

Лабораторная работа №3

Тема: «Методология объектно-ориентированного моделирования. Этап создания физической модели»

1. Цель работы:

Ознакомление с основными элементами определения, представления, проектирования и моделирования программных систем с помощью языка UML.

Задание:

1. Рассмотреть материал по объектно-ориентированному моделированию (Приложение 1 (лаб. раб. 2 и материалы лекций))
2. Определиться с диаграммами из семейства UML моделей на этапе создания физической модели автоматизированной системы (тематика выбрана в процессе лабораторной работы 1).
3. Разработать UML диаграммы этапа создания физической модели.
4. Задokumentировать прецеденты.

Типичное описание должно содержать следующие разделы.

- a. Краткое описание.
 - b. Участвующие субъекты.
 - c. Предусловия, необходимые для инициирования прецедента.
 - d. Детализированное описание потока событий, которое включает: основной поток, который можно разбить для того, чтобы показать подчиненные потоки событий (подчиненные потоки могут быть разделены дальше на еще более мелкие потоки, с целью сделать читаемость документа более удобной); альтернативные потоки для определения исключительных ситуаций.
 - e. Постусловия, определяющие состояние системы, по достижении которых прецедент завершается (**Пример в Приложении 1**).
5. Записать не менее 10 требований согласно синтаксиса требований: [обстоятельства] [субъект] [действие] [объект] [ограничение]

Пример: Когда сигнал получен [обстоятельства] система [субъект] должна установить [действие] разряд сигнала [объект] в течение двух секунд [ограничение].

6. Сформулировать нефункциональные требования.
7. Разработать алгоритм обработки данных (см. пример оформления кода с пояснениями в архиве **pikpo3_python.zip** или **pikpo3_java.zip**).
 - a. Алгоритм обработки должен быть реализован с помощью объектно-ориентированного подхода. Обработчик данных реализуется в отдельном классе (**DataProcessor**), который имеет 3 базовых метода:

чтение источника данных (**read**), запуск обработки данных (**run**), вывод результата на экран (**print_result**).

- b. Реализовать в классе необходимые методы обработки (например, очистка, назначение категорий и т.п.).
 - c. Использовать паттерн «Фабрика» (Factory) для вызова различных экземпляров DataProcessor в зависимости от типа входного набора данных (например, csv-файл, txt-файл).
 - d. Загрузить код на **GitHub** (см. «Памятка для работы с GIT»).
 - e. Разработать UML схему классов обработчика данных.
8. **Написать отчет.** Отчет должен включать комплекс статических и динамических моделей, описание алгоритма обработки данных, скриншоты с результатами выполнения алгоритма (фрагмент входного набора данных и вывод после выполнения обработки), скриншот GitHub-репозитория с загруженными файлами проекта.

Дополнительная информация:

ПРИМЕР реализации алгоритма обработки данных с помощью **Factory Pattern** находится в архивах:

pikpo3_python.zip – пример на **Python**

pikpo3_java.zip – пример на **Java** (задание повышенной сложности)

Для запуска примера на Python понадобится установка дополнительного пакета для работы с файлами наборов данных:

```
pip install pandas
```

Для запуска примера на Java необходимо установить сборщик пакетов **Maven** и среду **Liberica OpenJDK 8**. Установщик JDK можно скачать по ссылке:

<https://bell-sw.com/pages/downloads/#mn>

В процессе установки не забудьте установить системные переменные **PATH** и **JAVA_HOME**:

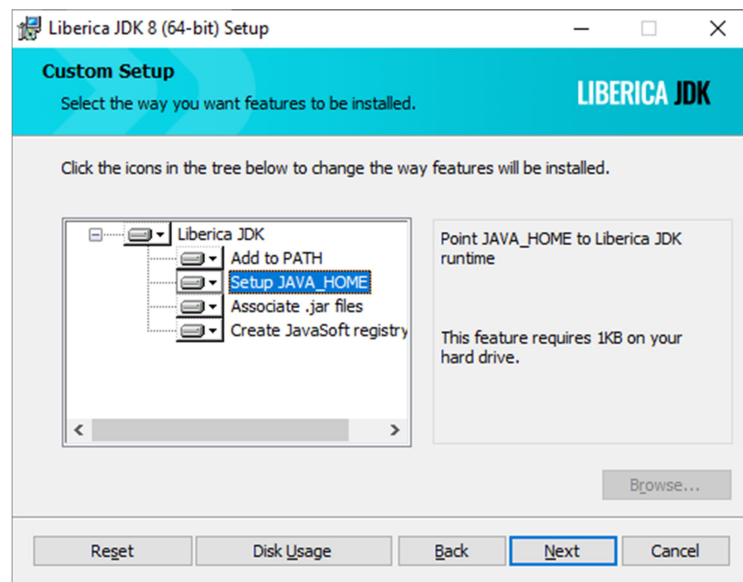


Рис. 1 – Установка переменных окружения ОС

Перед началом работы с проектом в IDE не забудьте установить соответствующую среду разработки Java (SDK) в настройках проекта (File -> Project Structure):

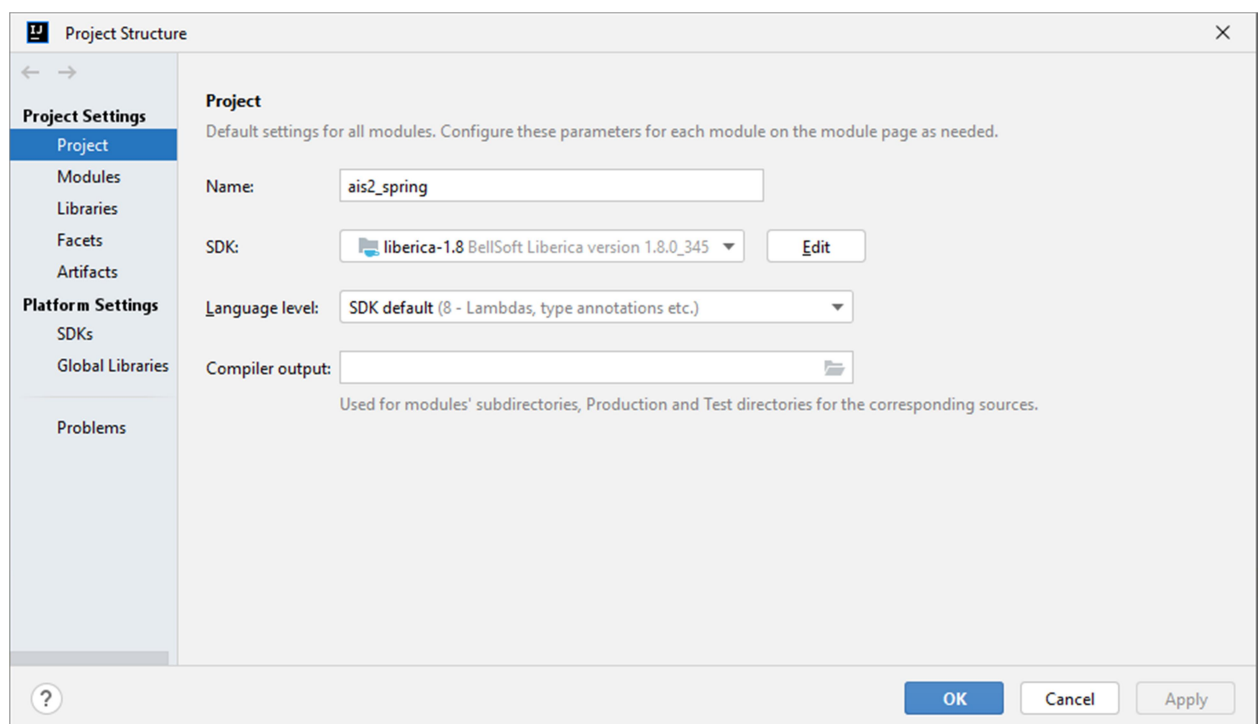


Рис. 2 – Установка Java SDK в IDE IDEA

Далее необходима установка сборщика зависимостей **Maven**:

<https://maven.apache.org/install.html>

Чтобы установить все указанные в **pom.xml** зависимости запустите команды **clean** и **install** в панели Maven и обновите проект:

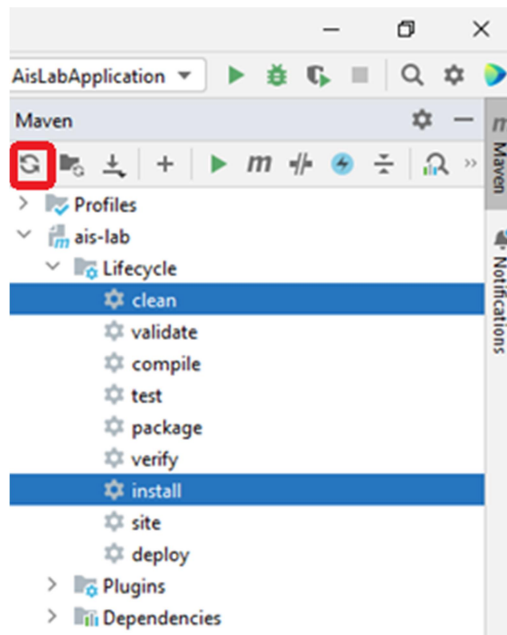


Рис. 3 – Панель Maven в IDEA IDE

Или введите команду в терминале IDE:

```
mvn clean install
```

Необходимое программное обеспечение:

<https://www.jetbrains.com/ru-ru/pycharm/download> – среда разработки на **Python** (PyCharm Community)

<https://www.jetbrains.com/ru-ru/idea/download> – среда разработки на **Java** (IDEA Community)

Реализация Factory Pattern:

<http://cpp-reference.ru/patterns/creational-patterns/factory-method/>

<https://refactoring.guru/ru/design-patterns/factory-method/python/example> (**Python**)

<https://refactoring.guru/ru/design-patterns/factory-method/java/example> (**Java**)

Инструменты для работы с файлами данных:

<https://pandas.pydata.org/docs/reference/frame.html> (DataFrame в **Python Pandas**)

https://pandas.pydata.org/docs/reference/general_functions.html (Основные методы для работы с DataFrame в **Python Pandas**)

<https://joinery.sh/v1.10/api/reference/joinery/DataFrame> (**Java Joinery**)

Требования к оформлению отчета:

Способ выполнения текста должен быть единым для всей работы. **Шрифт** – Times New Roman, кегль 14, **межстрочный интервал** – 1,5, **размеры полей**: левое – 30 мм; правое – 10 мм, верхнее – 20 мм; нижнее – 20 мм. Сокращения слов в тексте допускаются только

общепринятые. **Абзацный отступ (1,25)** должен быть одинаковым во всей работе. **Нумерация страниц** основного текста должна быть сквозной. Номер страницы на титульном листе не указывается. Сам номер располагается внизу по центру страницы или справа.

Приложение 1

Прецедент	Заклучение договора
Краткое описание	Данный прецедент необходим для регистрации нового абонента в сети.
Субъект	Оператор, клиент.
Предусловия	Оператору необходимо ознакомить с имеющимися операторами связи и выдать форму анкеты потенциальному абоненту.
Основной поток	<p>После выбора клиентом соответствующего оператора, он заполняет форму, после чего оператор проверяет правильность заполнения формы на бумажном носителе и вводит данные в систему следующим действием "Выбор оператора связи - Договор об оказании услуг связи".</p> <p>После чего в системе "Обслуживание абонентов" открывается форма по заключению абонента сети в системе.</p> <p>При этом сначала система спрашивает, кто будет регистрироваться: Физическое лицо или Юридическое, и только после этого выводится соответствующая регистрационная форма.</p> <p>Оператор вводит информацию об организации клиента, о контактном лице юридического лица, а также вводит номера счетов организации.</p> <p>Далее договор сохраняется. Производится сеанс связи с Сервером, в процессе которого эти данные передаются на сервер.</p>
Альтернативный поток	<p>В случае, если пользователь не ввел все поля, система выдает сообщение "Введите все поля", дает возможность пройти процесс регистрации снова при ошибке.</p> <p>Также оператор и имеет возможность отказа от регистрации абонента путем выбора соответствующей команды.</p>
Постусловия	После успешного завершения прецедента, клиент внесен в базу данных Абоненты сети на сервере.