



Министерство науки и высшего образования Российской  
Федерации Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Московский государственный технический университет имени  
Н.Э. Баумана (национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ Информатика и системы управления и искусственный интеллект

КАФЕДРА Системы обработки информации и управления

**Лабораторная работа №6**  
**По курсу**  
**«Технологии разработки программного**  
**обеспечения»**  
**«Прямое и обратное проектирование»**

Подготовил:

Студент группы

**ИУ5-14Б Журавлев Н.В**

03.12.2023

Проверила:

**Виноградова М.В.**

*2023 г.*

## **Цель работы:**

- Изучить унифицированный процесс разработки (RUP);
- Приобрести умения построения модели проектирования и процессов прямого и обратного проектирования;
- Получить навыки построения модели проектирования и процессов прямого и обратного проектирования в среде Sparx Enterprise Architect.

## **Полученное задание:**

1. Открыть в среде Enterprise Architect проект, созданный ранее.
2. Установить для проекта профиль языка программирования, который будет использоваться при обозначении типов данных и генерации кода.
3. Создать диаграмму трассировки граничных классов в классы форм.
4. Создать диаграмму трассировки управляющих классов в классы проектирования [2]. Создать подсистему (package) и переместить туда полученные классы.
5. Выполнить прямое и обратное проектирование package подсистемы (сгенерировать код, изменить код, синхронизировать).
6. Создать диаграмму трассировки пакетов анализа в подсистемы.
7. Создать инфологическую модель.
8. Создать диаграмму трассировки классов сущностей в сущности инфологической модели.
9. Сгенерировать даталогическую модель на основе инфологической модели.
10. Сгенерировать DDL-сценарий на основе даталогической модели.
11. Сгенерировать документацию проекта.
12. Выполнить проверку целостности проекта.
13. Получить краткую статистику проекта.
14. Построить и изучить шаблон проекта базы данных (DDL, документация).

## **Ход работы:**

### **1. Выбор языка программирования**

Для выбора языка необходимо перейти в пункт Configure>Model>Options>Source Code Engineering, выбрать ЯП из выпадающего списка в Default Language for Code Generation. Будем использовать язык Python.

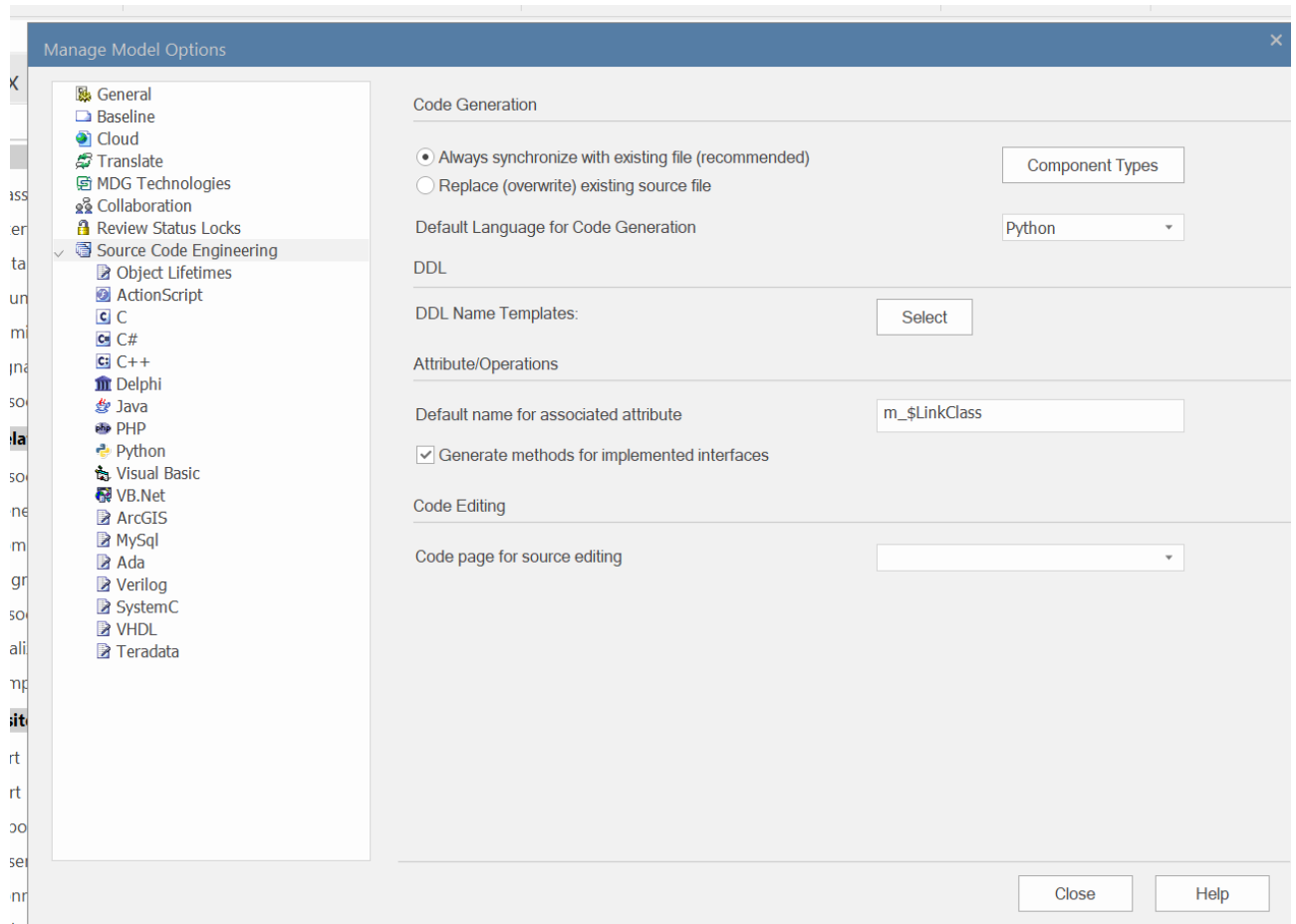


Рисунок 1. Выбор языка программирования

## 2. Диаграммы трассировки классов анализа в классы проектирования

### а. Трассировка управляющих классов

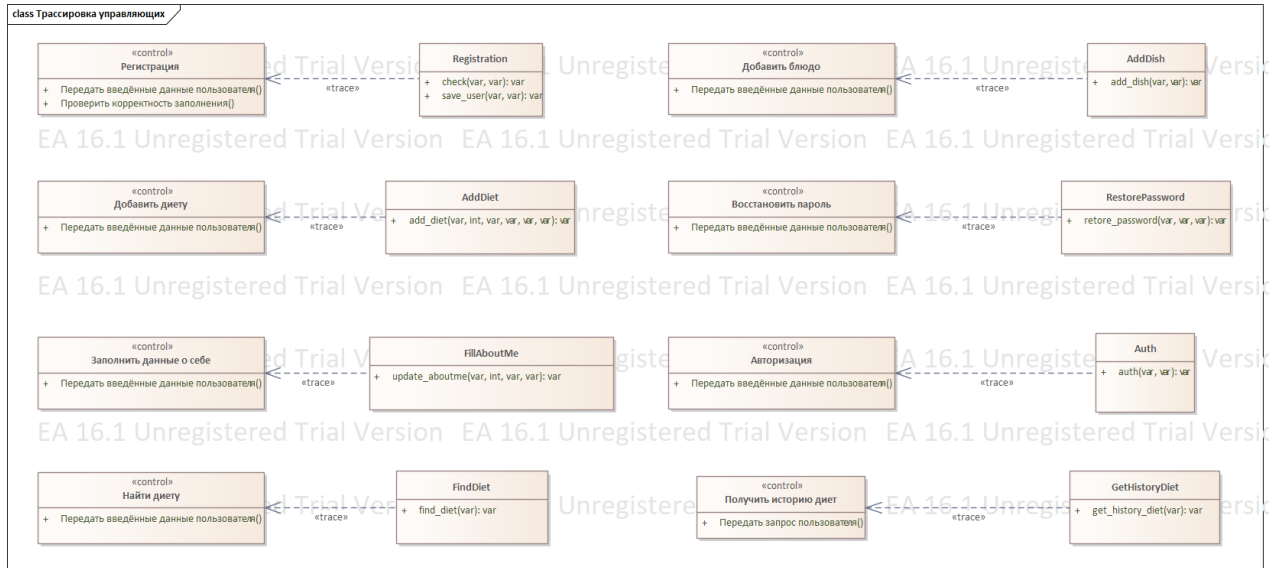


Рисунок 2. Управляющие классы

### б. Трассировка классов сущностей

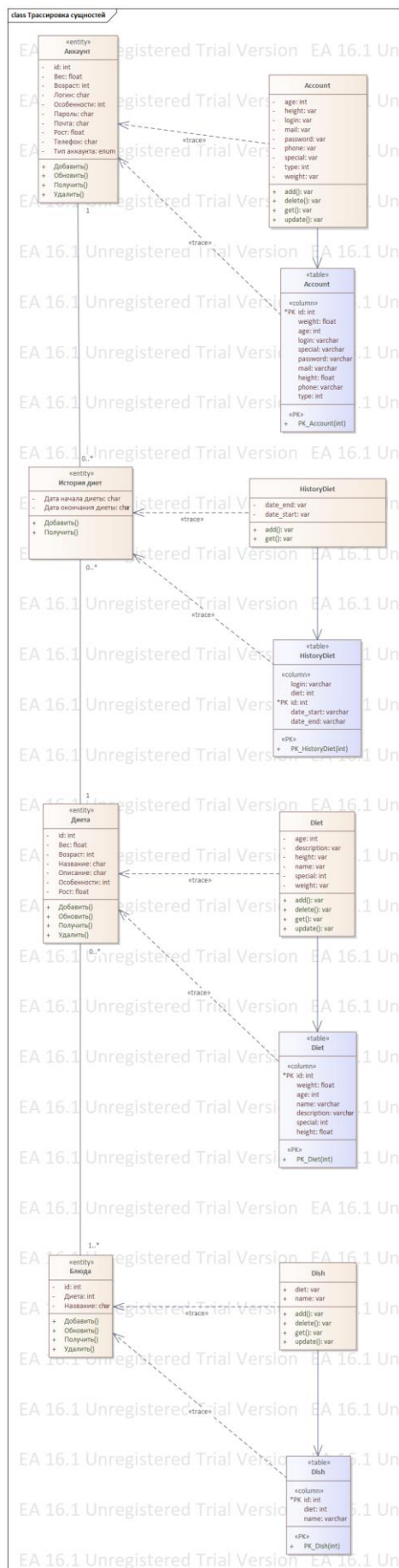


Рисунок 3. Трассировка классов сущностей

### с. Трассировка граничных классов

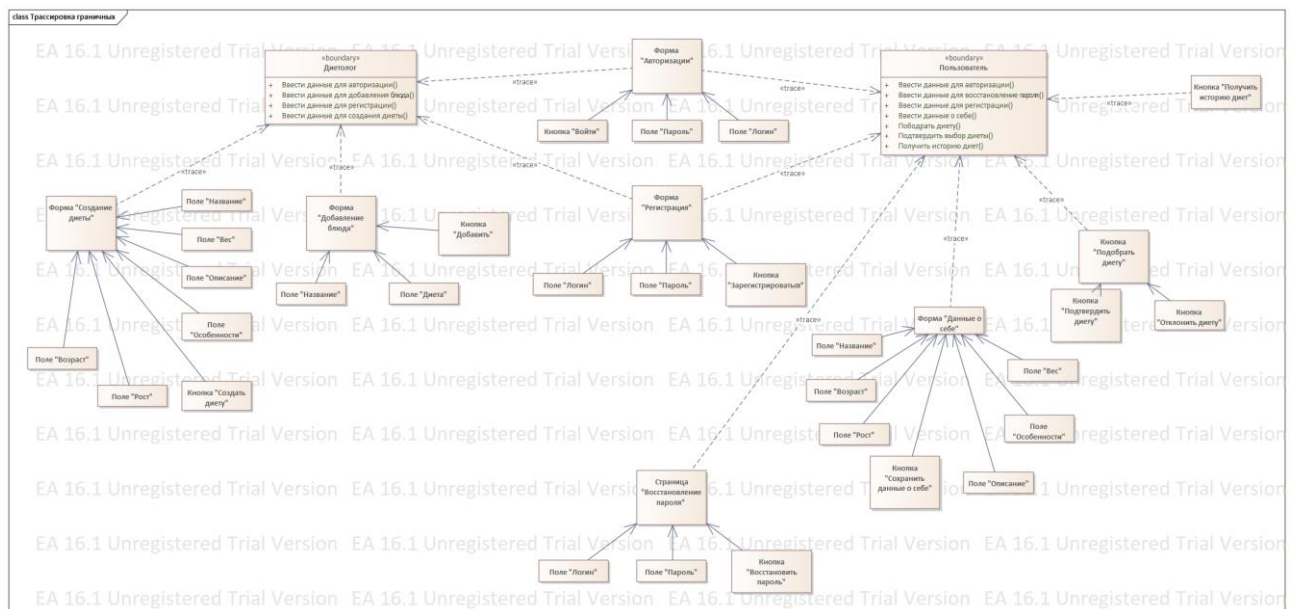


Рисунок 4. Трассировка граничных классов

### 3. Прямое и обратное проектирование

Выполним генерацию кода по данным классам – прямое проектирование. Для этого необходимо перейти в пункт меню Develop>Source Code>Generate>Generate All.

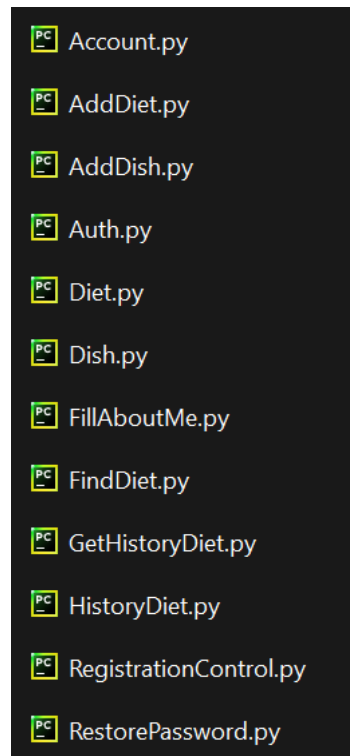


Рисунок 5. Результат прямого проектирования

### 4. Описание классов проектирования, их атрибутов, операций и их параметров

Класс Registration:

Операции:

`check(login, password)` – валидация логина и пароля.

`login` – логин

`password` - пароль

`save_user(login, password)` – добавить нового пользователя.

login – логин  
password - пароль

Класс AddDiet:

Операции:

add\_diet(weight, age, name, description, special, height) – добавление новой диеты.

weight – вес

age – возраст

name – название диеты

description – описание диеты

special – особенности

height - рост

Класс FillAboutMe:

Операции:

update\_aboutme(weight, age, special, height) – обновить информацию о пользователе.

weight – вес

age – возраст

special – особенности

height - рост

Класс FindDiet:

Операции:

find\_diet(login) – подобрать диету для пользователя.

login - логин

Класс AddDish:

Операции:

add\_dish(name, diet) – добавить блюдо.

name – название

diet - диета

Класс RestorePassword:

Операции:

restore\_password(login, phone, password) – восстановить пароль.

login – логин

phone – телефон

password – новый пароль

Класс GetHistoryDiet:

Операции:

get\_history\_diet(login) – получить историю диет пользователя.

login - логин

Класс Auth:

Операции:

auth(login, password) – авторизация.

login – логин

password – новый пароль

Класс Account:

Атрибуты:

age - возраст  
height - рост  
login - логин  
mail - почта  
password - пароль  
phone - телефон  
special - особенности  
type – тип аккаунта  
weight - вес

Операции:

add() – добавить пользователя  
delete() – удалить пользователя  
get() – получить информацию о пользователе  
update() – обновить информацию о пользователе

Класс HistoryDiet:

Атрибуты:

date\_end - дата окончания диеты  
date\_start – дата начал диеты

Операции:

add() – добавить диету в историю  
get() – получить всю историю диет

Класс Diet:

Атрибуты:

age - возраст  
description - описание  
height - рост  
name - имя  
special - особенности  
weight - вес

Операции:

add() – добавить диету  
delete() – удалить диету  
get() – получить диету  
update() – обновить диету

Класс Dish:

Атрибуты:

name - название  
diet - диета

Операции:

add() – добавить блюдо  
delete() – удалить блюдо  
get() – получить блюдо  
update() – обновить блюдо

## **5. Диаграмма трассировки пакетов в подсистемы**

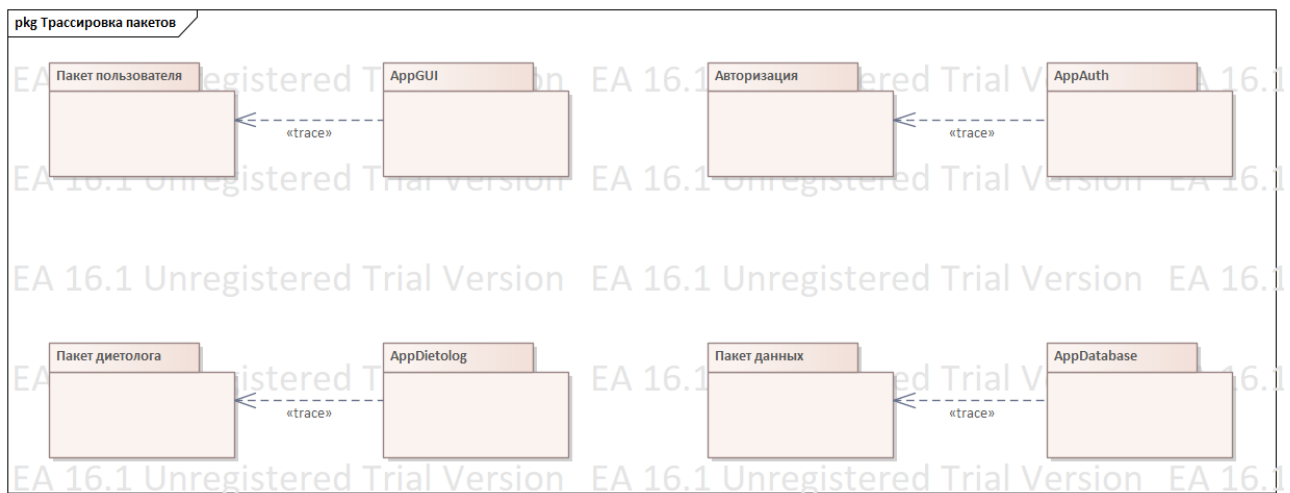


Рисунок 6. Дигарамма трассировка пакетов

## 6. Инфологическая и даталогическая модель (диаграммы)

Для генерации требуется выполнить следующие действия:

1. Выбрать в проводнике нужный package с инфологической моделью.
2. Перейти в пункт Design>Tools>Transform>Apply Transformation...
3. В открывшемся окне убедиться, что выбраны все классы диаграммы. Затем из списка возможных трансформаций выбрать ERD to Data Modeling. В открывшемся доп. окне выбрать целевой package, куда будет сохранена новая диаграмма и ее элементы.
4. Нажать Do Transform.





Рисунок 7. Даталогическая модель

Для генерации инфологической нужно выполнить следующие действия:

1. Выбрать в проводнике нужный package с диаграммой классов.
2. Перейти в пункт Design>Tools>Transform>Apply Transformation.
3. В открывшемся окне убедиться, что выбраны все классы диаграммы. Затем из списка возможных трансформаций выбрать Data Modeling to ERD. В открывшемся доп. окне выбрать целевой package, куда будет сохранена новая диаграмма и ее элементы.

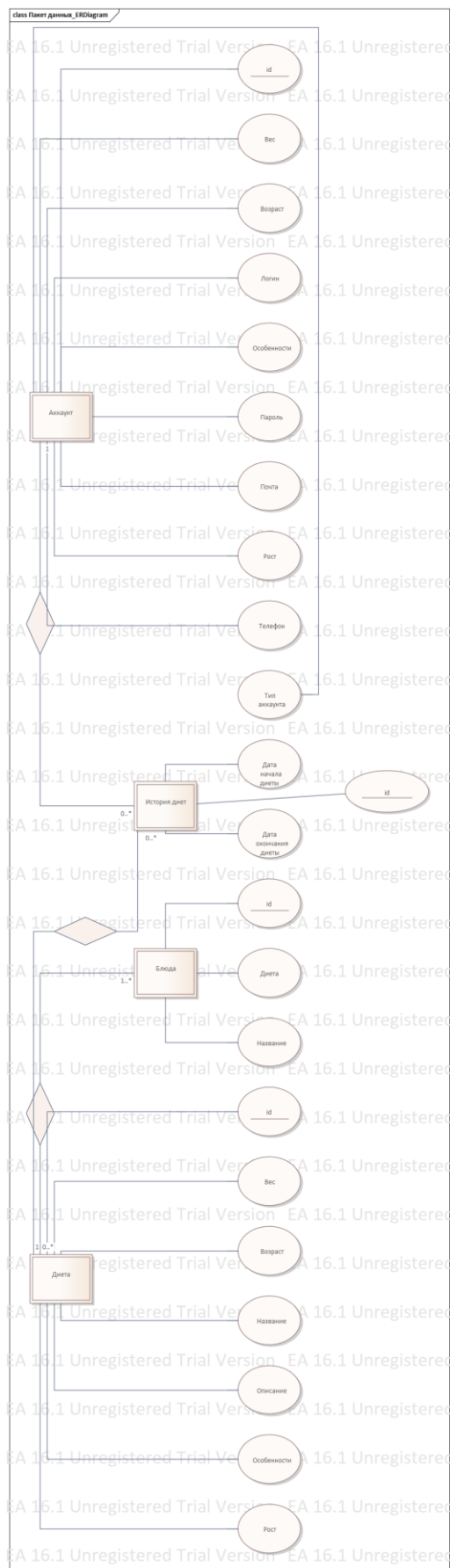


Рисунок 8. Инфологическая модель

## 7. DDL-сценарий

Для генерации DDL-сценария необходимо выполнить следующие действия:

1. Выбрав в обозревателе нужную диаграмму, перейти в Develop>Data Modeling>Generate.

2. В появившемся окне указать путь до будущего файла DDL-сцена

```
/* ----- */
/* Generated by Enterprise Architect Version 16.1 */
/* Created On : 01-дек-2023 16:50:56 */
/* DBMS : PostgreSQL */
/* ----- */

/* Drop Tables */

DROP TABLE IF EXISTS "Аккаунт" CASCADE
;

DROP TABLE IF EXISTS "Блюда" CASCADE
;

DROP TABLE IF EXISTS "Диета" CASCADE
;

DROP TABLE IF EXISTS "История диет" CASCADE
;

/* Create Tables */

CREATE TABLE "Аккаунт"
(
    id int NULL,
    "Вес" float NULL,
    "Возраст" int NULL,
    "Логин" char NULL,
    "Особенности" int NULL,
    "Пароль" char NULL,
    "Почта" char NULL,
    "Рост" float NULL,
    "Телефон" char NULL,
    "Тип аккаунта" enum NULL
)
;

CREATE TABLE "Блюда"
(
    id int NULL,
    "Диета" int NULL,
    "Название" char NULL
)
;

CREATE TABLE "Диета"
(
    id int NULL,
    "Вес" float NULL,
    "Возраст" int NULL,
    "Название" char NULL,
    "Описание" char NULL,
    "Особенности" int NULL,
    "Рост" float NULL
)
;
```

Рисунок 9. DDL-сценарий

## 8. Пример документации проекта (скриншот)

Для того, чтобы сгенерировать документацию нужно перейти в пункт Publish > Model Reports > Report Builder > Generate Documentation > Generate.

## Заполнить данные о себе

UseCase in package 'Use\_case'

Заполнить данные о себе

Version 1.0 Phase 1.0 Proposed

Kolya created on 20.11.2023. Last modified 20.11.2023

### INCOMING BEHAVIORAL RELATIONSHIPS

⇒ Trace «trace» from Заполнить данные о себе to Заполнить данные о себе

### INCOMING STRUCTURAL RELATIONSHIPS

⇒ Trace «trace» from Заполнить данные о себе to Заполнить данные о себе

[ Direction is 'Source -> Destination'. ]

### CONNECTORS

➤ **Include** «include» Source -> Destination

From: Подобрать диету : UseCase, Public

To: Заполнить данные о себе : UseCase, Public

➤ **Trace** «trace» Source -> Destination

From: Заполнить данные о себе : Collaborate, Public

To: Заполнить данные о себе : UseCase, Public

Page 41 of 242

Model Report

3 December, 2023

### ASSOCIATIONS

➤ Association (direction: Unspecified)

Source: Public (Actor) Пользователь

Target: Public (UseCase) Заполнить данные о себе

Рисунок 10. Фото документации

## 9. Результат проверки целостности проекта

Выполним проверку целостности через Configure > Model > Integrity > Project Integrity.

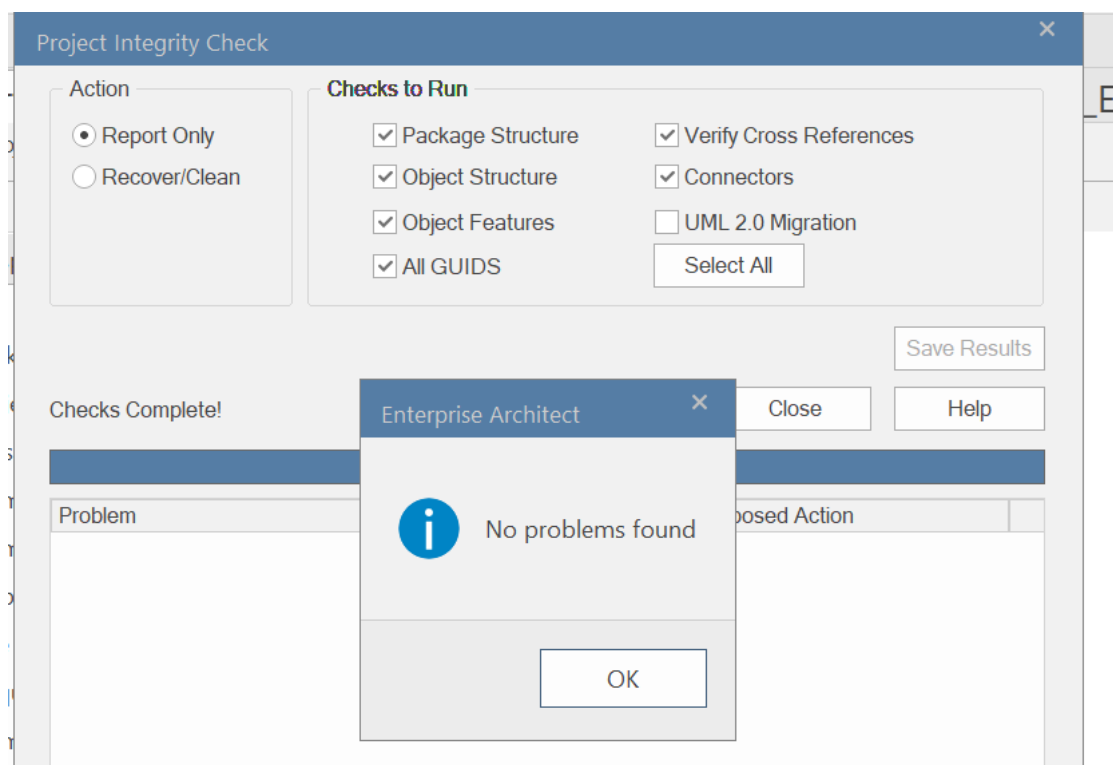


Рисунок 11. Репорт проверки

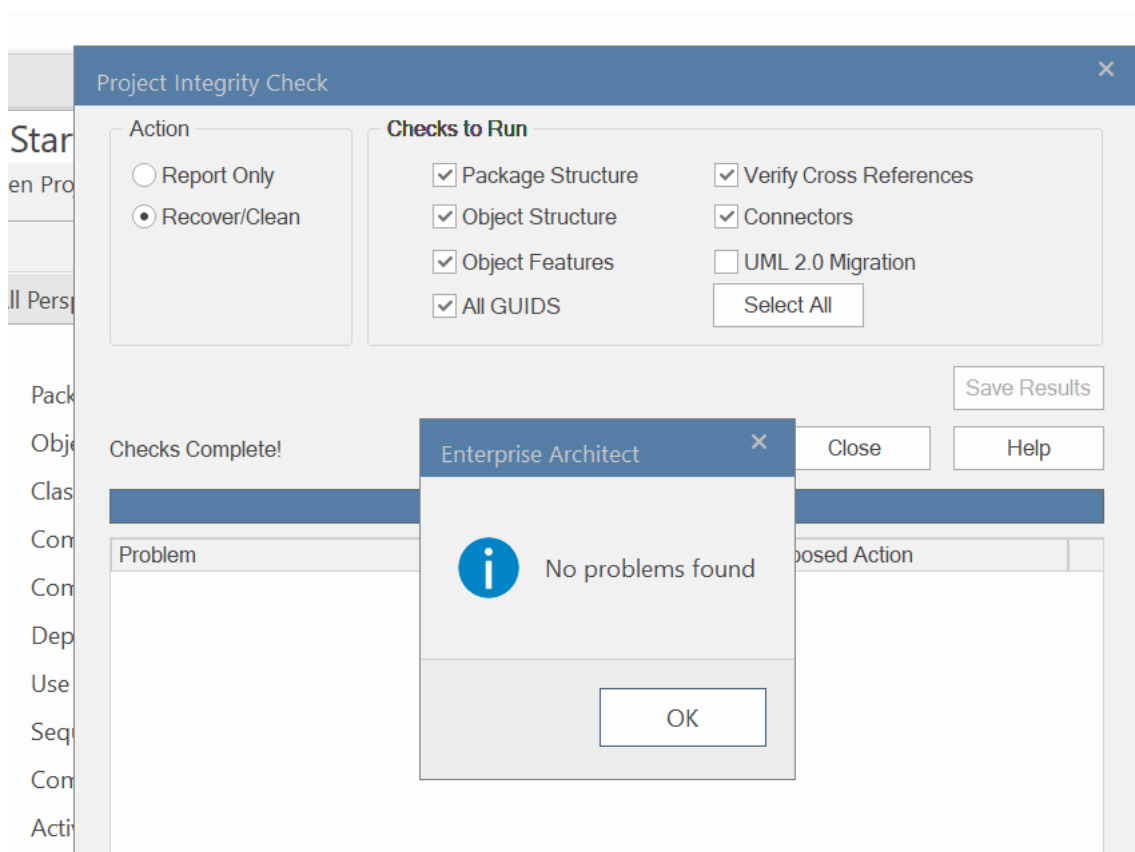


Рисунок 12. Атоматическое исправления ошибок

## 10. Краткая статистика проекта

Для просмотра статистики по содержимому проекта необходимо перейти в пункт: Construct > Project Management > QA > Statistic.

Start Page Project Statistics x	
Measure	Count
Total Packages	32
Total Diagrams	28
Total Elements	168
Total Connections	177
Elements in Diagrams	218
Element Attributes	111
Element Operations	74
Element Operation Parameters	42
Element Testing	0
Element Maintenance	0
Element Scenarios	0
Element Constraints	0
Element Requirement	0
Element Resource Allocation	0
Element Effort	0
Element Risks	0
Element Metrics	0
Element Files	0
Actor	2
Artifact	4
Boundary	3
Class	112
Collaborate	8
Package	31
UseCase	8

Рисунок 13. Статистика по проекту

### Выводы:

В ходе выполнения лабораторной работы были приобрести умения построения модели проектирования и процессов прямого и обратного проектирования. Получены навыки построения модели проектирования и процессов прямого и обратного проектирования в среде Sparx Enterprise Architect.

### Список источников

1. Sparx Systems – Текст. Изображение.: электронные // Sparx Systems : [сайт]. – URL: <https://sparxsystems.com/> (дата обращения 15.06.2022)
2. Виноградова, М. В. Унифицированный процесс разработки программного обеспечения: учебное пособие / М. В. Виноградова, В. И. Белоусова. — Москва: Издательство МГТУ им. Н.Э. Баумана, 2015. — 80, [2] с.: ил. ISBN 978-5-7038-4265-2
3. Якобсон А., Дуч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. / А. Якобсон, Г. Дуч, Дж. Рамбо. – Спб.: Питер. – 2002.
4. Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание. – Пер. с англ. – СПб: СимволПлюс, 2007. – 624 с., ил. ISBN13: 9785932860946 ISBN10: 5932860944
5. Руководство пользователя Enterprise Architect 15.1 – Текст. Изображение.:электронные //SparxSystems [https://sparxsystems.com/enterprise\\_architect\\_user\\_guide/15.1/index/index.html](https://sparxsystems.com/enterprise_architect_user_guide/15.1/index/index.html) обращения 15.06.2022)
6. Технологии разработки программного обеспечения: Учебник/ С. Орлов. — СПб.:Питер, 2002. — 464 с.: ил. ISBN 5-94723-145-X