSet rs, int i) throws SQLException {

LocoEntity locoEntity = new LocoEntity();

locoEntity.setIdLoco(trim(rs.getString("ID\_LOCO")));

locoEntity.setTitleLoco(trim(rs.getString("TITLE\_LOCO")));

return locoEntity;

}

}

private String trim(String str){

return str == null ? null : str.trim();

}

}

## Текст програми RecordStructureInfoDaoImpl.java

package com.locostatmanager.busines.dao;

import com.locostatmanager.busines.dao.entities.RecordStructureInfo;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.List;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.jdbc.core.support.JdbcDaoSupport;

/\*\*

\*

\* @author dmitry

\*/

public class RecordStructureInfoDaoImpl extends JdbcDaoSupport implements RecordStructureInfoDao {

public List<RecordStructureInfo> getByFsiType(String fsiType) {

return getJdbcTemplate().query("SELECT \* FROM RECORD\_STRUCTURE\_INFO WHERE FSI\_TYPE = ?", new Object[]{fsiType}, new RecordStructureInfoMapper());

}

private class RecordStructureInfoMapper implements RowMapper<RecordStructureInfo>{

public RecordStructureInfo mapRow(ResultSet rs, int i) throws SQLException {

RecordStructureInfo recordStructureInfo = new RecordStructureInfo();

recordStructureInfo.setFsiType(rs.getString("FSI\_TYPE"));

recordStructureInfo.setName(rs.getString("NAME"));

recordStructureInfo.setIndex(rs.getInt("INDEx"));

recordStructureInfo.setDelta(rs.getDouble("DELTA"));

return recordStructureInfo;

}

}

}

## Текст програми SensorDaoImpl.java

package com.locostatmanager.busines.dao;

import com.locostatmanager.busines.dao.entities.SensorEntity;

import com.locostatmanager.busines.exceptions.DataAccessException;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.List;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.jdbc.core.support.JdbcDaoSupport;

/\*\*

\*

\* @author dmitry

\*/

public class SensorDaoImpl extends JdbcDaoSupport implements SensorDao {

@Override

public void add(SensorEntity entity) throws DataAccessException {

try {

getJdbcTemplate().update("INSERT INTO SENSOR (NAME, DESCRIPTION, UNIT\_OF\_MEASURE) VALUES (?,?,?)",

new Object[]{entity.getName(), entity.getDescription(), entity.getUnitOfMeasure()});

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public List<SensorEntity> getAll() throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM SENSOR", new SensorRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

@Override

public SensorEntity getByName(String name) throws DataAccessException {

try {

return getJdbcTemplate().queryForObject("SELECT \* FROM SENSOR WHERE NAME = ?", new Object[]{name}, new SensorRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public String getCount() throws DataAccessException {

try {

return getJdbcTemplate().queryForObject("SELECT COUNT(\*) FROM SENSOR", String.class);

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public List<SensorEntity> getByUnitName(String unitName) throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM SENSOR WHERE UNIT\_NAME = ?",new Object[] {unitName}, new SensorRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

private class SensorRowMapper implements RowMapper<SensorEntity> {

public SensorEntity mapRow(ResultSet rs, int i) throws SQLException {

SensorEntity sensorEntity = new SensorEntity();

sensorEntity.setName(trim(rs.getString("NAME")));

sensorEntity.setDescription(trim(rs.getString("DESCRIPTION")));

sensorEntity.setUnitOfMeasure(trim(rs.getString("UNIT\_OF\_MEASURE")));

return sensorEntity;

}

}

private String trim(String str){

return str == null ? null : str.trim();

}

}

## Текст програми UserDaoImpl.java

package com.locostatmanager.busines.dao;

import com.locostatmanager.busines.dao.entities.UserEntity;

import com.locostatmanager.busines.exceptions.DataAccessException;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.jdbc.core.support.JdbcDaoSupport;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.List;

/\*\*

\* Created by anatoliy on 01.06.14.

\*/

public class UserDaoImpl extends JdbcDaoSupport implements UserDao {

public List<UserEntity> getAllUsers() throws DataAccessException {

try {

return getJdbcTemplate().query("SELECT \* FROM USER", new UserRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public UserEntity getUserByLogin(String login) throws DataAccessException {

try {

return getJdbcTemplate().queryForObject(

"SELECT \* FROM USER WHERE LOGIN = ?",

new Object[]{login},

new UserRowMapper());

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public void addUser(UserEntity entity) throws DataAccessException {

try {

getJdbcTemplate().update("INSERT INTO USER (LOGIN, NAME, PASSWORD, ROLE) VALUES (?,?,?,?)",

new Object[]{entity.getLogin(), entity.getName(), entity.getPassword(), entity.getRole()});

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public void updateUserName(String login, String newName) throws DataAccessException {

try {

getJdbcTemplate().update("UPDATE USER SET NAME = ? WHERE LOGIN = ?",

new Object[]{newName, login});

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public void updateUserPassword(String login, String newPassword) throws DataAccessException {

try {

getJdbcTemplate().update("UPDATE USER SET PASSWORD = ? WHERE LOGIN = ?",

new Object[]{newPassword, login});

} catch (Exception e) {

throw new DataAccessException(e);

}

}

public void deleteUser(String login) throws DataAccessException {

try {

getJdbcTemplate().update("DELETE FROM USER WHERE LOGIN = ?",

new Object[]{login});

} catch (Exception e) {

throw new DataAccessException(e);

}

}

private class UserRowMapper implements RowMapper<UserEntity> {

public UserEntity mapRow(ResultSet rs, int i) throws SQLException {

UserEntity entity = new UserEntity();

entity.setLogin(trim(rs.getString("LOGIN")));

entity.setName(trim(rs.getString("NAME")));

entity.setPassword(rs.getString("PASSWORD"));

entity.setRole(trim(rs.getString("ROLE")));

return entity;

}

}

private String trim(String str) {

return str == null ? null : str.trim();

}

}

## Текст програми FileInfo.java

package com.locostatmanager.busines.dao.entities;

import java.sql.Timestamp;

/\*\*

\*

\* @author dmitry

\*/

public class FileInfo {

private String fileName;

private Integer fileSize;

private Timestamp loadingDate;

public String getFileName() {

return fileName;

}

public void setFileName(String fileName) {

this.fileName = fileName;

}

public Integer getFileSize() {

return fileSize;

}

public void setFileSize(Integer fileSize) {

this.fileSize = fileSize;

}

public Timestamp getLoadingDate() {

return loadingDate;

}

public void setLoadingDate(Timestamp loadingDate) {

this.loadingDate = loadingDate;

}

}

## Текст програми FileStructureInfo.java

package com.locostatmanager.busines.dao.entities;

/\*\*

\*

\* @author dmitry

\*/

public class FileStructureInfo {

private Long id;

private Integer recordsStartIndex;

private Integer recordLength;

private String type;

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Integer getRecordsStartIndex() {

return recordsStartIndex;

}

public void setRecordsStartIndex(Integer recordsStartIndex) {

this.recordsStartIndex = recordsStartIndex;

}

public Integer getRecordLength() {

return recordLength;

}

public void setRecordLength(Integer recordLength) {

this.recordLength = recordLength;

}

}

## Текст програми LocoDataEntity.java

package com.locostatmanager.busines.dao.entities;

import org.codehaus.jackson.annotate.JsonProperty;

import java.sql.Timestamp;

/\*\*

\* Created by anatoliy on 22.04.14.

\*/

public class LocoDataEntity {

@JsonProperty("ID\_LOCO\_DATA")

private long idLocoData=0;

@JsonProperty("ID\_LOCO")

private String idLoco;

@JsonProperty("RECORD\_TIME\_LOCO\_DATA")

private Timestamp recordTimeLocoData;

@JsonProperty("NAPR\_ACCUM\_BATR\_SEC\_A")

private Byte naprAccumBatrSecA=0;

@JsonProperty("NAPR\_ACCUM\_BATR\_SEC\_B")

private Byte naprAccumBatrSecB=0;

@JsonProperty("TOK\_ACCUM\_BATR\_SEC\_A")

private Byte tokAccumBatrSecA=0;

@JsonProperty("TOK\_ACCUM\_BATR\_SEC\_B")

private Byte tokAccumBatrSecB=0;

@JsonProperty("TOK\_VOZ\_TED\_SEC\_A")

private Short tokVozTedSecA=0;

@JsonProperty("TOK\_VOZ\_TED\_SEC\_B")

private Short tokVozTedSecB=0;

@JsonProperty("TOK\_YAK\_TED12\_SEC\_A")

private Short tokYakTed12SecA=0;

@JsonProperty("TOK\_YAK\_TED12\_SEC\_B")

private Short tokYakTed12SecB=0;

@JsonProperty("TOK\_YAK\_TED34\_SEC\_A")

private Short tokYakTed34SecA=0;

@JsonProperty("TOK\_YAK\_TED34\_SEC\_B")

private Short tokYakTed34SecB=0;

@JsonProperty("NAPR\_CONT\_SET\_SEC\_A")

private Short naprContSetSecA=0;

@JsonProperty("NAPR\_CONT\_SET\_SEC\_B")

private Short naprContSetSecB=0;

@JsonProperty("TEMPR\_BUKS1\_STOR\_MASH\_SEC\_A")

private Byte temprBuks1StorMashSecA=0;

@JsonProperty("TEMPR\_BUKS1\_STOR\_POM\_MASH\_SEC\_A")

private Byte temprBuks1StorPomMashSecA=0;

@JsonProperty("TEMPR\_BUKS2\_STOR\_MASH\_SEC\_A")

private Byte temprBuks2StorMashSecA=0;

@JsonProperty("TEMPR\_BUKS2\_STOR\_POM\_MASH\_SEC\_A")

private Byte temprBuks2StorPomMashSecA=0;

@JsonProperty("TEMPR\_BUKS3\_STOR\_MASH\_SEC\_A")

private Byte temprBuks3StorMashSecA=0;

@JsonProperty("TEMPR\_BUKS3\_STOR\_POM\_MASH\_SEC\_A")

private Byte temprBuks3StorPomMashSecA=0;

@JsonProperty("TEMPR\_BUKS4\_STOR\_MASH\_SEC\_A")

private Byte temprBuks4StorMashSecA=0;

@JsonProperty("TEMPR\_BUKS4\_STOR\_POM\_MASH\_SEC\_A")

private Byte temprBuks4StorPomMashSecA=0;

@JsonProperty("TEMPR\_BUKS4\_STOR\_POM\_MASH\_SEC\_B")

private Byte temprBuks4StorPomMashSecB=0;

@JsonProperty("TEMPR\_BUKS4\_STOR\_MASH\_SEC\_B")

private Byte temprBuks4StorMashSecB=0;

@JsonProperty("TEMPR\_BUKS3\_STOR\_POM\_MASH\_SEC\_B")

private Byte temprBuks3StorPomMashSecB=0;

@JsonProperty("TEMPR\_BUKS3\_STOR\_MASH\_SEC\_B")

private Byte temprBuks3StorMashSecB=0;

@JsonProperty("TEMPR\_BUKS2\_STOR\_POM\_MASH\_SEC\_B")

private Byte temprBuks2StorPomMashSecB=0;

@JsonProperty("TEMPR\_BUKS2\_STOR\_MASH\_SEC\_B")

private Byte temprBuks2StorMashSecB=0;

@JsonProperty("TEMPR\_BUKS1\_STOR\_POM\_MASH\_SEC\_B")

private Byte temprBuks1StorPomMashSecB=0;

@JsonProperty("TEMPR\_BUKS1\_STOR\_MASH\_SEC\_B")

private Byte temprBuks1StorMashSecB=0;

@JsonProperty("TEMPR\_VHOD\_TED\_SEC\_A1")

private Byte temprVhodTedSecA1=0;

@JsonProperty("TEMPR\_VHOD\_TED\_SEC\_A2")

private Byte temprVhodTedSecA2=0;

@JsonProperty("TEMPR\_VHOD\_TED\_SEC\_B1")

private Byte temprVhodTedSecB1=0;

@JsonProperty("TEMPR\_VHOD\_TED\_SEC\_B2")

private Byte temprVhodTedSecB2=0;

@JsonProperty("TEMPR\_VIHOD\_TED1\_SEC\_A")

private Byte temprVihodTed1SecA=0;

@JsonProperty("TEMPR\_VIHOD\_TED2\_SEC\_A")

private Byte temprVihodTed2SecA=0;

@JsonProperty("TEMPR\_VIHOD\_TED3\_SEC\_A")

private Byte temprVihodTed3SecA=0;

@JsonProperty("TEMPR\_VIHOD\_TED4\_SEC\_A")

private Byte temprVihodTed4SecA=0;

@JsonProperty("TEMPR\_VIHOD\_TED1\_SEC\_B")

private Byte temprVihodTed1SecB=0;

@JsonProperty("TEMPR\_VIHOD\_TED2\_SEC\_B")

private Byte temprVihodTed2SecB=0;

@JsonProperty("TEMPR\_VIHOD\_TED3\_SEC\_B")

private Byte temprVihodTed3SecB=0;

@JsonProperty("TEMPR\_VIHOD\_TED4\_SEC\_B")

private Byte temprVihodTed4SecB=0;

@JsonProperty("TEMPR\_BPTR1\_SEC\_A")

private Short temprBptr1SecA=0;

@JsonProperty("TEMPR\_BPTR2\_SEC\_A")

private Short temprBptr2SecA=0;

@JsonProperty("TEMPR\_BPTR1\_SEC\_B")

private Short temprBptr1SecB=0;

@JsonProperty("TEMPR\_BPTR2\_SEC\_B")

private Short temprBptr2SecB=0;

public long getIdLocoData() {

return idLocoData;

}

public void setIdLocoData(long idLocoData) {

this.idLocoData = idLocoData;

}

public String getIdLoco() {

return idLoco;

}

public void setIdLoco(String idLoco) {

this.idLoco = idLoco;

}

public Timestamp getRecordTimeLocoData() {

return recordTimeLocoData;

}

public void setRecordTimeLocoData(Timestamp recordTimeLocoData) {

this.recordTimeLocoData = recordTimeLocoData;

}

public Byte getNaprAccumBatrSecA() {

return naprAccumBatrSecA;

}

public void setNaprAccumBatrSecA(Byte naprAccumBatrSecA) {

this.naprAccumBatrSecA = naprAccumBatrSecA;

}

public Byte getNaprAccumBatrSecB() {

return naprAccumBatrSecB;

}

public void setNaprAccumBatrSecB(Byte naprAccumBatrSecB) {

this.naprAccumBatrSecB = naprAccumBatrSecB;

}

public Byte getTokAccumBatrSecA() {

return tokAccumBatrSecA;

}

public void setTokAccumBatrSecA(Byte tokAccumBatrSecA) {

this.tokAccumBatrSecA = tokAccumBatrSecA;

}

public Byte getTokAccumBatrSecB() {

return tokAccumBatrSecB;

}

public void setTokAccumBatrSecB(Byte tokAccumBatrSecB) {

this.tokAccumBatrSecB = tokAccumBatrSecB;

}

public Short getTokVozTedSecA() {

return tokVozTedSecA;

}

public void setTokVozTedSecA(Short tokVozTedSecA) {

this.tokVozTedSecA = tokVozTedSecA;

}

public Short getTokVozTedSecB() {

return tokVozTedSecB;

}

public void setTokVozTedSecB(Short tokVozTedSecB) {

this.tokVozTedSecB = tokVozTedSecB;

}

public Short getTokYakTed12SecA() {

return tokYakTed12SecA;

}

public void setTokYakTed12SecA(Short tokYakTed12SecA) {

this.tokYakTed12SecA = tokYakTed12SecA;

}

public Short getTokYakTed12SecB() {

return tokYakTed12SecB;

}

public void setTokYakTed12SecB(Short tokYakTed12SecB) {

this.tokYakTed12SecB = tokYakTed12SecB;

}

public Short getTokYakTed34SecA() {

return tokYakTed34SecA;

}

public void setTokYakTed34SecA(Short tokYakTed34SecA) {

this.tokYakTed34SecA = tokYakTed34SecA;

}

public Short getTokYakTed34SecB() {

return tokYakTed34SecB;

}

public void setTokYakTed34SecB(Short tokYakTed34SecB) {

this.tokYakTed34SecB = tokYakTed34SecB;

}

public Short getNaprContSetSecA() {

return naprContSetSecA;

}

public void setNaprContSetSecA(Short naprContSetSecA) {

this.naprContSetSecA = naprContSetSecA;

}

public Short getNaprContSetSecB() {

return naprContSetSecB;

}

public void setNaprContSetSecB(Short naprContSetSecB) {

this.naprContSetSecB = naprContSetSecB;

}

public Byte getTemprBuks1StorMashSecA() {

return temprBuks1StorMashSecA;

}

public void setTemprBuks1StorMashSecA(Byte temprBuks1StorMashSecA) {

this.temprBuks1StorMashSecA = temprBuks1StorMashSecA;

}

public Byte getTemprBuks1StorPomMashSecA() {

return temprBuks1StorPomMashSecA;

}

public void setTemprBuks1StorPomMashSecA(Byte temprBuks1StorPomMashSecA) {

this.temprBuks1StorPomMashSecA = temprBuks1StorPomMashSecA;

}

public Byte getTemprBuks2StorMashSecA() {

return temprBuks2StorMashSecA;

}

public void setTemprBuks2StorMashSecA(Byte temprBuks2StorMashSecA) {

this.temprBuks2StorMashSecA = temprBuks2StorMashSecA;

}

public Byte getTemprBuks2StorPomMashSecA() {

return temprBuks2StorPomMashSecA;

}

public void setTemprBuks2StorPomMashSecA(Byte temprBuks2StorPomMashSecA) {

this.temprBuks2StorPomMashSecA = temprBuks2StorPomMashSecA;

}

public Byte getTemprBuks3StorMashSecA() {

return temprBuks3StorMashSecA;

}

public void setTemprBuks3StorMashSecA(Byte temprBuks3StorMashSecA) {

this.temprBuks3StorMashSecA = temprBuks3StorMashSecA;

}

public Byte getTemprBuks3StorPomMashSecA() {

return temprBuks3StorPomMashSecA;

}

public void setTemprBuks3StorPomMashSecA(Byte temprBuks3StorPomMashSecA) {

this.temprBuks3StorPomMashSecA = temprBuks3StorPomMashSecA;

}

public Byte getTemprBuks4StorMashSecA() {

return temprBuks4StorMashSecA;

}

public void setTemprBuks4StorMashSecA(Byte temprBuks4StorMashSecA) {

this.temprBuks4StorMashSecA = temprBuks4StorMashSecA;

}

public Byte getTemprBuks4StorPomMashSecA() {

return temprBuks4StorPomMashSecA;

}

public void setTemprBuks4StorPomMashSecA(Byte temprBuks4StorPomMashSecA) {

this.temprBuks4StorPomMashSecA = temprBuks4StorPomMashSecA;

}

public Byte getTemprBuks4StorPomMashSecB() {

return temprBuks4StorPomMashSecB;

}

public void setTemprBuks4StorPomMashSecB(Byte temprBuks4StorPomMashSecB) {

this.temprBuks4StorPomMashSecB = temprBuks4StorPomMashSecB;

}

public Byte getTemprBuks4StorMashSecB() {

return temprBuks4StorMashSecB;

}

public void setTemprBuks4StorMashSecB(Byte temprBuks4StorMashSecB) {

this.temprBuks4StorMashSecB = temprBuks4StorMashSecB;

}

public Byte getTemprBuks3StorPomMashSecB() {

return temprBuks3StorPomMashSecB;

}

public void setTemprBuks3StorPomMashSecB(Byte temprBuks3StorPomMashSecB) {

this.temprBuks3StorPomMashSecB = temprBuks3StorPomMashSecB;

}

public Byte getTemprBuks3StorMashSecB() {

return temprBuks3StorMashSecB;

}

public void setTemprBuks3StorMashSecB(Byte temprBuks3StorMashSecB) {

this.temprBuks3StorMashSecB = temprBuks3StorMashSecB;

}

public Byte getTemprBuks2StorPomMashSecB() {

return temprBuks2StorPomMashSecB;

}

public void setTemprBuks2StorPomMashSecB(Byte temprBuks2StorPomMashSecB) {

this.temprBuks2StorPomMashSecB = temprBuks2StorPomMashSecB;

}

public Byte getTemprBuks2StorMashSecB() {

return temprBuks2StorMashSecB;

}

public void setTemprBuks2StorMashSecB(Byte temprBuks2StorMashSecB) {

this.temprBuks2StorMashSecB = temprBuks2StorMashSecB;

}

public Byte getTemprBuks1StorPomMashSecB() {

return temprBuks1StorPomMashSecB;

}

public void setTemprBuks1StorPomMashSecB(Byte temprBuks1StorPomMashSecB) {

this.temprBuks1StorPomMashSecB = temprBuks1StorPomMashSecB;

}

public Byte getTemprBuks1StorMashSecB() {

return temprBuks1StorMashSecB;

}

public void setTemprBuks1StorMashSecB(Byte temprBuks1StorMashSecB) {

this.temprBuks1StorMashSecB = temprBuks1StorMashSecB;

}

public Byte getTemprVhodTedSecA1() {

return temprVhodTedSecA1;

}

public void setTemprVhodTedSecA1(Byte temprVhodTedSecA1) {

this.temprVhodTedSecA1 = temprVhodTedSecA1;

}

public Byte getTemprVhodTedSecA2() {

return temprVhodTedSecA2;

}

public void setTemprVhodTedSecA2(Byte temprVhodTedSecA2) {

this.temprVhodTedSecA2 = temprVhodTedSecA2;

}

public Byte getTemprVhodTedSecB1() {

return temprVhodTedSecB1;

}

public void setTemprVhodTedSecB1(Byte temprVhodTedSecB1) {

this.temprVhodTedSecB1 = temprVhodTedSecB1;

}

public Byte getTemprVhodTedSecB2() {

return temprVhodTedSecB2;

}

public void setTemprVhodTedSecB2(Byte temprVhodTedSecB2) {

this.temprVhodTedSecB2 = temprVhodTedSecB2;

}

public Byte getTemprVihodTed1SecA() {

return temprVihodTed1SecA;

}

public void setTemprVihodTed1SecA(Byte temprVihodTed1SecA) {

this.temprVihodTed1SecA = temprVihodTed1SecA;

}

public Byte getTemprVihodTed2SecA() {

return temprVihodTed2SecA;

}

public void setTemprVihodTed2SecA(Byte temprVihodTed2SecA) {

this.temprVihodTed2SecA = temprVihodTed2SecA;

}

public Byte getTemprVihodTed3SecA() {

return temprVihodTed3SecA;

}

public void setTemprVihodTed3SecA(Byte temprVihodTed3SecA) {

this.temprVihodTed3SecA = temprVihodTed3SecA;

}

public Byte getTemprVihodTed4SecA() {

return temprVihodTed4SecA;

}

public void setTemprVihodTed4SecA(Byte temprVihodTed4SecA) {

this.temprVihodTed4SecA = temprVihodTed4SecA;

}

public Byte getTemprVihodTed1SecB() {

return temprVihodTed1SecB;

}

public void setTemprVihodTed1SecB(Byte temprVihodTed1SecB) {

this.temprVihodTed1SecB = temprVihodTed1SecB;

}

public Byte getTemprVihodTed2SecB() {

return temprVihodTed2SecB;

}

public void setTemprVihodTed2SecB(Byte temprVihodTed2SecB) {

this.temprVihodTed2SecB = temprVihodTed2SecB;

}

public Byte getTemprVihodTed3SecB() {

return temprVihodTed3SecB;

}

public void setTemprVihodTed3SecB(Byte temprVihodTed3SecB) {

this.temprVihodTed3SecB = temprVihodTed3SecB;

}

public Byte getTemprVihodTed4SecB() {

return temprVihodTed4SecB;

}

public void setTemprVihodTed4SecB(Byte temprVihodTed4SecB) {

this.temprVihodTed4SecB = temprVihodTed4SecB;

}

public Short getTemprBptr1SecA() {

return temprBptr1SecA;

}

public void setTemprBptr1SecA(Short temprBptr1SecA) {

this.temprBptr1SecA = temprBptr1SecA;

}

public Short getTemprBptr2SecA() {

return temprBptr2SecA;

}

public void setTemprBptr2SecA(Short temprBptr2SecA) {

this.temprBptr2SecA = temprBptr2SecA;

}

public Short getTemprBptr1SecB() {

return temprBptr1SecB;

}

public void setTemprBptr1SecB(Short temprBptr1SecB) {

this.temprBptr1SecB = temprBptr1SecB;

}

public Short getTemprBptr2SecB() {

return temprBptr2SecB;

}

public void setTemprBptr2SecB(Short temprBptr2SecB) {

this.temprBptr2SecB = temprBptr2SecB;

}

@Override

public boolean equals(Object o) {

if (this == o) {

return true;

}

if (o == null || getClass() != o.getClass()) {

return false;

}

LocoDataEntity that = (LocoDataEntity) o;

if (idLocoData != that.idLocoData) {

return false;

}

if (idLoco != null ? !idLoco.equals(that.idLoco) : that.idLoco != null) {

return false;

}

if (naprAccumBatrSecA != null ? !naprAccumBatrSecA.equals(that.naprAccumBatrSecA) : that.naprAccumBatrSecA != null) {

return false;

}

if (naprAccumBatrSecB != null ? !naprAccumBatrSecB.equals(that.naprAccumBatrSecB) : that.naprAccumBatrSecB != null) {

return false;

}

if (naprContSetSecA != null ? !naprContSetSecA.equals(that.naprContSetSecA) : that.naprContSetSecA != null) {

return false;

}

if (naprContSetSecB != null ? !naprContSetSecB.equals(that.naprContSetSecB) : that.naprContSetSecB != null) {

return false;

}

if (recordTimeLocoData != null ? !recordTimeLocoData.equals(that.recordTimeLocoData) : that.recordTimeLocoData != null) {

return false;

}

if (temprBptr1SecA != null ? !temprBptr1SecA.equals(that.temprBptr1SecA) : that.temprBptr1SecA != null) {

return false;

}

if (temprBptr1SecB != null ? !temprBptr1SecB.equals(that.temprBptr1SecB) : that.temprBptr1SecB != null) {

return false;

}

if (temprBptr2SecA != null ? !temprBptr2SecA.equals(that.temprBptr2SecA) : that.temprBptr2SecA != null) {

return false;

}

if (temprBptr2SecB != null ? !temprBptr2SecB.equals(that.temprBptr2SecB) : that.temprBptr2SecB != null) {

return false;

}

if (temprBuks1StorMashSecA != null ? !temprBuks1StorMashSecA.equals(that.temprBuks1StorMashSecA) : that.temprBuks1StorMashSecA != null) {

return false;

}

if (temprBuks1StorMashSecB != null ? !temprBuks1StorMashSecB.equals(that.temprBuks1StorMashSecB) : that.temprBuks1StorMashSecB != null) {

return false;

}

if (temprBuks1StorPomMashSecA != null ? !temprBuks1StorPomMashSecA.equals(that.temprBuks1StorPomMashSecA) : that.temprBuks1StorPomMashSecA != null) {

return false;

}

if (temprBuks1StorPomMashSecB != null ? !temprBuks1StorPomMashSecB.equals(that.temprBuks1StorPomMashSecB) : that.temprBuks1StorPomMashSecB != null) {

return false;

}

if (temprBuks2StorMashSecA != null ? !temprBuks2StorMashSecA.equals(that.temprBuks2StorMashSecA) : that.temprBuks2StorMashSecA != null) {

return false;

}

if (temprBuks2StorMashSecB != null ? !temprBuks2StorMashSecB.equals(that.temprBuks2StorMashSecB) : that.temprBuks2StorMashSecB != null) {

return false;

}

if (temprBuks2StorPomMashSecA != null ? !temprBuks2StorPomMashSecA.equals(that.temprBuks2StorPomMashSecA) : that.temprBuks2StorPomMashSecA != null) {

return false;

}

if (temprBuks2StorPomMashSecB != null ? !temprBuks2StorPomMashSecB.equals(that.temprBuks2StorPomMashSecB) : that.temprBuks2StorPomMashSecB != null) {

return false;

}

if (temprBuks3StorMashSecA != null ? !temprBuks3StorMashSecA.equals(that.temprBuks3StorMashSecA) : that.temprBuks3StorMashSecA != null) {

return false;

}

if (temprBuks3StorMashSecB != null ? !temprBuks3StorMashSecB.equals(that.temprBuks3StorMashSecB) : that.temprBuks3StorMashSecB != null) {

return false;

}

if (temprBuks3StorPomMashSecA != null ? !temprBuks3StorPomMashSecA.equals(that.temprBuks3StorPomMashSecA) : that.temprBuks3StorPomMashSecA != null) {

return false;

}

if (temprBuks3StorPomMashSecB != null ? !temprBuks3StorPomMashSecB.equals(that.temprBuks3StorPomMashSecB) : that.temprBuks3StorPomMashSecB != null) {

return false;

}

if (temprBuks4StorMashSecA != null ? !temprBuks4StorMashSecA.equals(that.temprBuks4StorMashSecA) : that.temprBuks4StorMashSecA != null) {

return false;

}

if (temprBuks4StorMashSecB != null ? !temprBuks4StorMashSecB.equals(that.temprBuks4StorMashSecB) : that.temprBuks4StorMashSecB != null) {

return false;

}

if (temprBuks4StorPomMashSecA != null ? !temprBuks4StorPomMashSecA.equals(that.temprBuks4StorPomMashSecA) : that.temprBuks4StorPomMashSecA != null) {

return false;

}

if (temprBuks4StorPomMashSecB != null ? !temprBuks4StorPomMashSecB.equals(that.temprBuks4StorPomMashSecB) : that.temprBuks4StorPomMashSecB != null) {

return false;

}

if (temprVhodTedSecA1 != null ? !temprVhodTedSecA1.equals(that.temprVhodTedSecA1) : that.temprVhodTedSecA1 != null) {

return false;

}

if (temprVhodTedSecA2 != null ? !temprVhodTedSecA2.equals(that.temprVhodTedSecA2) : that.temprVhodTedSecA2 != null) {

return false;

}

if (temprVhodTedSecB1 != null ? !temprVhodTedSecB1.equals(that.temprVhodTedSecB1) : that.temprVhodTedSecB1 != null) {

return false;

}

if (temprVhodTedSecB2 != null ? !temprVhodTedSecB2.equals(that.temprVhodTedSecB2) : that.temprVhodTedSecB2 != null) {

return false;

}

if (temprVihodTed1SecA != null ? !temprVihodTed1SecA.equals(that.temprVihodTed1SecA) : that.temprVihodTed1SecA != null) {

return false;

}

if (temprVihodTed1SecB != null ? !temprVihodTed1SecB.equals(that.temprVihodTed1SecB) : that.temprVihodTed1SecB != null) {

return false;

}

if (temprVihodTed2SecA != null ? !temprVihodTed2SecA.equals(that.temprVihodTed2SecA) : that.temprVihodTed2SecA != null) {

return false;

}

if (temprVihodTed2SecB != null ? !temprVihodTed2SecB.equals(that.temprVihodTed2SecB) : that.temprVihodTed2SecB != null) {

return false;

}

if (temprVihodTed3SecA != null ? !temprVihodTed3SecA.equals(that.temprVihodTed3SecA) : that.temprVihodTed3SecA != null) {

return false;

}

if (temprVihodTed3SecB != null ? !temprVihodTed3SecB.equals(that.temprVihodTed3SecB) : that.temprVihodTed3SecB != null) {

return false;

}

if (temprVihodTed4SecA != null ? !temprVihodTed4SecA.equals(that.temprVihodTed4SecA) : that.temprVihodTed4SecA != null) {

return false;

}

if (temprVihodTed4SecB != null ? !temprVihodTed4SecB.equals(that.temprVihodTed4SecB) : that.temprVihodTed4SecB != null) {

return false;

}

if (tokAccumBatrSecA != null ? !tokAccumBatrSecA.equals(that.tokAccumBatrSecA) : that.tokAccumBatrSecA != null) {

return false;

}

if (tokAccumBatrSecB != null ? !tokAccumBatrSecB.equals(that.tokAccumBatrSecB) : that.tokAccumBatrSecB != null) {

return false;

}

if (tokVozTedSecA != null ? !tokVozTedSecA.equals(that.tokVozTedSecA) : that.tokVozTedSecA != null) {

return false;

}

if (tokVozTedSecB != null ? !tokVozTedSecB.equals(that.tokVozTedSecB) : that.tokVozTedSecB != null) {

return false;

}

if (tokYakTed12SecA != null ? !tokYakTed12SecA.equals(that.tokYakTed12SecA) : that.tokYakTed12SecA != null) {

return false;

}

if (tokYakTed12SecB != null ? !tokYakTed12SecB.equals(that.tokYakTed12SecB) : that.tokYakTed12SecB != null) {

return false;

}

if (tokYakTed34SecA != null ? !tokYakTed34SecA.equals(that.tokYakTed34SecA) : that.tokYakTed34SecA != null) {

return false;

}

if (tokYakTed34SecB != null ? !tokYakTed34SecB.equals(that.tokYakTed34SecB) : that.tokYakTed34SecB != null) {

return false;

}

return true;

}

@Override

public int hashCode() {

int result = (int) (idLocoData ^ (idLocoData >>> 32));

result = 31 \* result + (idLoco != null ? idLoco.hashCode() : 0);

result = 31 \* result + (recordTimeLocoData != null ? recordTimeLocoData.hashCode() : 0);

result = 31 \* result + (naprAccumBatrSecA != null ? naprAccumBatrSecA.hashCode() : 0);

result = 31 \* result + (naprAccumBatrSecB != null ? naprAccumBatrSecB.hashCode() : 0);

result = 31 \* result + (tokAccumBatrSecA != null ? tokAccumBatrSecA.hashCode() : 0);

result = 31 \* result + (tokAccumBatrSecB != null ? tokAccumBatrSecB.hashCode() : 0);

result = 31 \* result + (tokVozTedSecA != null ? tokVozTedSecA.hashCode() : 0);

result = 31 \* result + (tokVozTedSecB != null ? tokVozTedSecB.hashCode() : 0);

result = 31 \* result + (tokYakTed12SecA != null ? tokYakTed12SecA.hashCode() : 0);

result = 31 \* result + (tokYakTed12SecB != null ? tokYakTed12SecB.hashCode() : 0);

result = 31 \* result + (tokYakTed34SecA != null ? tokYakTed34SecA.hashCode() : 0);

result = 31 \* result + (tokYakTed34SecB != null ? tokYakTed34SecB.hashCode() : 0);

result = 31 \* result + (naprContSetSecA != null ? naprContSetSecA.hashCode() : 0);

result = 31 \* result + (naprContSetSecB != null ? naprContSetSecB.hashCode() : 0);

result = 31 \* result + (temprBuks1StorMashSecA != null ? temprBuks1StorMashSecA.hashCode() : 0);

result = 31 \* result + (temprBuks1StorPomMashSecA != null ? temprBuks1StorPomMashSecA.hashCode() : 0);

result = 31 \* result + (temprBuks2StorMashSecA != null ? temprBuks2StorMashSecA.hashCode() : 0);

result = 31 \* result + (temprBuks2StorPomMashSecA != null ? temprBuks2StorPomMashSecA.hashCode() : 0);

result = 31 \* result + (temprBuks3StorMashSecA != null ? temprBuks3StorMashSecA.hashCode() : 0);

result = 31 \* result + (temprBuks3StorPomMashSecA != null ? temprBuks3StorPomMashSecA.hashCode() : 0);

result = 31 \* result + (temprBuks4StorMashSecA != null ? temprBuks4StorMashSecA.hashCode() : 0);

result = 31 \* result + (temprBuks4StorPomMashSecA != null ? temprBuks4StorPomMashSecA.hashCode() : 0);

result = 31 \* result + (temprBuks4StorPomMashSecB != null ? temprBuks4StorPomMashSecB.hashCode() : 0);

result = 31 \* result + (temprBuks4StorMashSecB != null ? temprBuks4StorMashSecB.hashCode() : 0);

result = 31 \* result + (temprBuks3StorPomMashSecB != null ? temprBuks3StorPomMashSecB.hashCode() : 0);

result = 31 \* result + (temprBuks3StorMashSecB != null ? temprBuks3StorMashSecB.hashCode() : 0);

result = 31 \* result + (temprBuks2StorPomMashSecB != null ? temprBuks2StorPomMashSecB.hashCode() : 0);

result = 31 \* result + (temprBuks2StorMashSecB != null ? temprBuks2StorMashSecB.hashCode() : 0);

result = 31 \* result + (temprBuks1StorPomMashSecB != null ? temprBuks1StorPomMashSecB.hashCode() : 0);

result = 31 \* result + (temprBuks1StorMashSecB != null ? temprBuks1StorMashSecB.hashCode() : 0);

result = 31 \* result + (temprVhodTedSecA1 != null ? temprVhodTedSecA1.hashCode() : 0);

result = 31 \* result + (temprVhodTedSecA2 != null ? temprVhodTedSecA2.hashCode() : 0);

result = 31 \* result + (temprVhodTedSecB1 != null ? temprVhodTedSecB1.hashCode() : 0);

result = 31 \* result + (temprVhodTedSecB2 != null ? temprVhodTedSecB2.hashCode() : 0);

result = 31 \* result + (temprVihodTed1SecA != null ? temprVihodTed1SecA.hashCode() : 0);

result = 31 \* result + (temprVihodTed2SecA != null ? temprVihodTed2SecA.hashCode() : 0);

result = 31 \* result + (temprVihodTed3SecA != null ? temprVihodTed3SecA.hashCode() : 0);

result = 31 \* result + (temprVihodTed4SecA != null ? temprVihodTed4SecA.hashCode() : 0);

result = 31 \* result + (temprVihodTed1SecB != null ? temprVihodTed1SecB.hashCode() : 0);

result = 31 \* result + (temprVihodTed2SecB != null ? temprVihodTed2SecB.hashCode() : 0);

result = 31 \* result + (temprVihodTed3SecB != null ? temprVihodTed3SecB.hashCode() : 0);

result = 31 \* result + (temprVihodTed4SecB != null ? temprVihodTed4SecB.hashCode() : 0);

result = 31 \* result + (temprBptr1SecA != null ? temprBptr1SecA.hashCode() : 0);

result = 31 \* result + (temprBptr2SecA != null ? temprBptr2SecA.hashCode() : 0);

result = 31 \* result + (temprBptr1SecB != null ? temprBptr1SecB.hashCode() : 0);

result = 31 \* result + (temprBptr2SecB != null ? temprBptr2SecB.hashCode() : 0);

return result;

}

}

## Текст програми LocoEntity.java

package com.locostatmanager.busines.dao.entities;

/\*\*

\* Created by anatoliy on 22.04.14.

\*/

public class LocoEntity {

private String idLoco;

private String titleLoco;

public String getIdLoco() {

return idLoco;

}

public void setIdLoco(String idLoco) {

this.idLoco = idLoco;

}

public String getTitleLoco() {

return titleLoco;

}

public void setTitleLoco(String titleLoco) {

this.titleLoco = titleLoco;

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

LocoEntity that = (LocoEntity) o;

if (idLoco != null ? !idLoco.equals(that.idLoco) : that.idLoco != null) return false;

if (titleLoco != null ? !titleLoco.equals(that.titleLoco) : that.titleLoco != null) return false;

return true;

}

@Override

public int hashCode() {

int result = idLoco != null ? idLoco.hashCode() : 0;

result = 31 \* result + (titleLoco != null ? titleLoco.hashCode() : 0);

return result;

}

}

## Текст програми RecordStructureInfo.java

package com.locostatmanager.busines.dao.entities;

/\*\*

\*

\* @author dmitry

\*/

public class RecordStructureInfo {

private String name;

private Integer index;

private Double delta;

private String fsiType;

public String getFsiType() {

return fsiType;

}

public void setFsiType(String fsiType) {

this.fsiType = fsiType;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Integer getIndex() {

return index;

}

public void setIndex(Integer index) {

this.index = index;

}

public Double getDelta() {

return delta;

}

public void setDelta(Double delta) {

this.delta = delta;

}

}

## Текст програми SensorEntity.java

package com.locostatmanager.busines.dao.entities;

/\*\*

\* Created by anatoliy on 22.04.14.

\*/

public class SensorEntity {

private String name;

private String description;

private String unitOfMeasure;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public String getUnitOfMeasure() {

return unitOfMeasure;

}

public void setUnitOfMeasure(String unitOfMeasure) {

this.unitOfMeasure = unitOfMeasure;

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

SensorEntity that = (SensorEntity) o;

if (description != null ? !description.equals(that.description) : that.description != null) return false;

if (name != null ? !name.equals(that.name) : that.name != null) return false;

if (unitOfMeasure != null ? !unitOfMeasure.equals(that.unitOfMeasure) : that.unitOfMeasure != null)

return false;

return true;

}

@Override

public int hashCode() {

int result = name != null ? name.hashCode() : 0;

result = 31 \* result + (description != null ? description.hashCode() : 0);

result = 31 \* result + (unitOfMeasure != null ? unitOfMeasure.hashCode() : 0);

return result;

}

@Override

public String toString() {

return "SensorEntity{" +

"name='" + name + '\'' +

", description='" + description + '\'' +

", unitOfMeasure='" + unitOfMeasure + '\'' +

'}';

}

}

## Текст програми UserEntity.java

package com.locostatmanager.busines.dao.entities;

import org.hibernate.validator.constraints.NotEmpty;

/\*\*

\* Created by anatoliy on 01.06.14.

\*/

public class UserEntity {

@NotEmpty

private String login;

@NotEmpty

private String password;

@NotEmpty

private String name;

@NotEmpty

private String role;

public String getLogin() {

return login;

}

public void setLogin(String login) {

this.login = login;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

UserEntity that = (UserEntity) o;

if (login != null ? !login.equals(that.login) : that.login != null) return false;

if (name != null ? !name.equals(that.name) : that.name != null) return false;

if (password != null ? !password.equals(that.password) : that.password != null) return false;

if (role != null ? !role.equals(that.role) : that.role != null) return false;

return true;

}

@Override

public int hashCode() {

int result = login != null ? login.hashCode() : 0;

result = 31 \* result + (password != null ? password.hashCode() : 0);

result = 31 \* result + (name != null ? name.hashCode() : 0);

result = 31 \* result + (role != null ? role.hashCode() : 0);

return result;

}

@Override

public String toString() {

return "UserEntity{" +

"login='" + login + '\'' +

", password='" + password + '\'' +

", name='" + name + '\'' +

", role='" + role + '\'' +

'}';

}

}

## Текст програми CustomException.java

package com.locostatmanager.busines.exceptions;

/\*\*

\*

\* @author dmitry

\*/

public class CustomException extends Exception {

protected String type;

public CustomException(String type, Throwable cause) {

super(cause);

this.type = type;

}

}

## Текст програми DataAccessException.java

package com.locostatmanager.busines.exceptions;

/\*\*

\*

\* @author dmitry

\*/

public class DataAccessException extends CustomException {

public DataAccessException(Throwable cause) {

super("dataAccess", cause);

}

public DataAccessException(String message) {

super("dataAccess", new Exception(message));

}

}

## Текст програми ValidationException.java

package com.locostatmanager.busines.exceptions;

/\*\*

\*

\* @author dmitry

\*/

public class ValidationException extends CustomException {

public ValidationException(Throwable cause) {

super("validation", cause);

}

public ValidationException(String message) {

super("validation", new Exception(message));

}

}

## Текст програми DataLoadingServiceImpl.java

package com.locostatmanager.busines.service;

import com.locostatmanager.busines.dao.DataDao;

import com.locostatmanager.busines.dao.FileStructureInfoDao;

import com.locostatmanager.busines.dao.RecordStructureInfoDao;

import com.locostatmanager.busines.dao.entities.FileStructureInfo;

import com.locostatmanager.busines.dao.entities.LocoDataEntity;

import com.locostatmanager.busines.dao.entities.RecordStructureInfo;

import java.io.IOException;

import java.sql.Timestamp;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Calendar;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Random;

import org.codehaus.jackson.map.ObjectMapper;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

/\*\*

\*

\* @author dmitry

\*/

@Service

public class DataLoadingServiceImpl implements DataLoadingService {

@Autowired

private FileStructureInfoDao fileStructureInfoDao;

@Autowired

private RecordStructureInfoDao recordStructureInfoDao;

@Autowired

private DataDao dataDao;

@Override

@Transactional

public void loadData(byte[] file, String type) throws Exception {

FileStructureInfo fileStructureInfo = fileStructureInfoDao.get(type);

List<byte[]> recors = getRecordsFromFile(file, fileStructureInfo);

List<RecordStructureInfo> recordsInfo = recordStructureInfoDao.getByFsiType(fileStructureInfo.getType());

List<Map<String, Double>> values = getValuesFromRecords(recordsInfo, recors);

List<LocoDataEntity> entitys = createEntities(values);

long i = 10;

for (LocoDataEntity entity : entitys) {

try {

dataDao.add(entity);

} catch (Exception e) {

e.printStackTrace();

}

}

}

private List<byte[]> getRecordsFromFile(byte[] file, FileStructureInfo fileStructureInfo) {

List<byte[]> records = new ArrayList<byte[]>();

int cursor = fileStructureInfo.getRecordsStartIndex();

while (cursor < file.length) {

byte[] record = Arrays.copyOfRange(file, cursor, cursor + fileStructureInfo.getRecordLength());

records.add(record);

cursor += fileStructureInfo.getRecordLength();

}

return records;

}

private List<Map<String, Double>> getValuesFromRecords(List<RecordStructureInfo> recordsInfo, List<byte[]> records) {

List<Map<String, Double>> values = new ArrayList<Map<String, Double>>();

for (byte[] record : records) {

Map<String, Double> value = new HashMap<String, Double>();

for (RecordStructureInfo recordStructureInfo : recordsInfo) {

Byte val = record[recordStructureInfo.getIndex()];

value.put(recordStructureInfo.getName().replace(" ", ""), val.doubleValue() \* recordStructureInfo.getDelta());

values.add(value);

}

}

return values;

}

private List<LocoDataEntity> createEntities(List<Map<String, Double>> values) throws IOException {

List<LocoDataEntity> entitys = new ArrayList<LocoDataEntity>();

for (Map<String, Double> value : values) {

ObjectMapper mapper = new ObjectMapper();

String jsonEntity = mapper.writeValueAsString(value);

LocoDataEntity entity = mapper.readValue(jsonEntity, LocoDataEntity.class);

entity.setIdLoco("L001dp");

entity.setRecordTimeLocoData(new Timestamp(Calendar.getInstance().getTimeInMillis()));

entitys.add(entity);

}

return entitys;

}

}

## Текст програми FileCheckInServiceImpl.java

package com.locostatmanager.busines.service;

import com.locostatmanager.busines.dao.FileInfoDao;

import com.locostatmanager.busines.dao.entities.FileInfo;

import com.locostatmanager.busines.exceptions.DataAccessException;

import com.locostatmanager.busines.exceptions.ValidationException;

import java.sql.Timestamp;

import java.util.Calendar;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

/\*\*

\*

\* @author dmitry

\*/

@Repository

public class FileCheckInServiceImpl implements FileCheckInService {

@Autowired

private FileInfoDao fileInfoDao;

@Transactional

public void checkIn(String fileName, Integer size) throws DataAccessException, ValidationException {

FileInfo fileInfo = fileInfoDao.get(fileName, size);

if (null != fileInfo) {

throw new ValidationException("Data from file with name " + fileName + " was alrwady loaded at:" + fileInfo.getLoadingDate());

}else {

fileInfo = new FileInfo();

fileInfo.setFileName(fileName);

fileInfo.setFileSize(size);

fileInfo.setLoadingDate(new Timestamp(Calendar.getInstance().getTimeInMillis()));

fileInfoDao.add(fileInfo);

}

}

}

## Текст програми LocomotiveServiceImpl.java

package com.locostatmanager.busines.service;

import com.locostatmanager.busines.dao.LocomotiveDao;

import com.locostatmanager.busines.dao.entities.LocoEntity;

import com.locostatmanager.busines.exceptions.DataAccessException;

import com.locostatmanager.busines.exceptions.ValidationException;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

/\*\*

\*

\* @author dmitry

\*/

@Service

public class LocomotiveServiceImpl implements LocomotiveService {

@Autowired

private LocomotiveDao locomotiveDao;

@Transactional(propagation = Propagation.REQUIRES\_NEW)

public void add(LocoEntity entity) throws ValidationException, DataAccessException {

locomotiveDao.add(entity);

}

@Transactional

public List<LocoEntity> getAll() throws ValidationException, DataAccessException {

return locomotiveDao.getAll();

}

@Transactional

public LocoEntity getById(String id) throws ValidationException, DataAccessException {

return locomotiveDao.getById(id);

}

@Transactional

public LocoEntity getByTitle(String title) throws ValidationException, DataAccessException {

return locomotiveDao.getByTitle(title);

}

@Transactional

public String getCount() throws ValidationException, DataAccessException {

return locomotiveDao.getCount();

}

@Transactional

public void add(String id, String title) throws ValidationException, DataAccessException {

if (null == id || "".equals(id)) {

throw new ValidationException("incorrect id");

} else if (null == title || "".equals(title)) {

throw new ValidationException("incorrect title");

}

LocoEntity entity = new LocoEntity();

entity.setIdLoco(id);

entity.setTitleLoco(title);

locomotiveDao.add(entity);

}

@Transactional

public void delete(String id) throws ValidationException, DataAccessException {

if (null == id || "".equals(id)) {

throw new ValidationException("incorrect id");

}

locomotiveDao.delete(id);

}

}

## Текст програми SensorServiceImpl.java

package com.locostatmanager.busines.service;

import com.locostatmanager.busines.constants.Units;

import com.locostatmanager.busines.dao.SensorDao;

import com.locostatmanager.busines.dao.entities.SensorEntity;

import com.locostatmanager.busines.exceptions.DataAccessException;

import com.locostatmanager.busines.exceptions.ValidationException;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

/\*\*

\*

\* @author dmitry

\*/

@Service

public class SensorServiceImpl implements SensorService {

@Autowired

private SensorDao sensorDao;

public void add(SensorEntity entity) throws DataAccessException, ValidationException {

sensorDao.add(entity);

}

public List<SensorEntity> getAll() throws DataAccessException, ValidationException {

return sensorDao.getAll();

}

public List<SensorEntity> getBuks() throws DataAccessException, ValidationException {

return sensorDao.getByUnitName(Units.BUKS.toString());

}

public List<SensorEntity> getOsnovn() throws DataAccessException, ValidationException {

return sensorDao.getByUnitName(Units.OSN.toString());

}

public List<SensorEntity> getPTres() throws DataAccessException, ValidationException {

return sensorDao.getByUnitName(Units.PTRES.toString());

}

public List<SensorEntity> getTDvig() throws DataAccessException, ValidationException {

return sensorDao.getByUnitName(Units.TDVIG.toString());

}

@Transactional

public SensorEntity getByName(String name) throws DataAccessException, ValidationException {

if (null == name || "".equals(name)) {

throw new ValidationException("Incorrect parametr");

}

return sensorDao.getByName(name);

}

public String getCount() throws DataAccessException, ValidationException {

return sensorDao.getCount();

}

}

## Текст програми StatisticServiceImpl.java

package com.locostatmanager.busines.service;

import com.locostatmanager.busines.dao.DataDao;

import com.locostatmanager.busines.dao.entities.LocoDataEntity;

import com.locostatmanager.busines.dao.entities.LocoEntity;

import com.locostatmanager.busines.exceptions.DataAccessException;

import com.locostatmanager.busines.exceptions.ValidationException;

import com.locostatmanager.busines.message.LocomotiveStatistic;

import java.sql.Time;

import java.sql.Timestamp;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

/\*\*

\*

\* @author dmitry

\*/

@Service

public class StatisticServiceImpl implements StatisticService {

@Autowired

private DataDao dataDao;

@Autowired

private LocomotiveService locomotiveService;

@Transactional(propagation = Propagation.REQUIRES\_NEW)

public void add(LocoDataEntity entity) throws DataAccessException, ValidationException {

dataDao.add(entity);

}

@Transactional

public List<LocoDataEntity> getAll() throws DataAccessException, ValidationException {

return dataDao.getAll();

}

@Transactional

public List<LocoDataEntity> getAfter(Timestamp date) throws DataAccessException, ValidationException {

return dataDao.getAfter(date);

}

@Transactional

public List<LocoDataEntity> getBefore(Timestamp date) throws DataAccessException, ValidationException {

return dataDao.getBefore(date);

}

@Transactional

public List<LocoDataEntity> getBetween(Timestamp startDate, Timestamp endDate) throws DataAccessException, ValidationException {

return dataDao.getBetween(startDate, endDate);

}

@Transactional

public List<LocoDataEntity> getAfter(Timestamp date, LocoEntity entity) throws DataAccessException, ValidationException {

return dataDao.getAfter(date, entity);

}

@Transactional

public List<LocoDataEntity> getBefore(Timestamp date, LocoEntity entity) throws DataAccessException, ValidationException {

return dataDao.getBefore(date, entity);

}

@Transactional

public List<LocoDataEntity> getByLocomotive(LocoEntity entity) throws DataAccessException, ValidationException {

return dataDao.getByLocomotive(entity);

}

@Transactional

public LocoDataEntity getById(String id) throws DataAccessException, ValidationException {

return dataDao.getById(id);

}

public List<LocoDataEntity> getBetween(String startDate, String endDate, String locomotiveId) throws DataAccessException, ValidationException {

Timestamp start = null;

Timestamp end = null;

try {

SimpleDateFormat dateFormat = new SimpleDateFormat("dd.MM.yyyy hh:mm");

start = new Timestamp(dateFormat.parse(startDate).getTime());

end = new Timestamp(dateFormat.parse(endDate).getTime());

} catch (ParseException e) {

throw new ValidationException(e);

}

if (null == locomotiveId || "".equals(locomotiveId)) {

throw new ValidationException(new Exception("not valid locomotive id"));

}

return dataDao.getBetween(start, end, locomotiveId);

}

public int getCount() throws DataAccessException, ValidationException {

return dataDao.getCount();

}

@Transactional

public List<LocomotiveStatistic> getLocomotivesRatio() throws ValidationException, DataAccessException {

List<LocomotiveStatistic> statistics = new ArrayList<LocomotiveStatistic>();

List<LocoEntity> list = locomotiveService.getAll();

for (LocoEntity entity : list) {

LocomotiveStatistic ls = new LocomotiveStatistic();

ls.setLocoName(entity.getIdLoco());

ls.setLocoPortion(dataDao.getRecordsCount(entity.getIdLoco()));

statistics.add(ls);

}

return statistics;

}

@Transactional

public List<LocomotiveStatistic> getLocomotivesPercentage() throws ValidationException, DataAccessException {

List<LocomotiveStatistic> statistics = new ArrayList<LocomotiveStatistic>();

List<LocoEntity> list = locomotiveService.getAll();

int allRecordsCount = dataDao.getCount();

for (LocoEntity entity : list) {

LocomotiveStatistic ls = new LocomotiveStatistic();

ls.setLocoName(entity.getIdLoco());

ls.setLocoPortion( 100.00 \* dataDao.getRecordsCount(entity.getIdLoco()) / allRecordsCount);

statistics.add(ls);

}

return statistics;

}

}