

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Таныгин Максим Олегович  
Должность: и.о. декана факультета фундаментальной и прикладной информатики  
Дата подписания: 21.09.2023 13:09:47  
Уникальный программный ключ:  
65ab2aa0d384efe8480e6a4c688eddbc475e411a

**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ  
Проректор по учебной работе  
*О.Г. Локтионова*  
« 24 » 12



**ГЕНЕРАЦИЯ КОДА ПРИ КОНСТРУИРОВАНИИ ПРОГРАММНОГО  
ОБЕСПЕЧЕНИЯ**

Методические указания по выполнению лабораторной работы  
по дисциплине «Конструирование программного обеспечения»  
для студентов направления подготовки 09.03.04 «Программная  
инженерия»

Курск 2017

УДК 004.652

Составители: В.Г. Белов, Т.М. Белова

Рецензент

Кандидат технических наук, доцент кафедры программной инженерии ЮЗГУ И.Н. Ефремова

**Генерация кода при конструировании программного обеспечения:** методические указания по выполнению лабораторной работы по дисциплине "Конструирование программного обеспечения" для студентов направления подготовки 09.03.04 "Программная инженерия" / Юго-Зап. гос. ун-т; сост.: В.Г. Белов, Т.М. Белова, – Курск, 2017. – 26 с.: ил. 36.

Изложена последовательность действий при генерации кода на языке Java по диаграмме классов в среде Eclipse.

Материал предназначен для студентов направления подготовки 09.03.04 «Программная инженерия», а также будет полезен студентам всех направлений подготовки, изучающим технологии проектирования программно-информационных систем.

Текст печатается в авторской редакции.

Подписано в печать 24.12.17. Формат 60x84 1/16.  
Усл. печ. л. 14. Уч.-изд. л. 13 Тираж 100 экз. Заказ 4389. Бесплатно.

Юго-Западный государственный университет  
305040, Курск, ул.50 лет Октября, 94.

## Содержание

|  |    |
|--|----|
| 1 Цель лабораторной работы .....                 | 4  |
| 2 Порядок выполнения лабораторной работы .....   | 5  |
| 3 Содержание отчета по лабораторной работе ..... | 26 |
| 4 Вопросы к защите лабораторной работы .....     | 27 |
| 5 Индивидуальные задания .....                   | 28 |

## **1 Цель лабораторной работы**

Целью лабораторной работы является приобретение знаний, умений и навыков для использования инструмента проектирования UML-диаграмм Parugus в качестве преобразования диаграммы классов в Java-код в интегрированной среде разработки Eclipse.

Процесс преобразования UML-модели в код путем отображения на некоторый язык реализации является одним из основных этапов прямого проектирования программно-информационных систем.

## 2 Порядок выполнения лабораторной работы

1. Для моделирования UML-диаграмм в интегрированной среде разработки Eclipse необходимо установить инструмент Papyrus (рисунки 1–5).

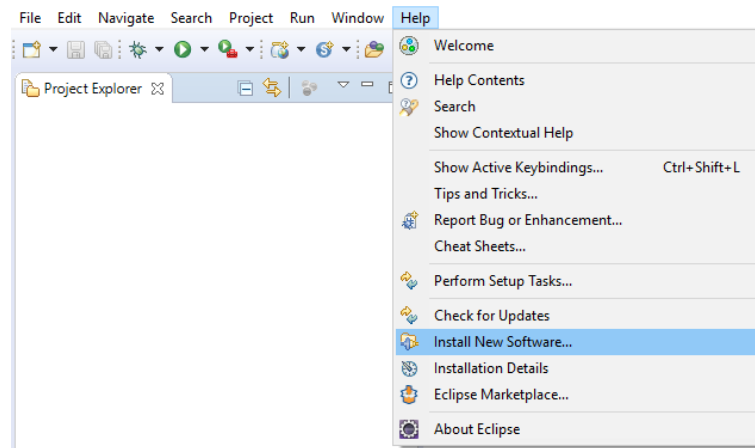


Рисунок 1

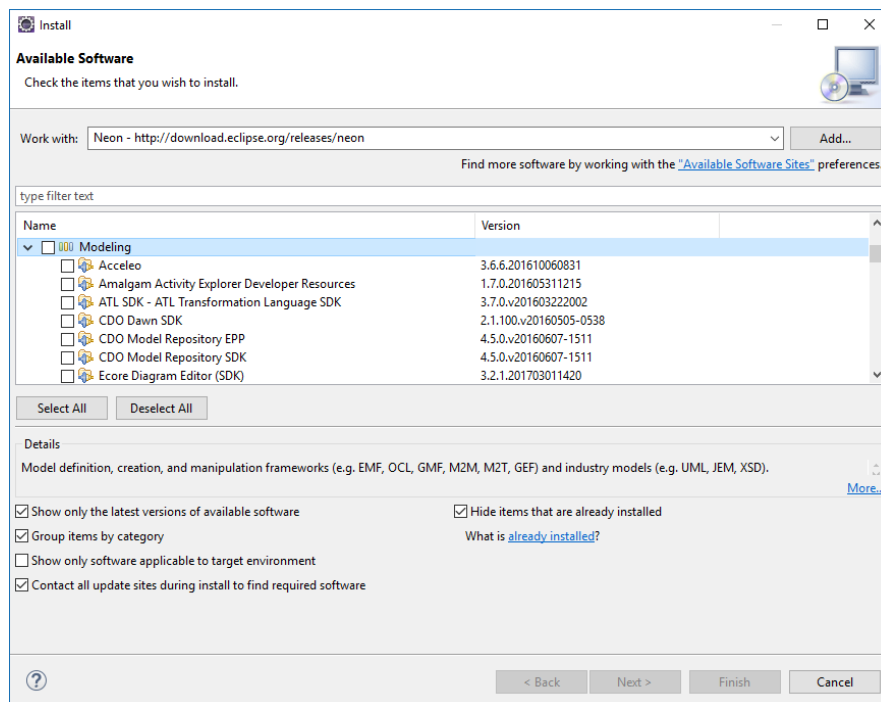


Рисунок 2

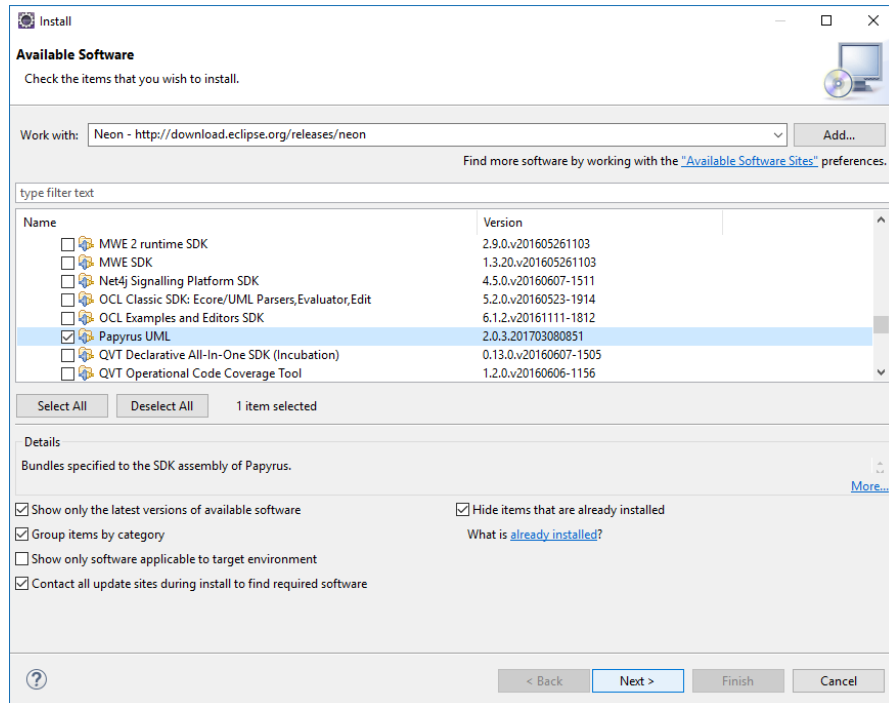


Рисунок 3

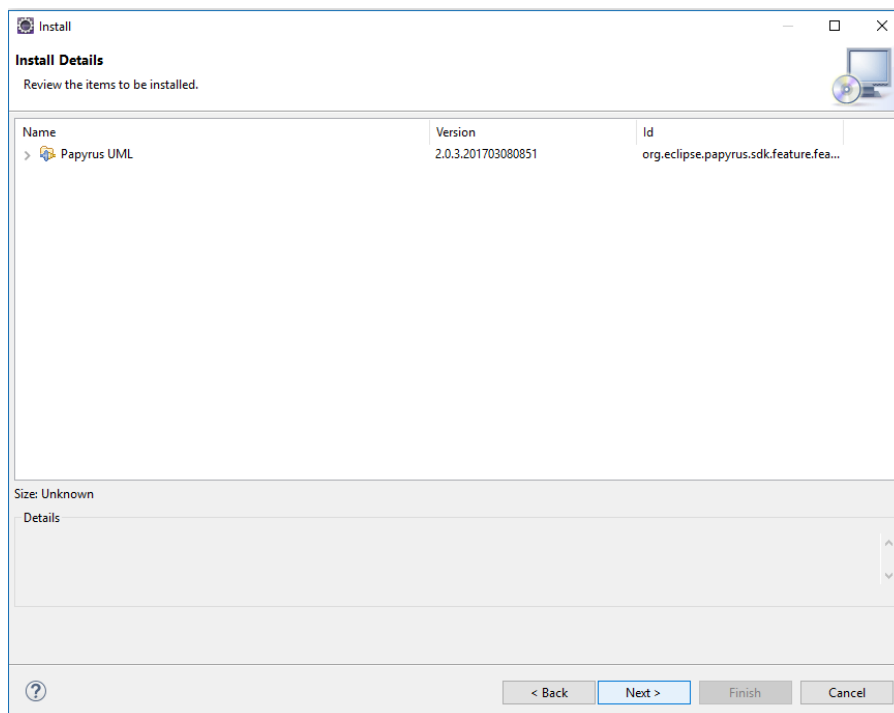


Рисунок 4

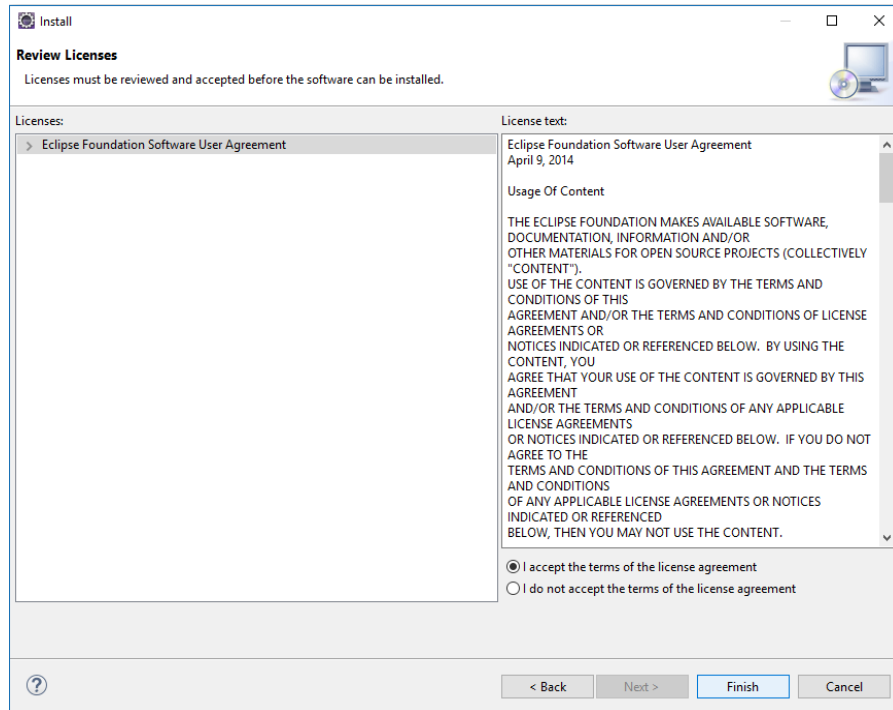


Рисунок 5

2. Для дальнейшего использования инструмента построения UML-диаграмм Papyrus в качестве проектирования программно-информационных систем необходимо установить плагин Designer-JAVA (рисунки 6–10)

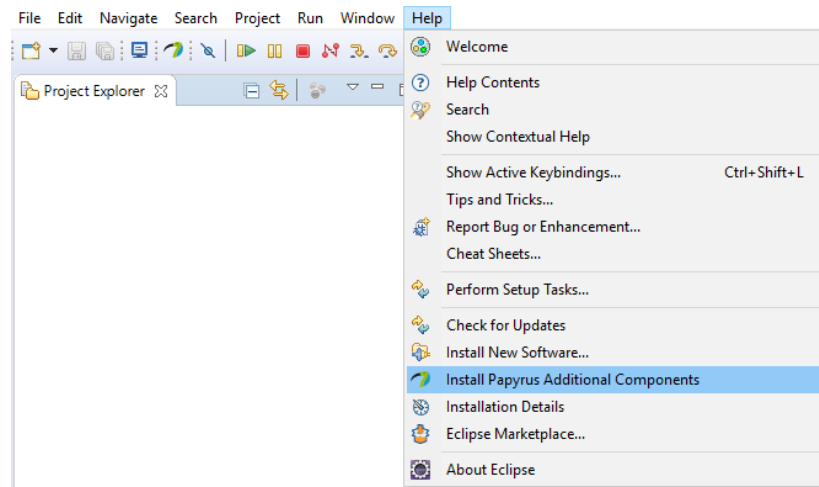


Рисунок 6

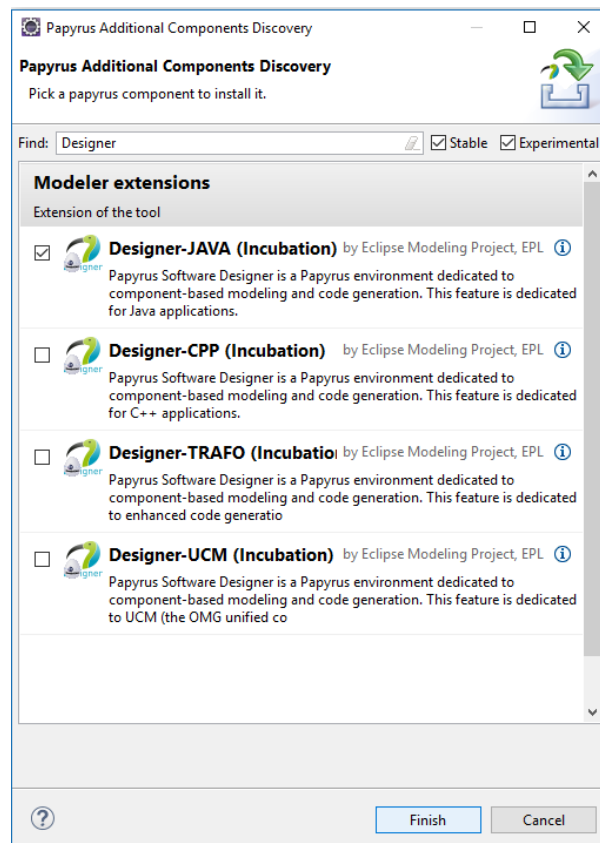


Рисунок 7



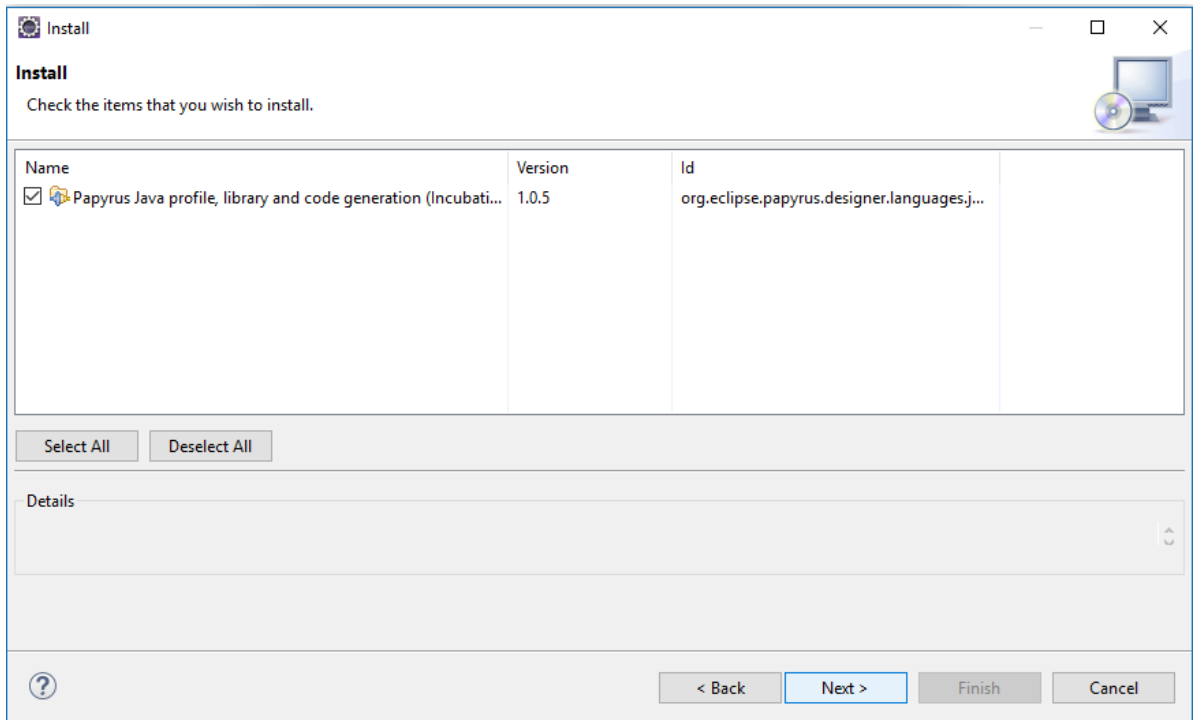


Рисунок 8

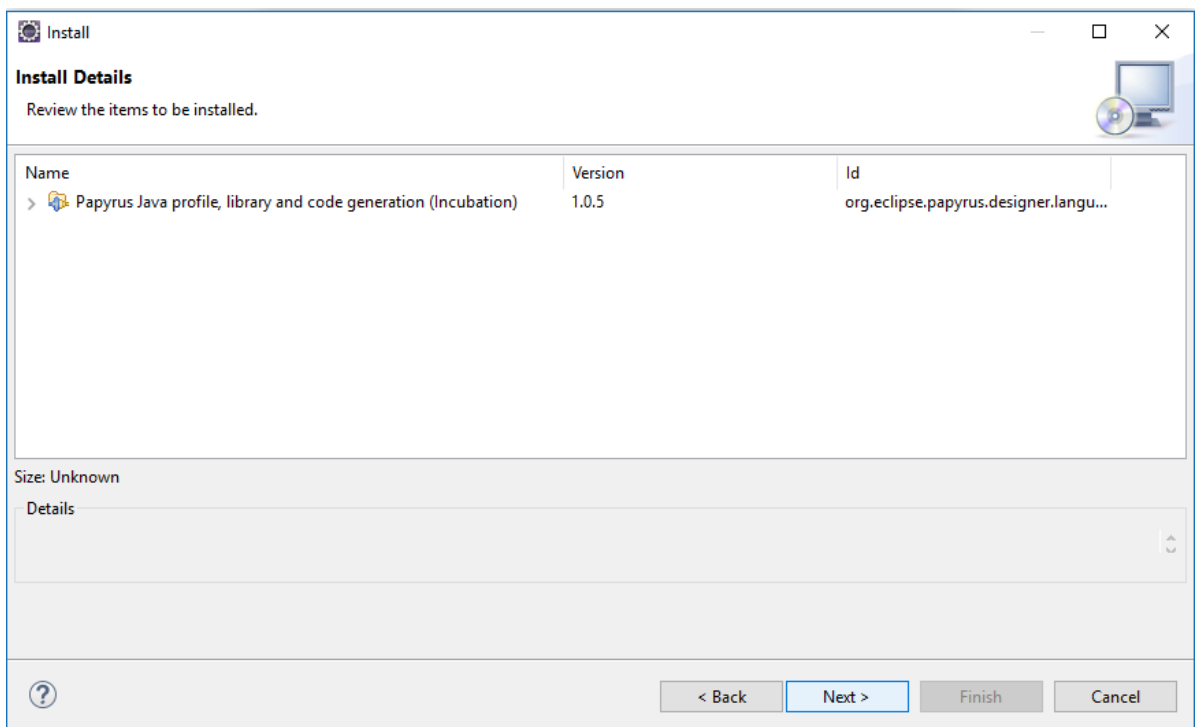


Рисунок 9

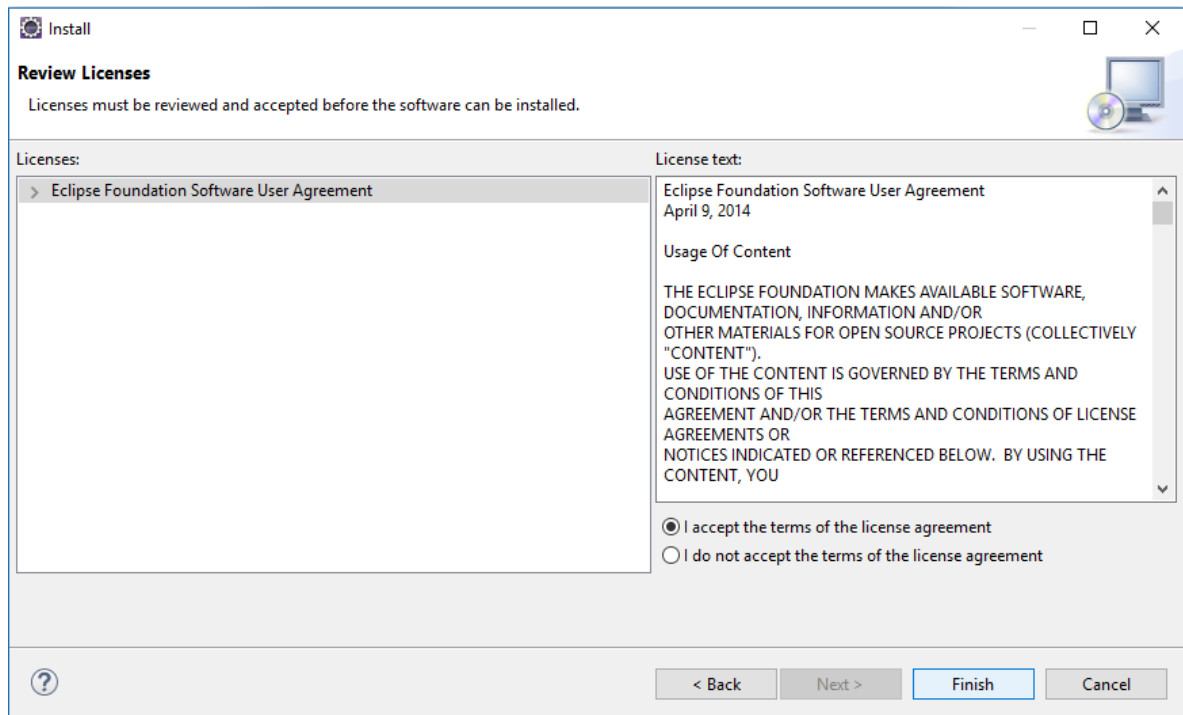


Рисунок 10

3. Для того, чтобы Java-код генерировался в проект, в котором находится диаграмма классов, необходимо в свойствах Preferences изменить префикс имени Java проектов при их генерации в код (рисунки 11–14).

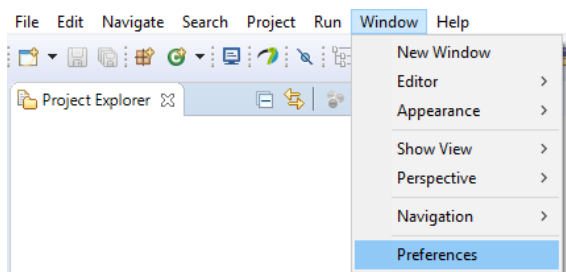


Рисунок 11

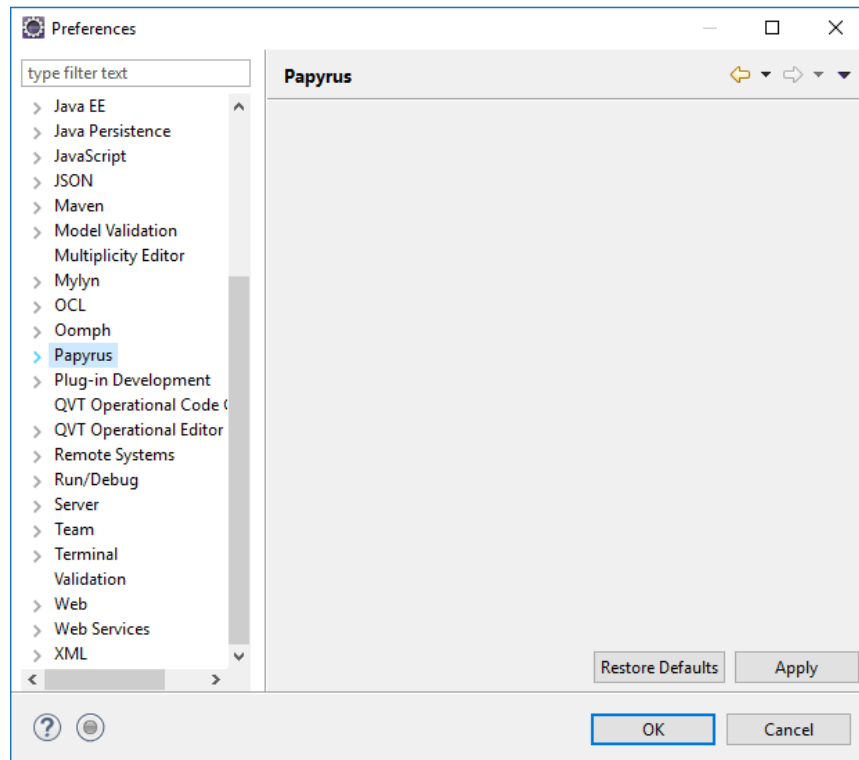


Рисунок 12

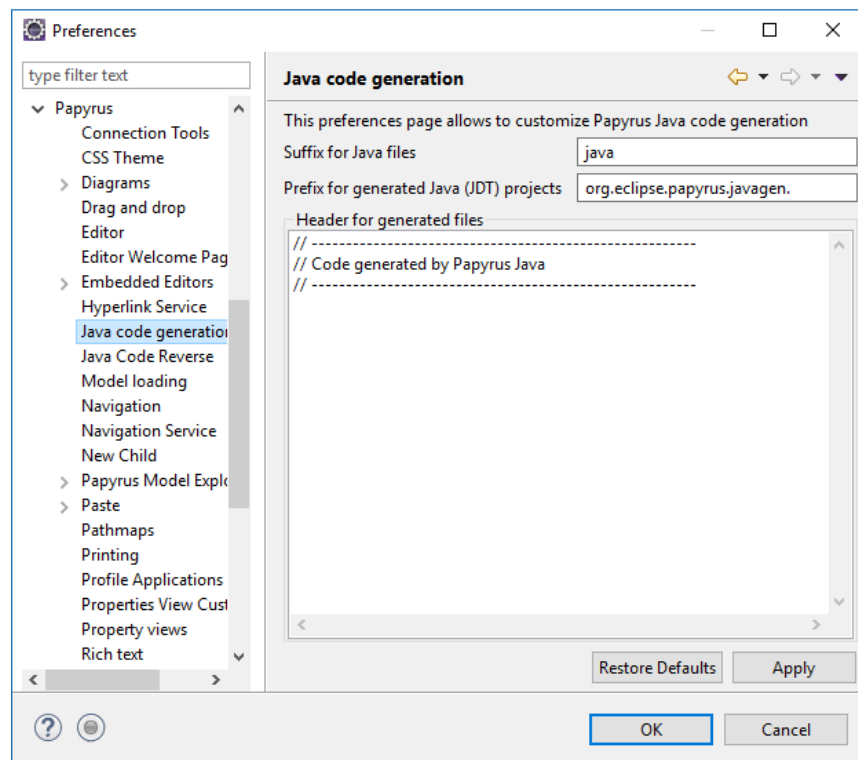


Рисунок 13

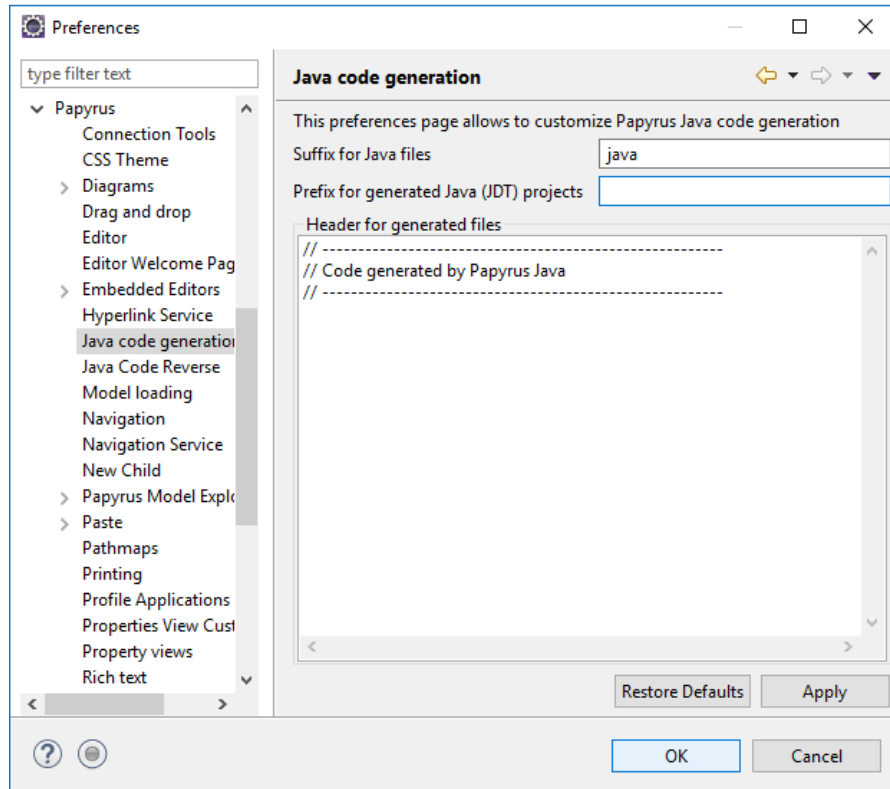


Рисунок 14

4. Для дальнейшей работы необходимо создать новый Java-проект (рисунки 15–16) и в нем создать папку для хранения разработанной диаграммы классов (рисунки 17–18).

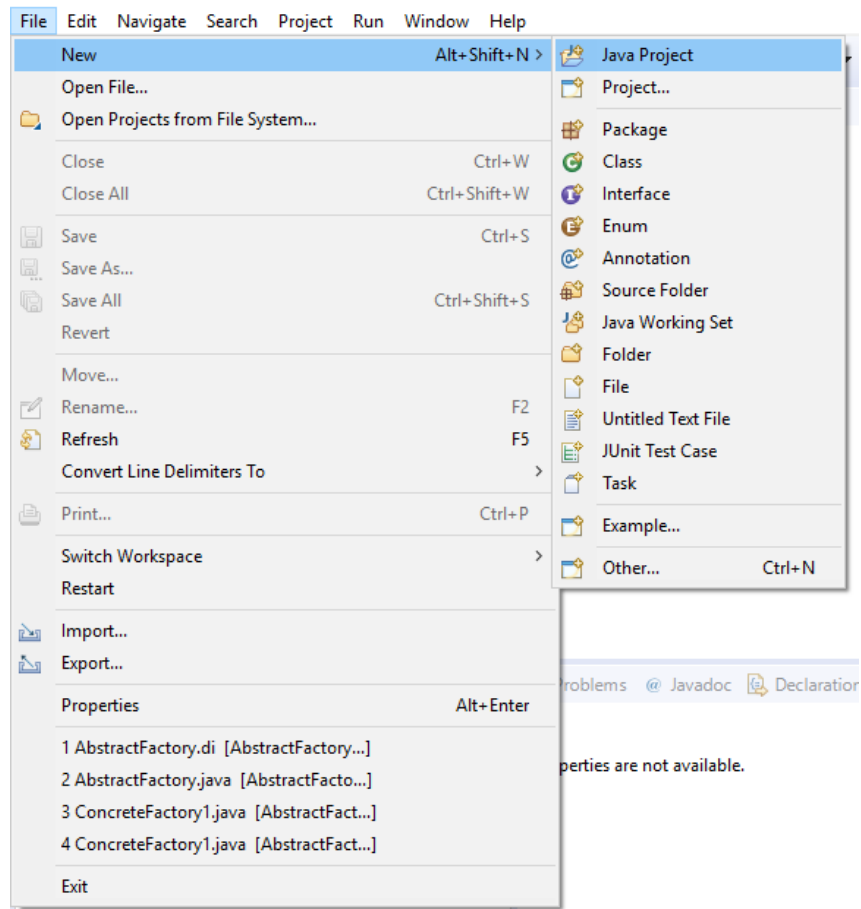


Рисунок 15

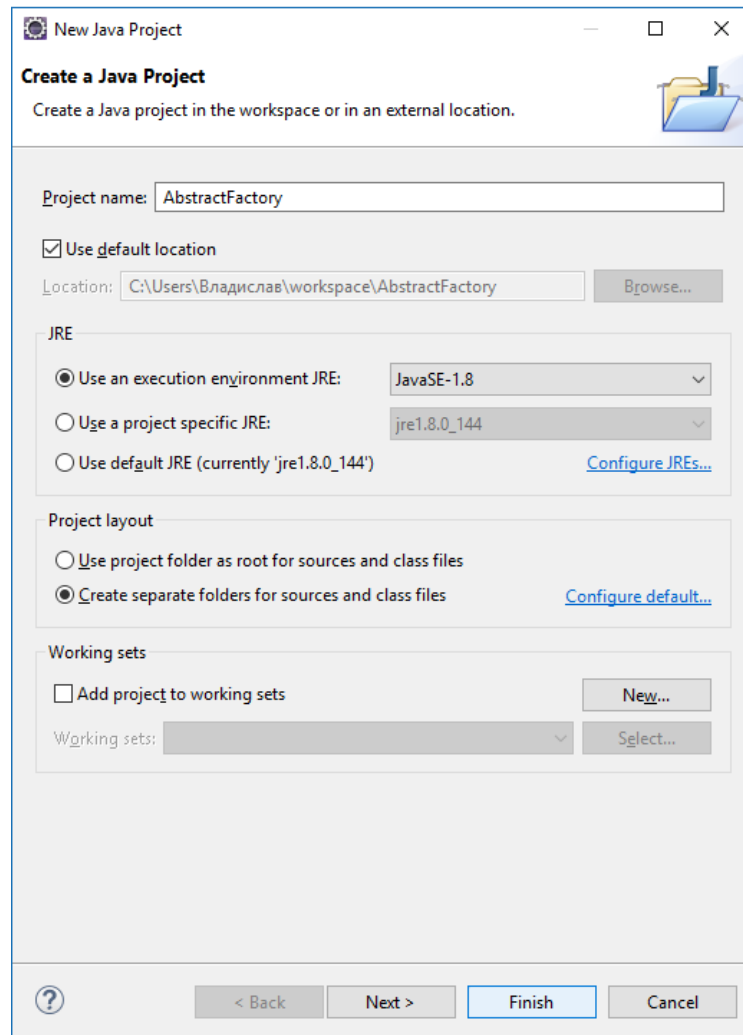


Рисунок 16

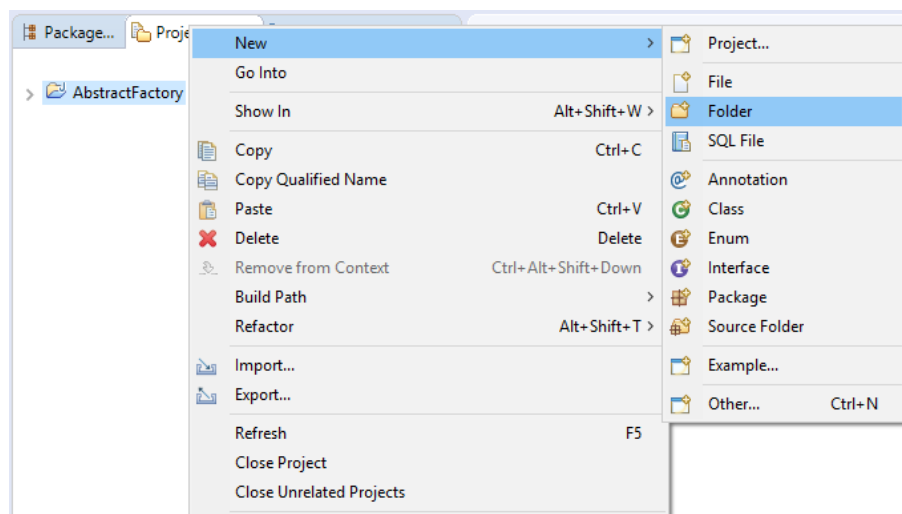


Рисунок 17

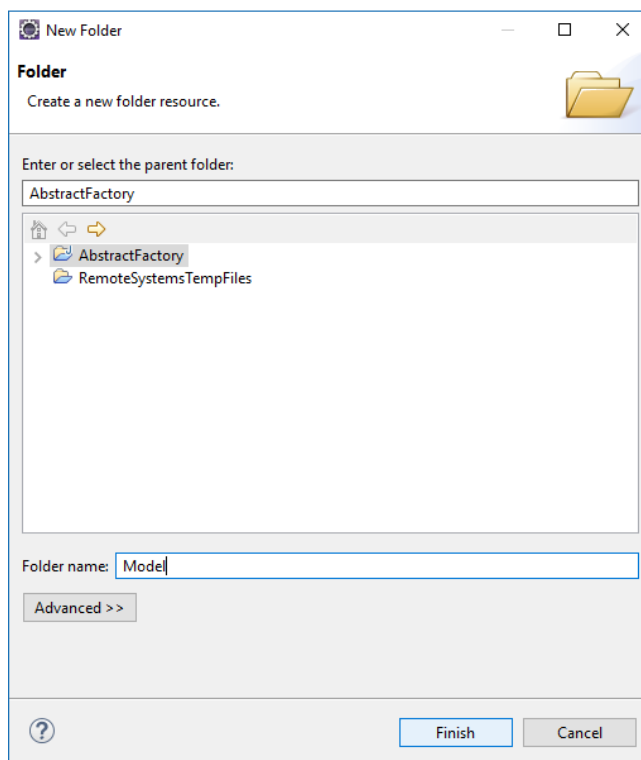


Рисунок 18

5. В созданной папке необходимо создать файлы для работы с инструментом проектирования UML-диаграмм Papyrus (рисунки 19–24). При этом наименование элемента корневой модели (Root model element name) нужно изменить на имя проекта (рисунок 23).

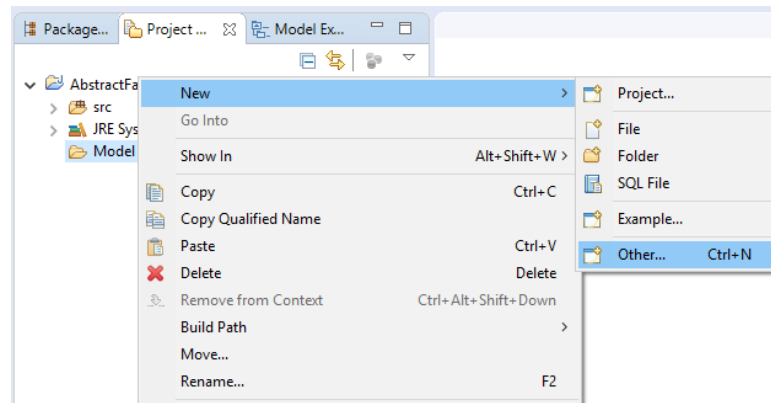


Рисунок 19

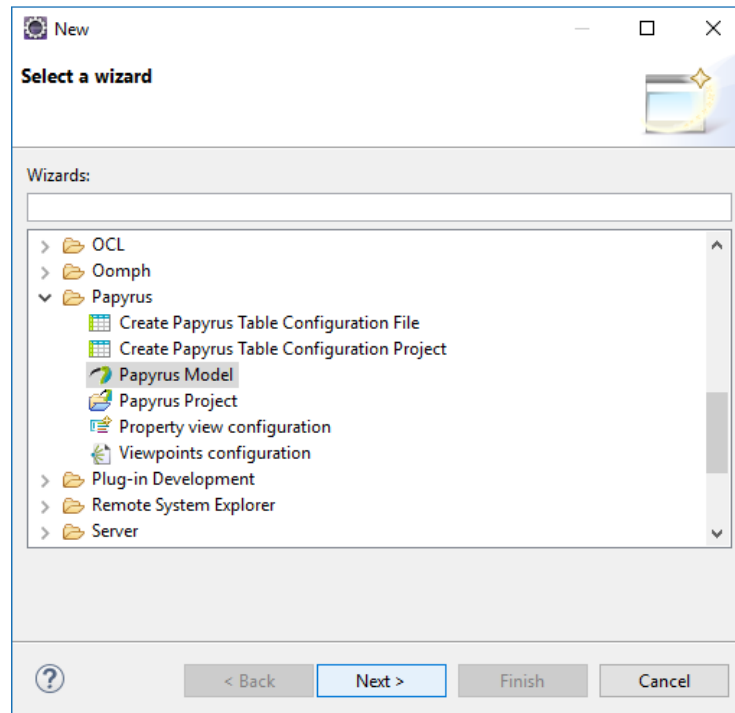


Рисунок 20



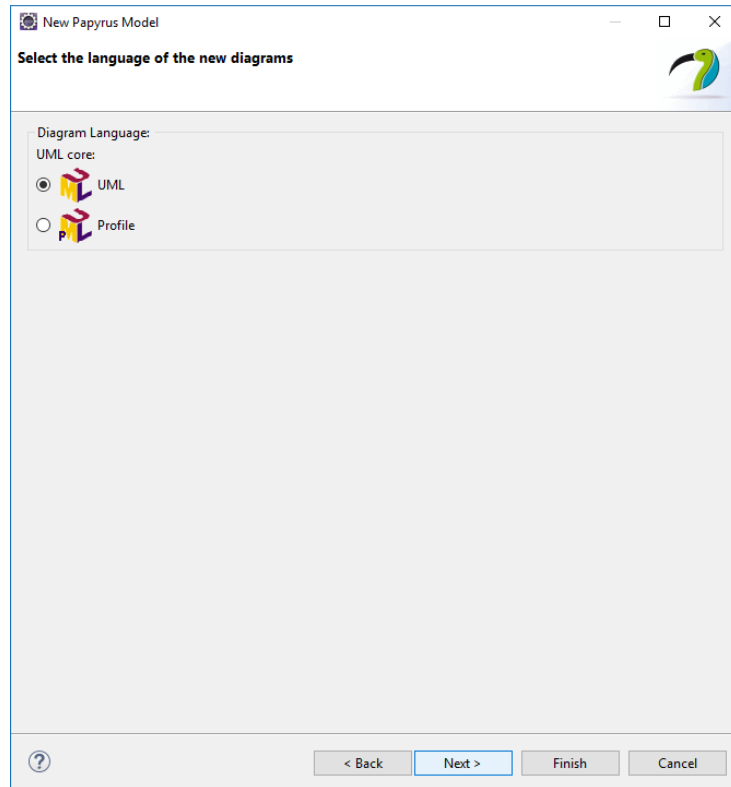


Рисунок 21

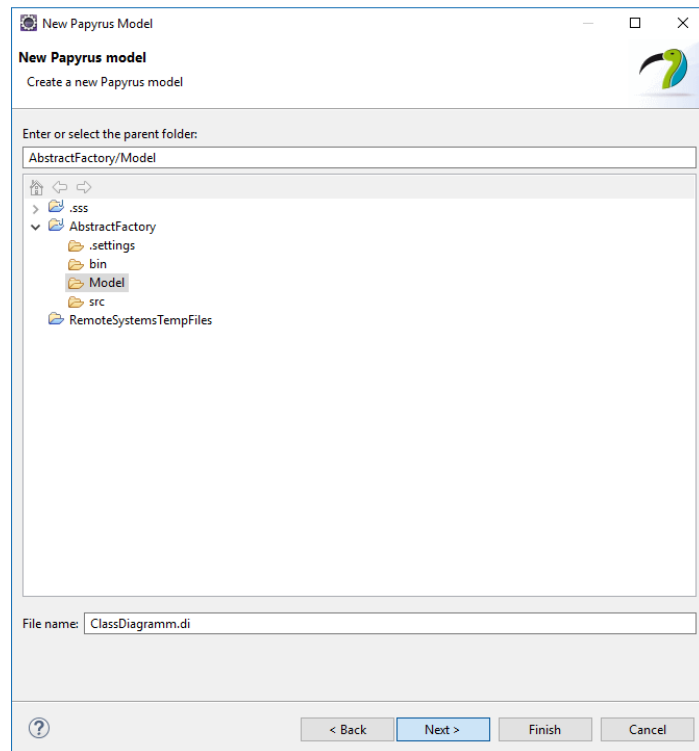


Рисунок 22

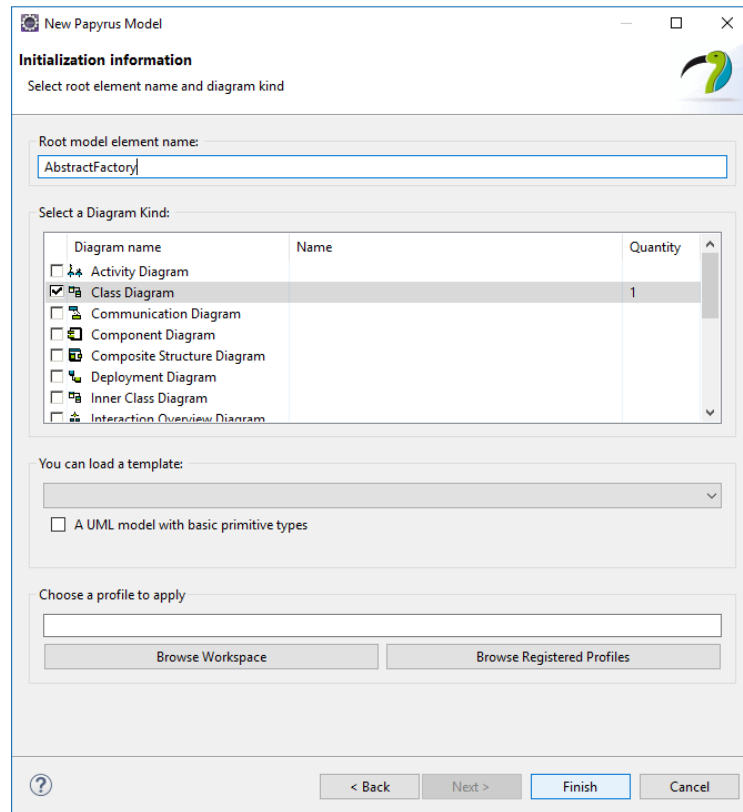


Рисунок 23

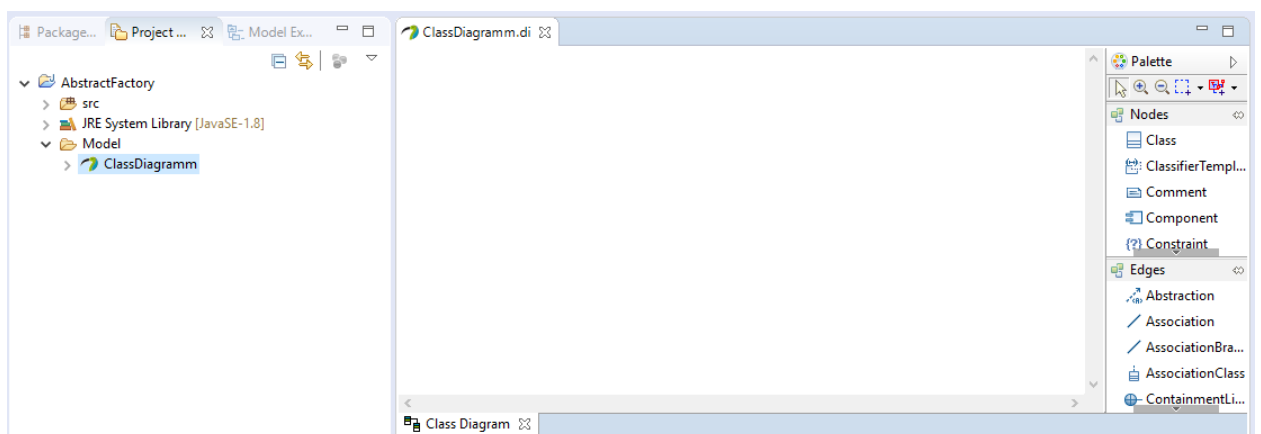


Рисунок 24

6. Для использования типов данных языка Java в диаграмме классов необходимо к созданной модели, которая находится в обозревателе моделей (рисунок 25), подключить библиотеку `JavaPrimitiveTypes` (рисунки 26–28).

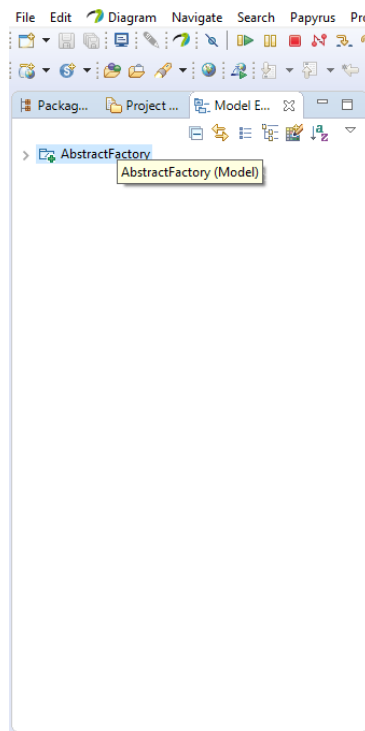


Рисунок 25

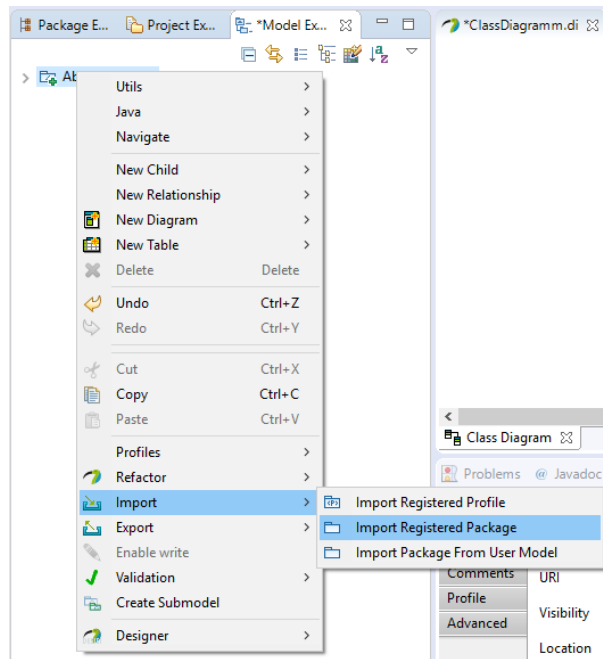


Рисунок 26

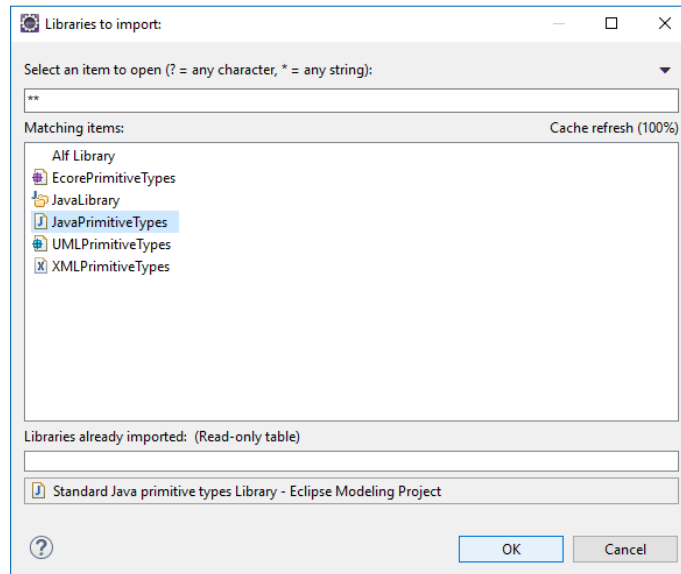


Рисунок 27

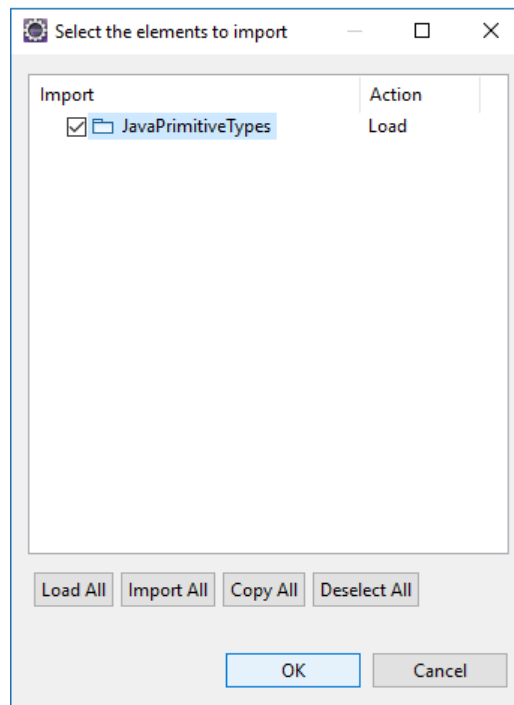


Рисунок 28

7. При реализации связи между интерфейсом и его классов-наследников в диаграмме классов необходимо переопределить методы в классах-наследниках, которые есть в интерфейсе (рисунок 29).

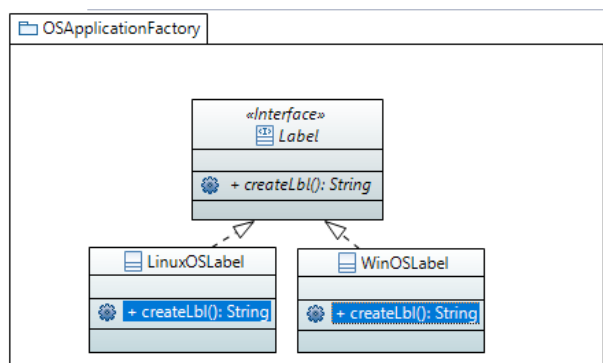


Рисунок 29

8. При реализации ассоциации между двумя элементами диаграммы классов необходимо учесть, что один класс создает в себе объект другого класса (или интерфейса), но не отображает его на диаграмме (рисунок 30).

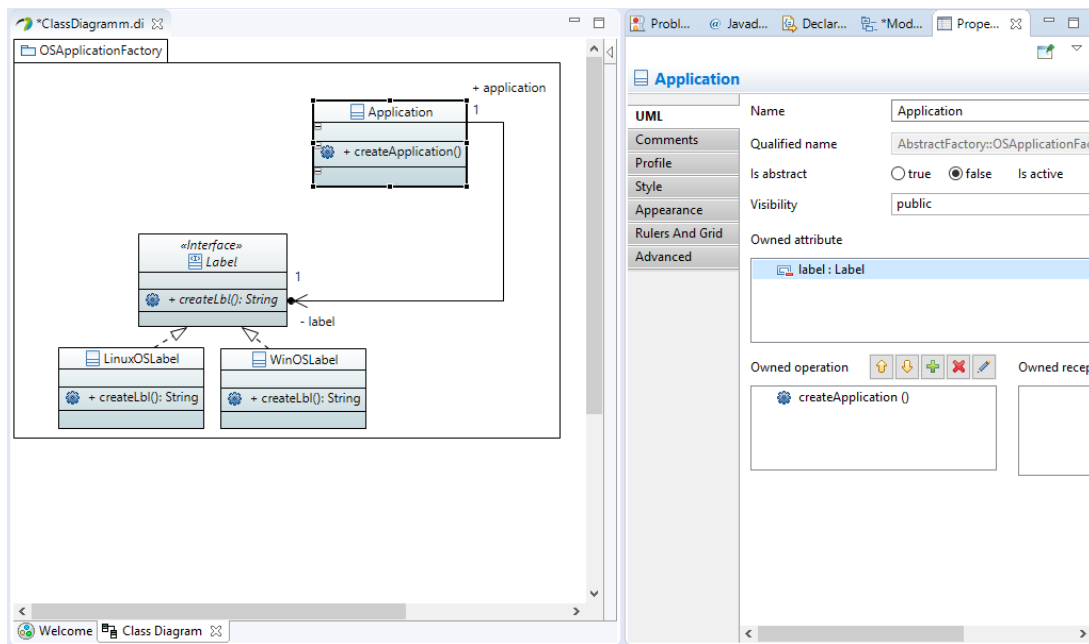


Рисунок 30

9. После разработки в проектировщике UML-диаграмм Rarugus необходимой диаграммы классов (рисунок 31) необходимо сгенерировать из модели Java-код (рисунок 32).

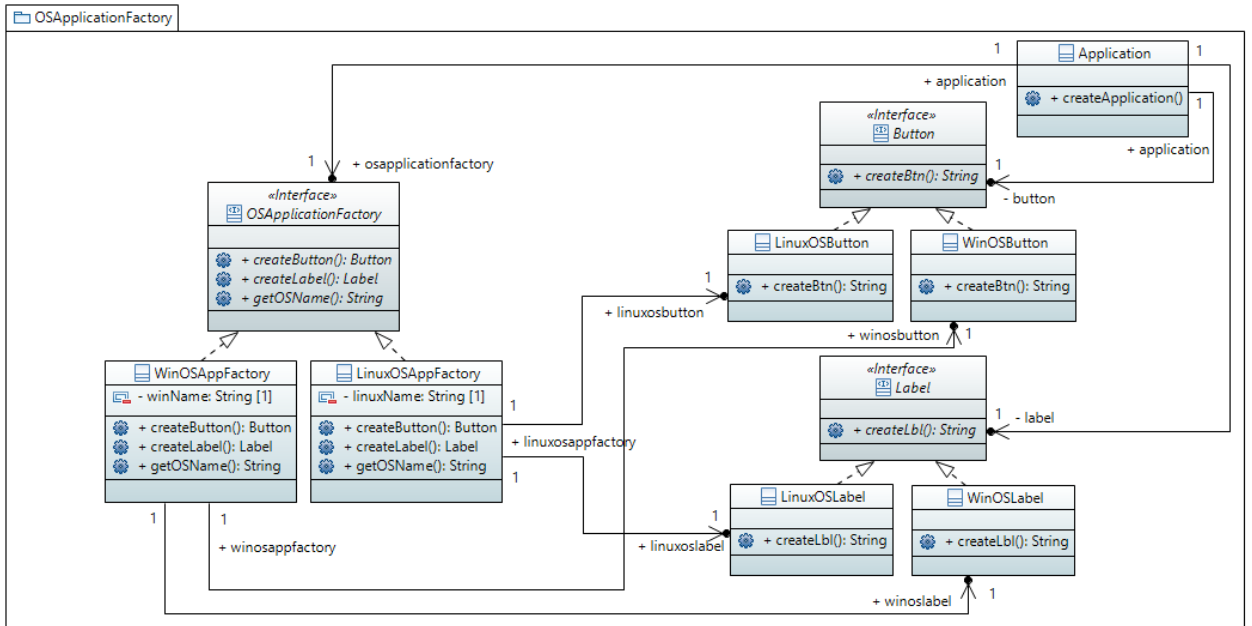


Рисунок 31

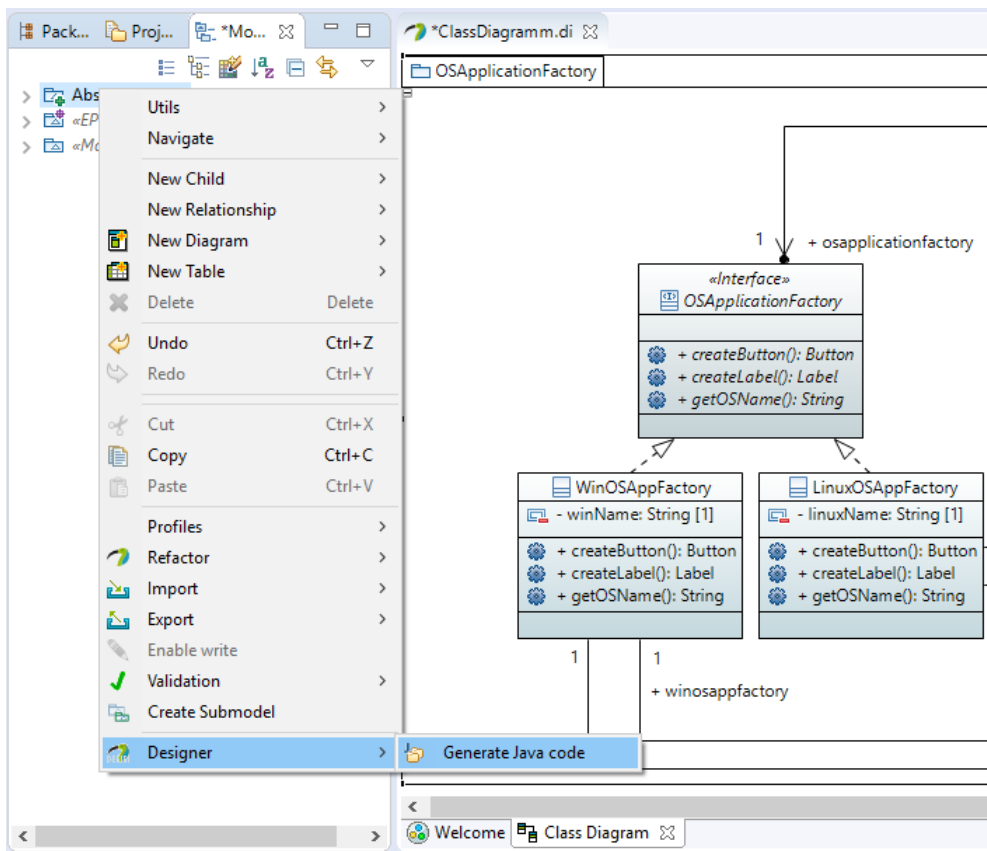


Рисунок 32

10. Сгенерированные объекты в виде Java-файлов можно увидеть в папке `src`, расположенной в созданном ранее проекте (рисунок 33).

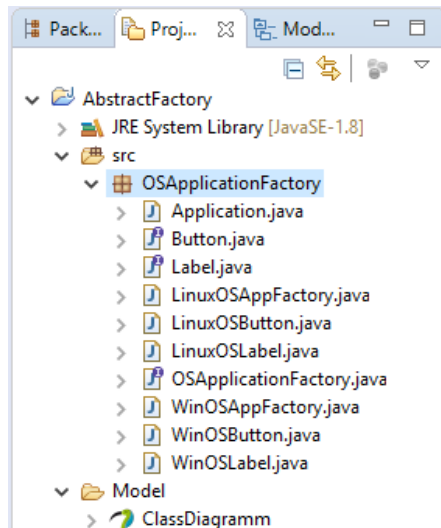


Рисунок 33

11. Примеры сгенерированного кода некоторых объектов представлены на рисунках 34–36.

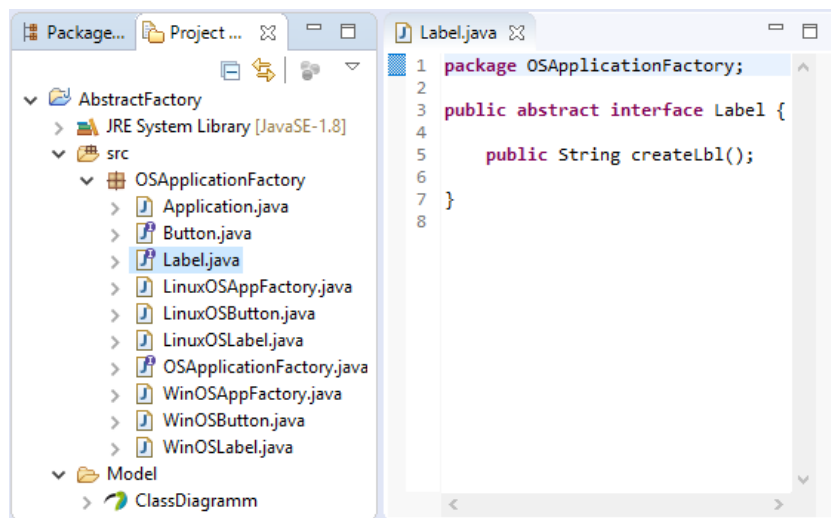


Рисунок 34



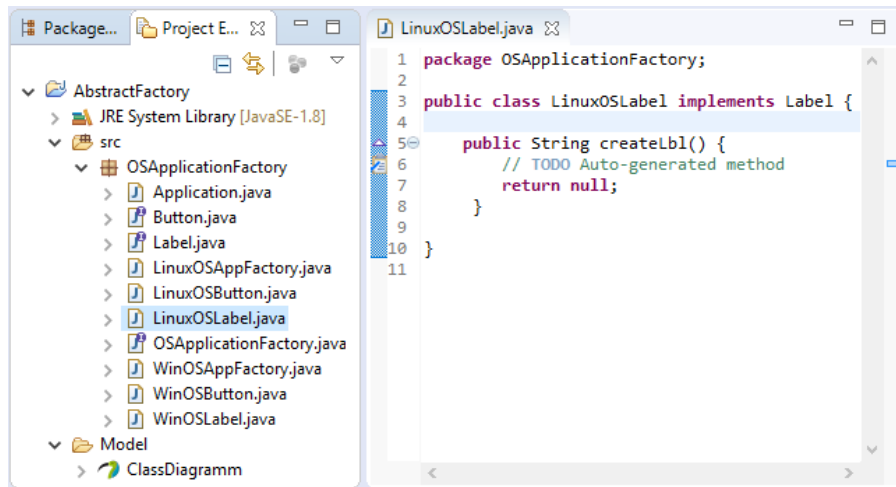


Рисунок 35

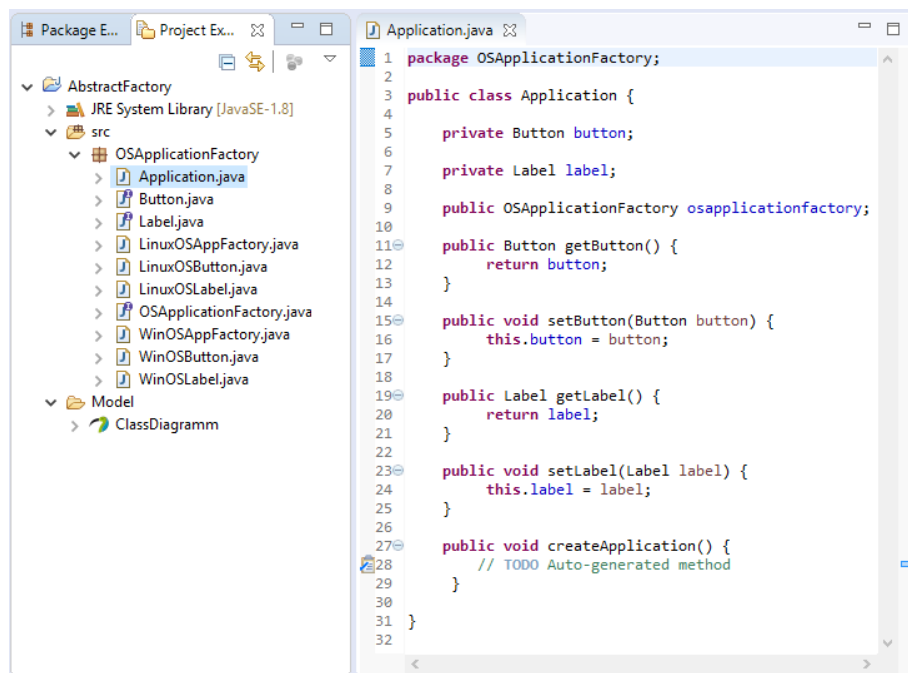


Рисунок 36

### **3 Содержание отчета по лабораторной работе**

В сводный отчет по лабораторным работам в качестве одного из разделов или подразделов включаются скриншоты, показывающие выполнение преобразования диаграммы классов, созданной в инструменте проектирования UML-диаграмм Parvus по индивидуальному заданию, в Java-код.

#### **4 Вопросы к защите лабораторной работы**

1. На какие этапы подразделяется проектирование программно-информационной системы?
2. Что такое модель программно-информационной системы?
3. Для чего используется диаграмма классов в проектировании программно-информационных систем?
4. Что такое прямое проектирование программно-информационной системы?
5. Чем обратное проектирование программно-информационных систем отличается от прямого проектирования?
6. Зачем используют язык моделирования UML в проектировании программно-информационных систем?
7. Каким образом можно реализовать программный код из диаграммы классов программно-информационных систем в интегрированной среде разработки Eclipse?

## 5 Индивидуальные задания

1. Реализовать диаграмму классов для системы покупки билета на авиарейс. Полученную модель сгенерировать в программный код.

2. Реализовать диаграмму классов для системы выдачи книг в библиотеке. Полученную модель сгенерировать в программный код.

3. Реализовать диаграмму классов для системы бронирования номера в гостинице. Полученную модель сгенерировать в программный код.

4. Реализовать диаграмму классов для системы обслуживания банковских кредитов. Полученную модель сгенерировать в программный код.

5. Реализовать диаграмму классов для системы оплаты покупок на кассе в супермаркете. Полученную модель сгенерировать в программный код.

6. Реализовать диаграмму классов для системы формирования штата предприятия. Полученную модель сгенерировать в программный код.

7. Реализовать диаграмму классов для системы учета автомобиле на платной стоянке. Полученную модель сгенерировать в программный код.