|  |  |
| --- | --- |
|  | **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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа №6**

**По курсу**

**«Технологии разработки программного обеспечения»**

***«Прямое и обратное проектирование»***

Студент \_\_ИУ5-12М\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_Пермяков Д.К.\_\_\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Проверила **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Виноградова М.В.\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2025 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.

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

**Подготовка проекта**

Открыли в среде Enterprise Architect проект, созданный ранее. Установили для проекта профиль языка программирования, который будет использоваться при обозначении типов данных и генерации кода.

**Диаграммы трассировки классов анализа в классы проектирования.**

Создали диаграмму трассировки граничных классов в классы форм (рис 1.1).

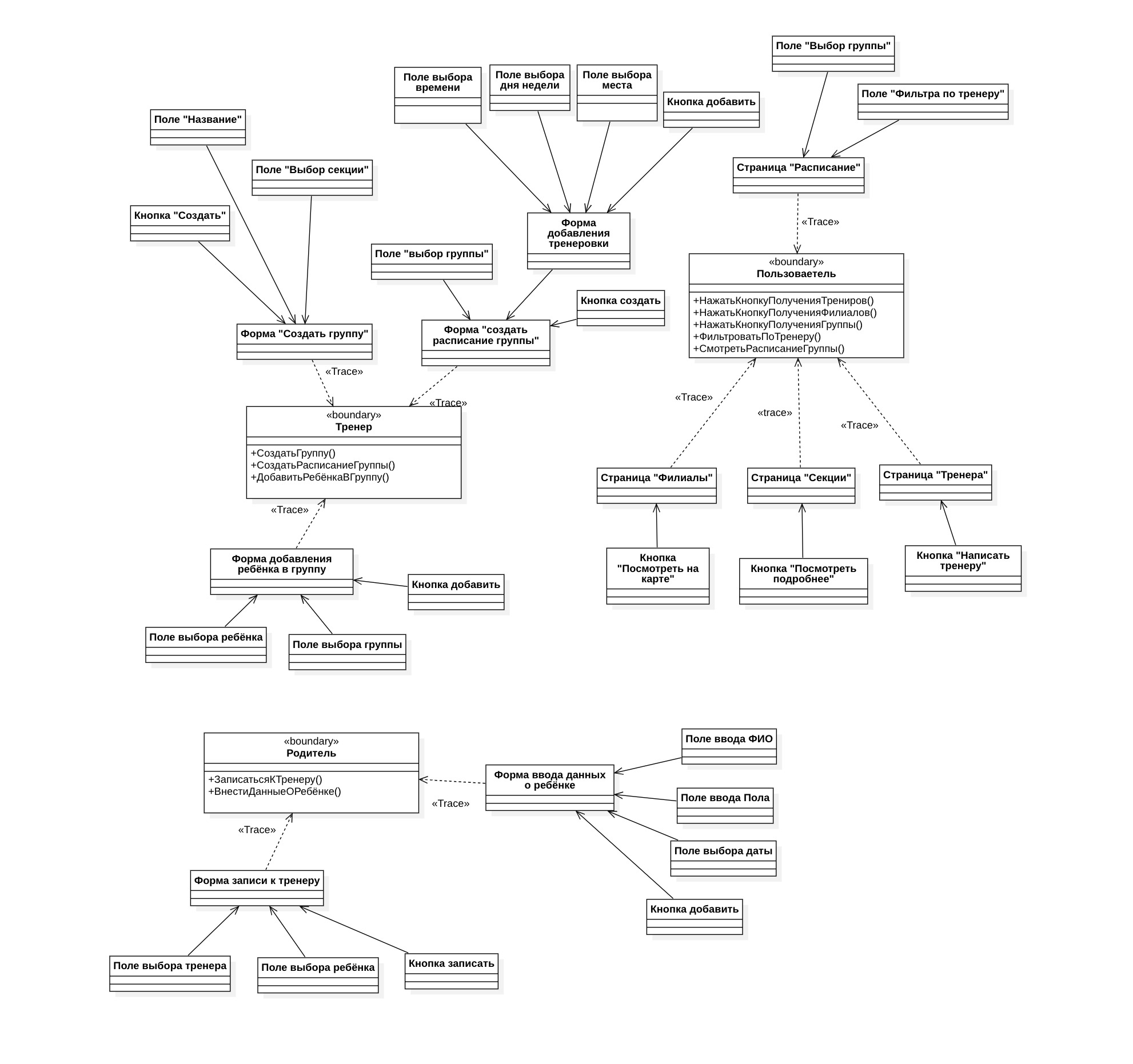


Рис 1.1. Диаграмма трассировки граничных классов в классы форм

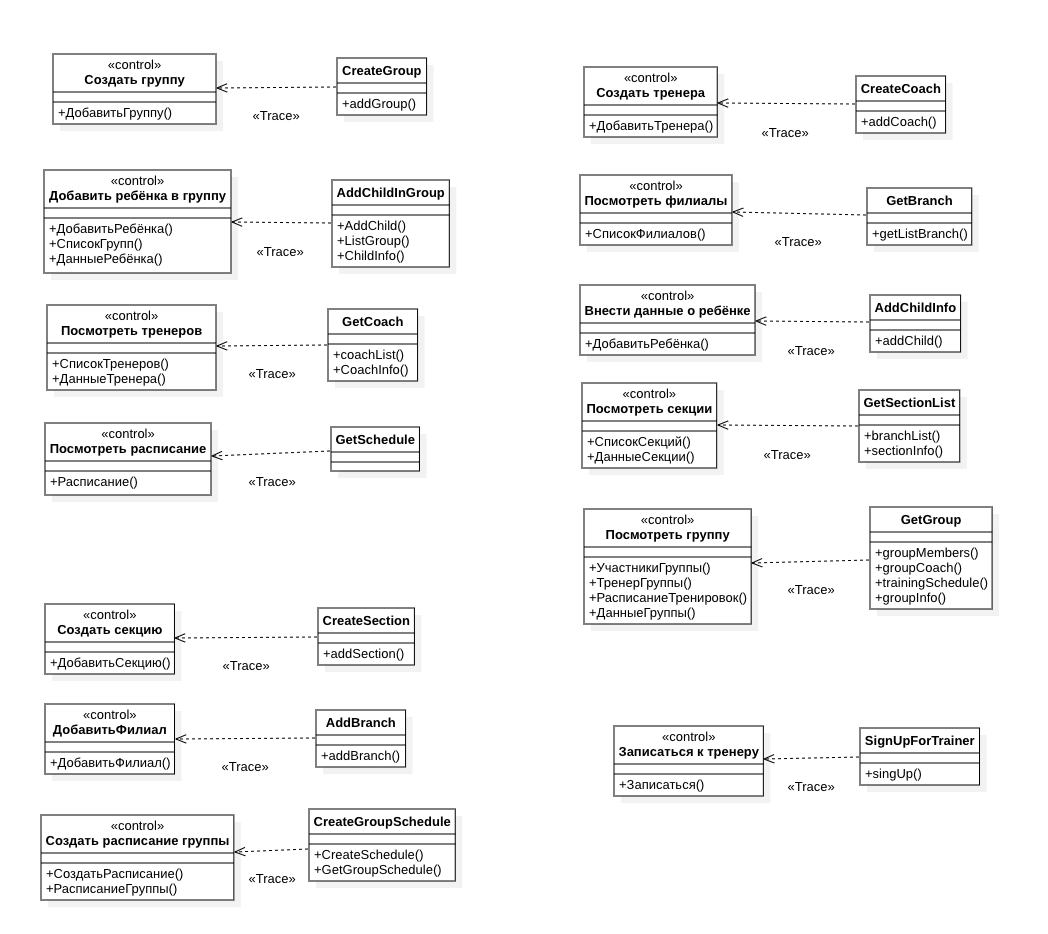


Рис 1.2. Диаграмма трассировки управляющих классов в классы проектирования.

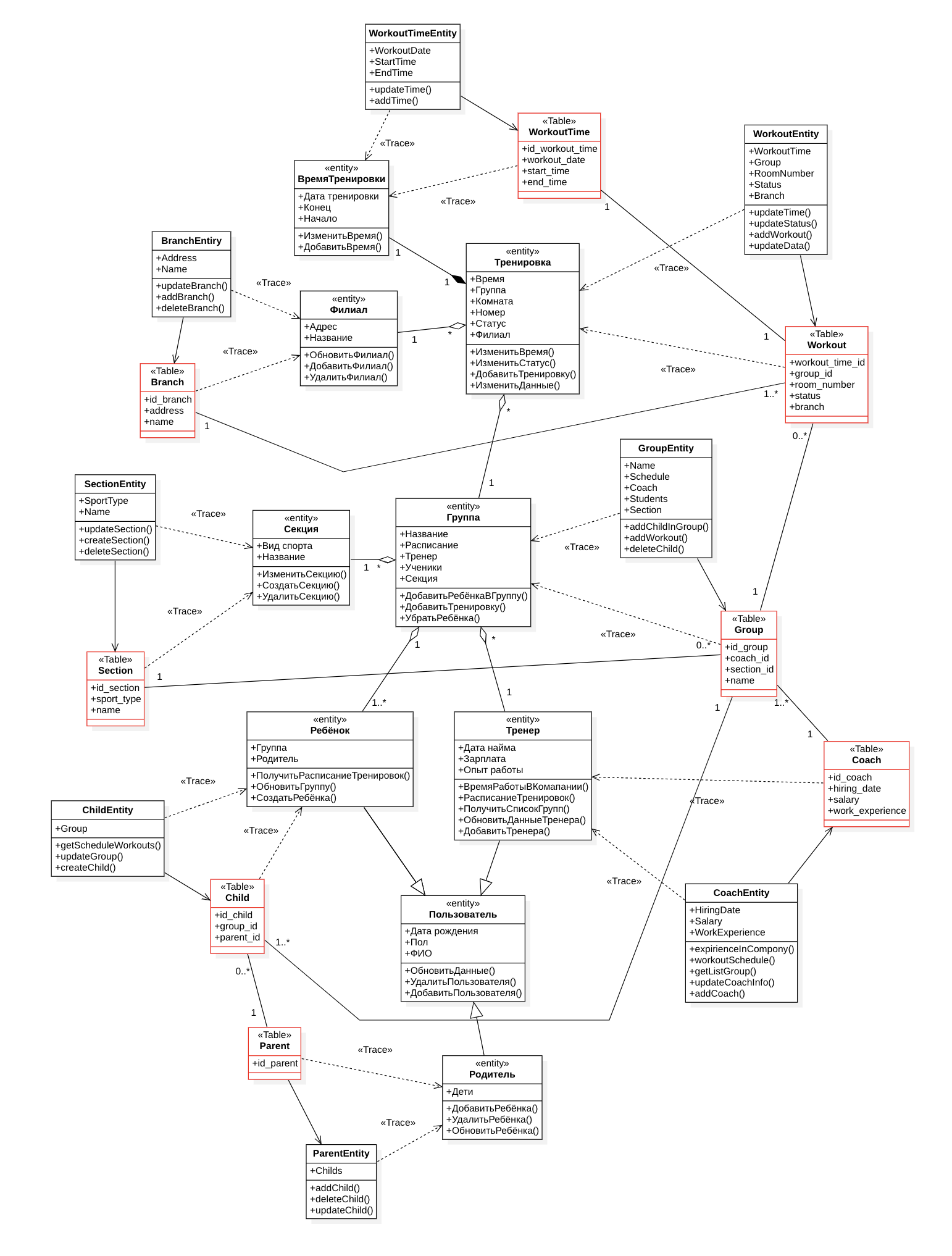
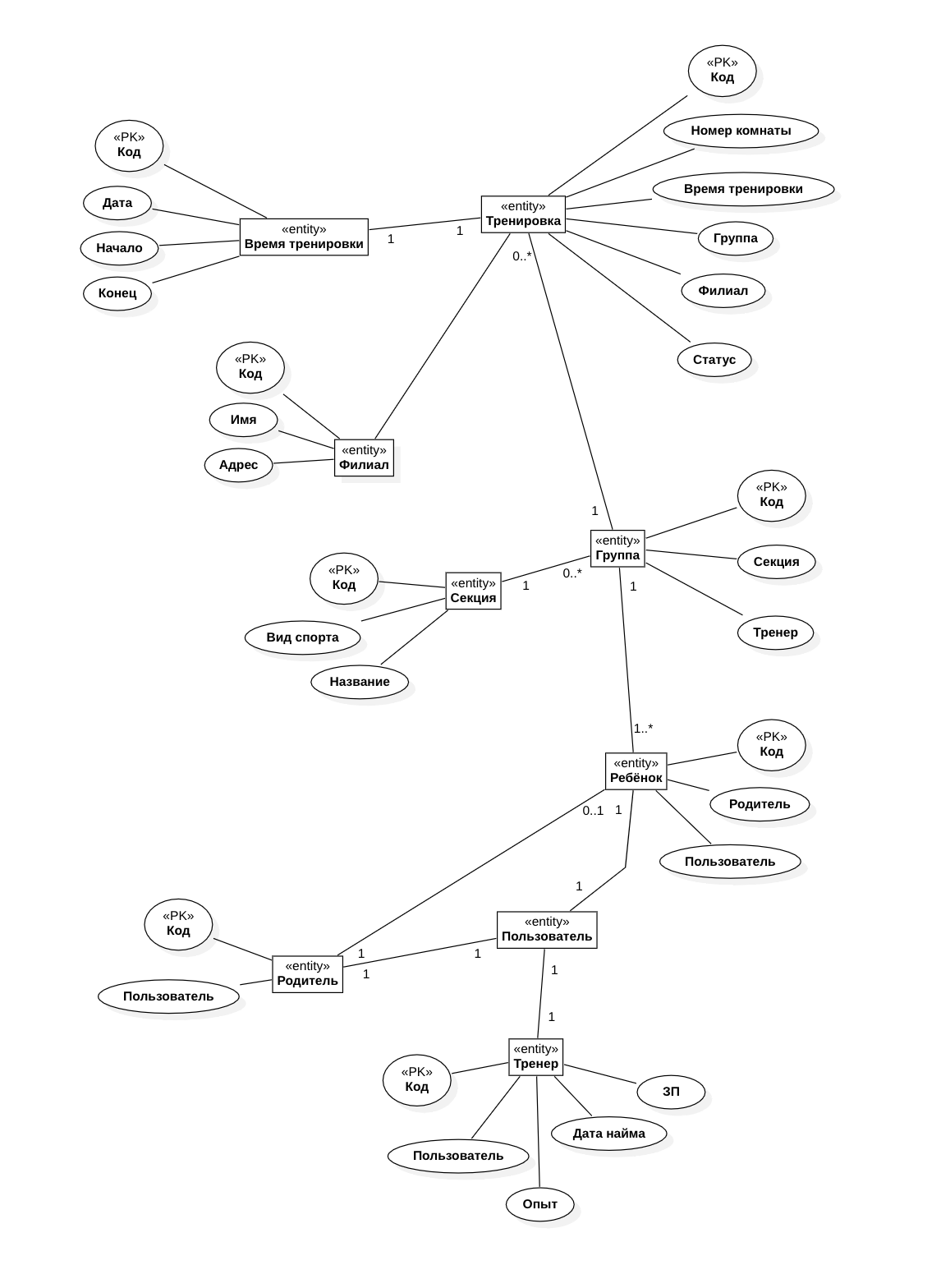


Рис 1.3. Диаграмма трассировки сущностей в сущности проектирования

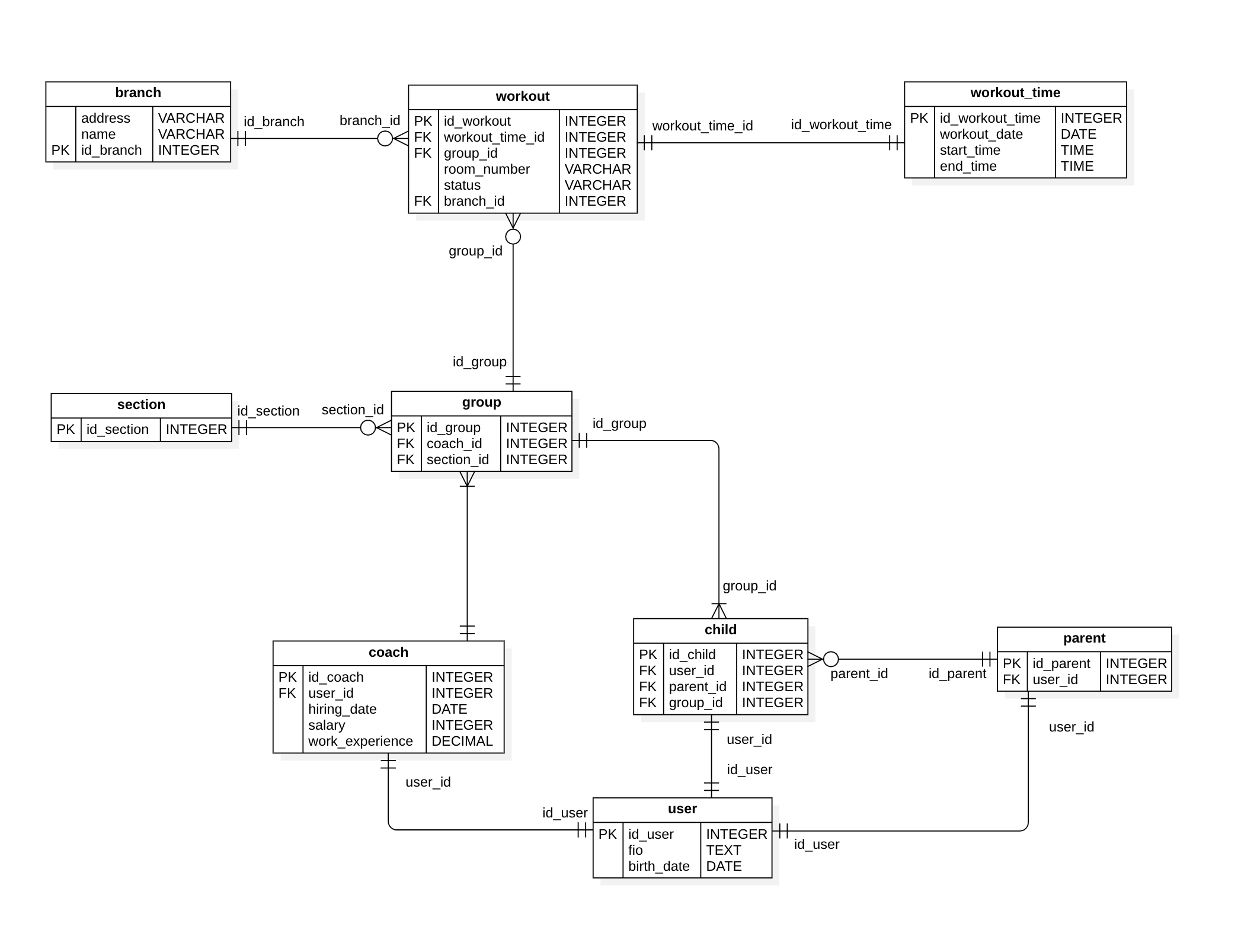
**Создать диаграмму трассировки классов сущностей в сущности инфологической модели.**

****

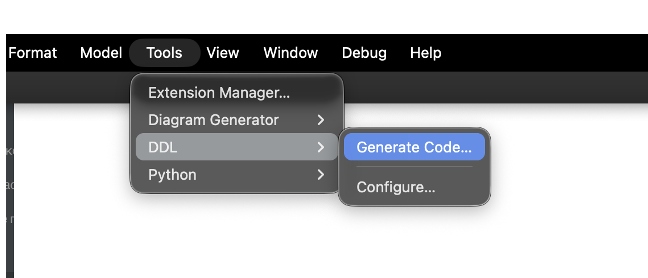
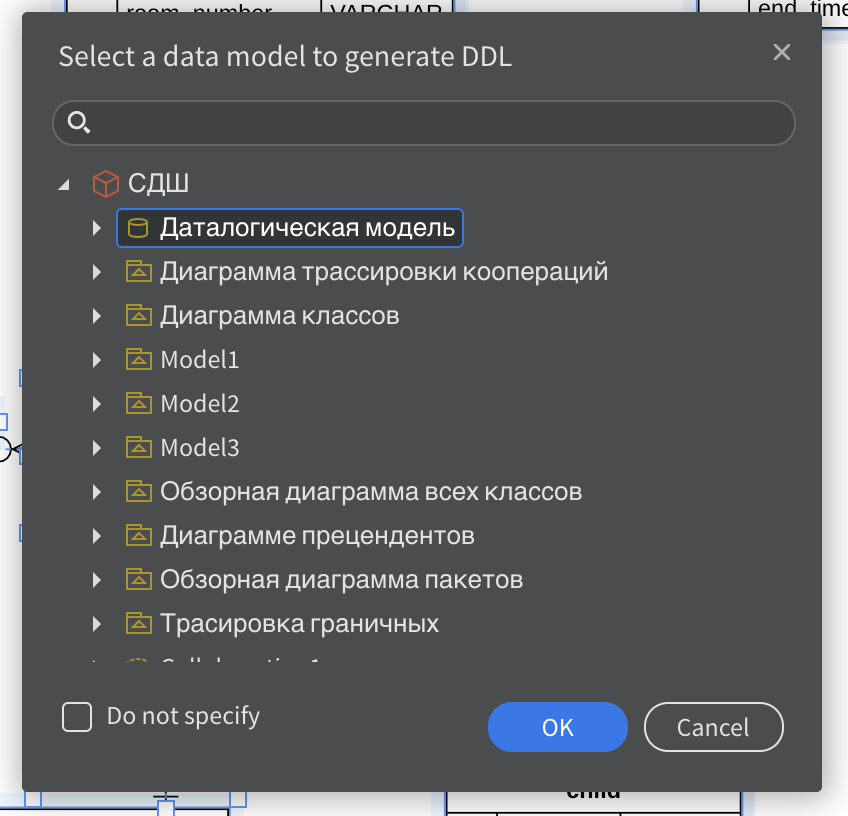
**Создать инфологическую модель**



**Даталогическая модель, сделанная вручную**



Сгенерим DDL

SET FOREIGN\_KEY\_CHECKS = 0;

DROP TABLE IF EXISTS `branch`;

DROP TABLE IF EXISTS `workout`;

DROP TABLE IF EXISTS `workout\_time`;

DROP TABLE IF EXISTS `group`;

DROP TABLE IF EXISTS `section`;

DROP TABLE IF EXISTS `user`;

DROP TABLE IF EXISTS `coach`;

DROP TABLE IF EXISTS `parent`;

DROP TABLE IF EXISTS `child`;

SET FOREIGN\_KEY\_CHECKS = 1;

CREATE TABLE `branch` (

`address` VARCHAR NOT NULL,

`name` VARCHAR NOT NULL,

`id\_branch` INTEGER NOT NULL,

PRIMARY KEY (`id\_branch`)

);

CREATE TABLE `workout` (

`id\_workout` INTEGER NOT NULL,

`workout\_time\_id` INTEGER NOT NULL,

`group\_id` INTEGER NOT NULL,

`room\_number` VARCHAR NOT NULL,

`status` VARCHAR NOT NULL,

`branch\_id` INTEGER NOT NULL,

PRIMARY KEY (`id\_workout`)

);

CREATE TABLE `workout\_time` (

`id\_workout\_time` INTEGER NOT NULL,

`workout\_date` DATE NOT NULL,

`start\_time` TIME NOT NULL,

`end\_time` TIME NOT NULL,

PRIMARY KEY (`id\_workout\_time`)

);

CREATE TABLE `group` (

`id\_group` INTEGER NOT NULL,

`coach\_id` INTEGER NOT NULL,

`section\_id` INTEGER NOT NULL,

PRIMARY KEY (`id\_group`)

);

CREATE TABLE `section` (

`id\_section` INTEGER NOT NULL,

PRIMARY KEY (`id\_section`)

);

CREATE TABLE `user` (

`id\_user` INTEGER NOT NULL,

`fio` TEXT NOT NULL,

`birth\_date` DATE NOT NULL,

PRIMARY KEY (`id\_user`)

);

CREATE TABLE `coach` (

`id\_coach` INTEGER NOT NULL,

`user\_id` INTEGER NOT NULL,

`hiring\_date` DATE NOT NULL,

`salary` INTEGER NOT NULL,

`work\_experience` DECIMAL NOT NULL,

PRIMARY KEY (`id\_coach`)

);

CREATE TABLE `parent` (

`id\_parent` INTEGER NOT NULL,

`user\_id` INTEGER NOT NULL,

PRIMARY KEY (`id\_parent`)

);

CREATE TABLE `child` (

`id\_child` INTEGER NOT NULL,

`user\_id` INTEGER NOT NULL,

`parent\_id` INTEGER NOT NULL,

PRIMARY KEY (`id\_child`)

);

ALTER TABLE `workout` ADD FOREIGN KEY (`workout\_time\_id`) REFERENCES `workout\_time`(`id\_workout\_time`);

ALTER TABLE `workout` ADD FOREIGN KEY (`group\_id`) REFERENCES `group`(`id\_group`);

ALTER TABLE `workout` ADD FOREIGN KEY (`branch\_id`) REFERENCES `branch`(`id\_branch`);

ALTER TABLE `group` ADD FOREIGN KEY (`coach\_id`) REFERENCES `coach`(`id\_coach`);

ALTER TABLE `group` ADD FOREIGN KEY (`section\_id`) REFERENCES `section`(`id\_section`);

ALTER TABLE `coach` ADD FOREIGN KEY (`user\_id`) REFERENCES `user`(`id\_user`);

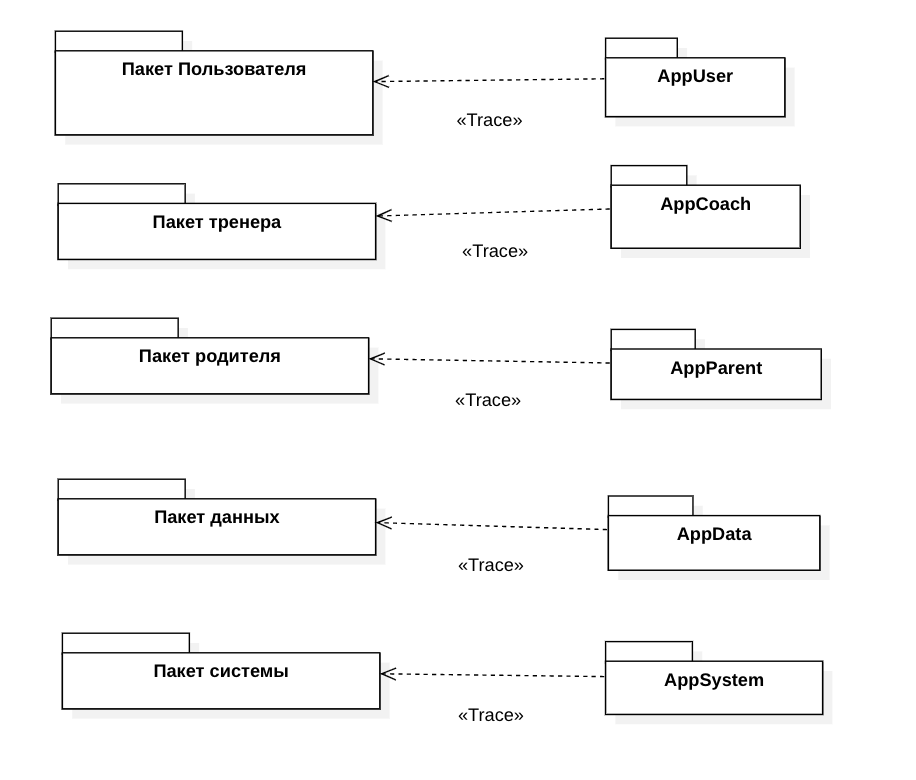
ALTER TABLE `parent` ADD FOREIGN KEY (`user\_id`) REFERENCES `user`(`id\_user`);

ALTER TABLE `child` ADD FOREIGN KEY (`user\_id`) REFERENCES `user`(`id\_user`);

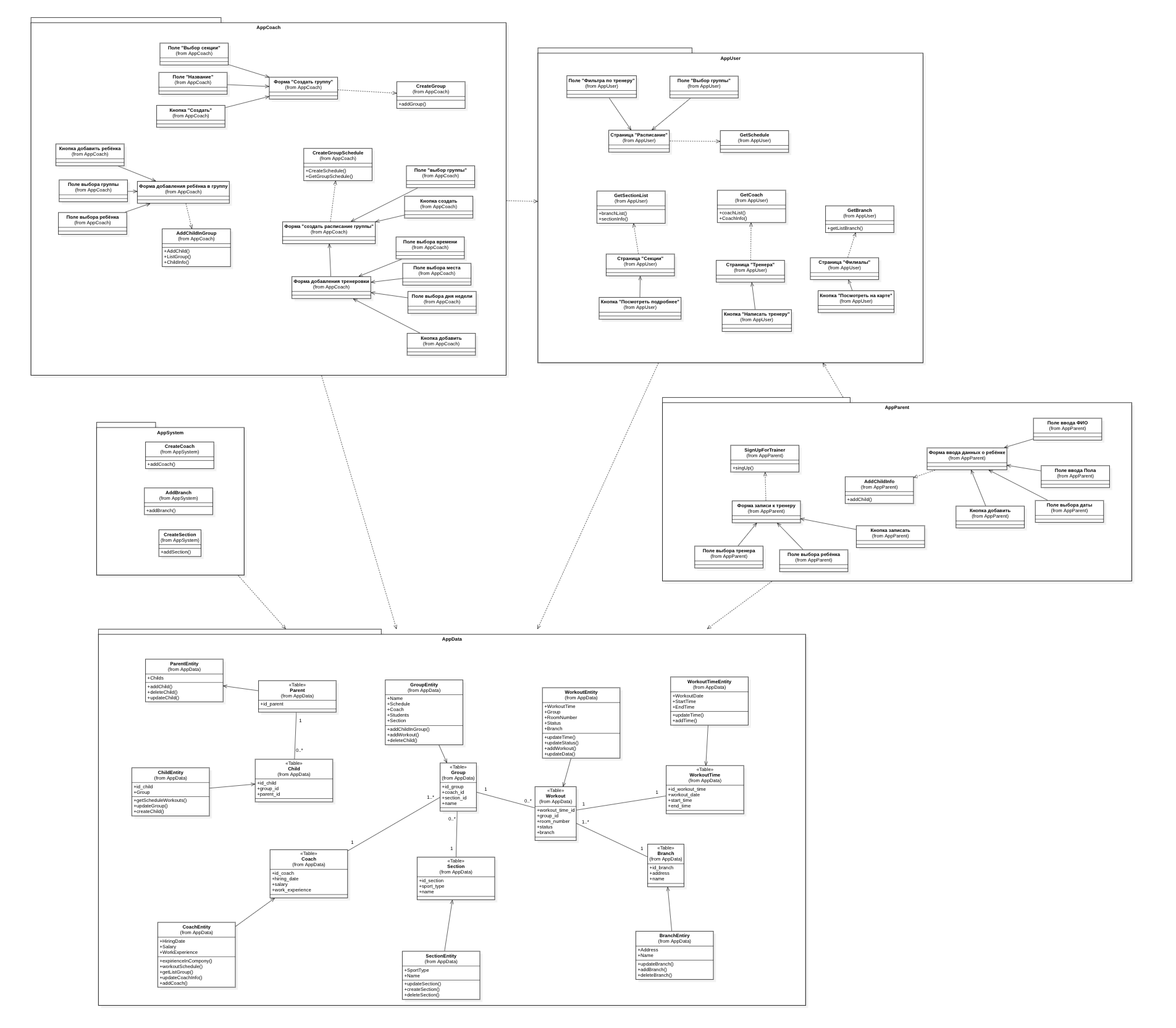
ALTER TABLE `child` ADD FOREIGN KEY (`parent\_id`) REFERENCES `parent`(`id\_parent`);

ALTER TABLE `child` ADD FOREIGN KEY (`group\_id`) REFERENCES `group`(`id\_group`);

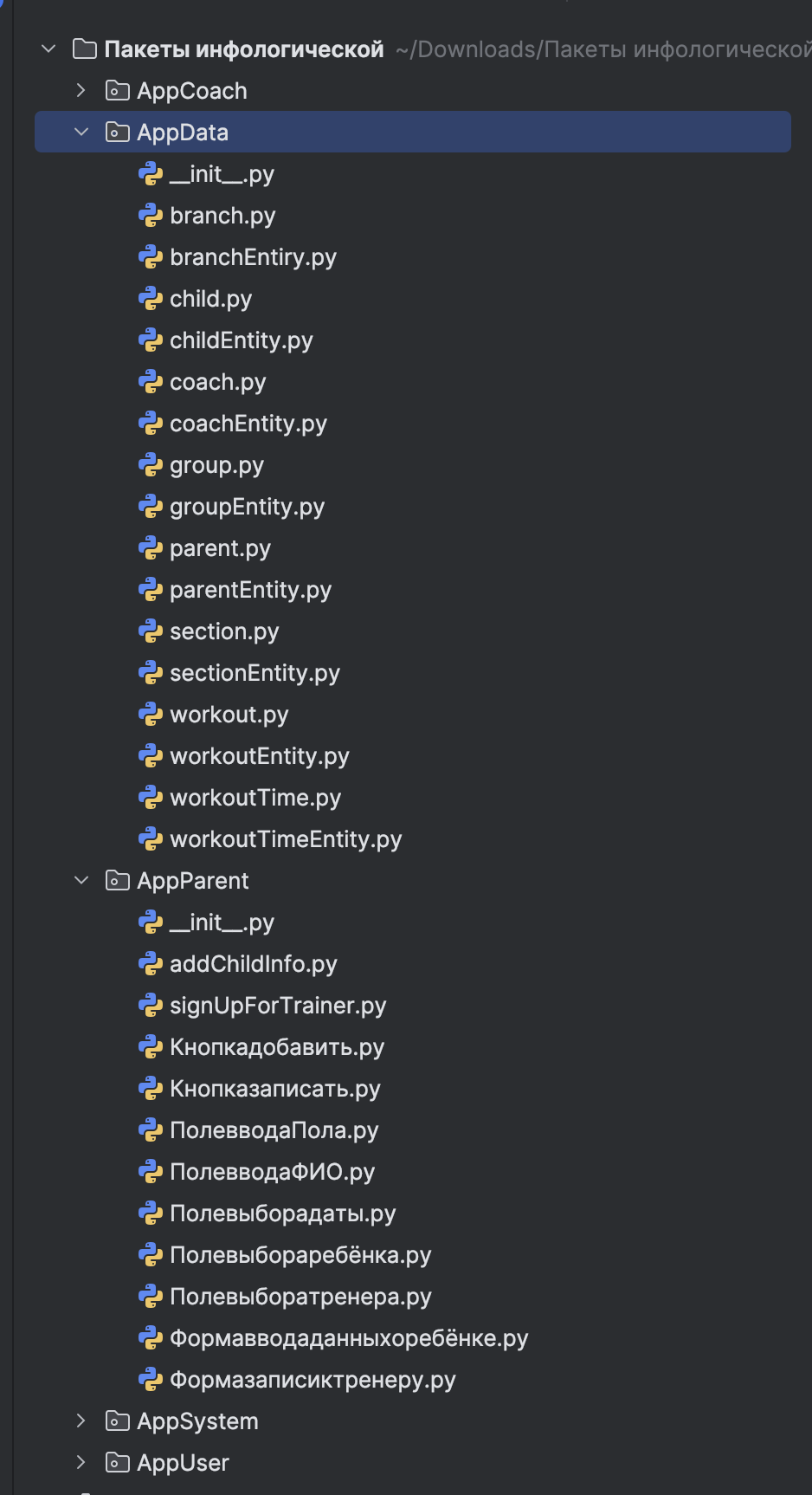
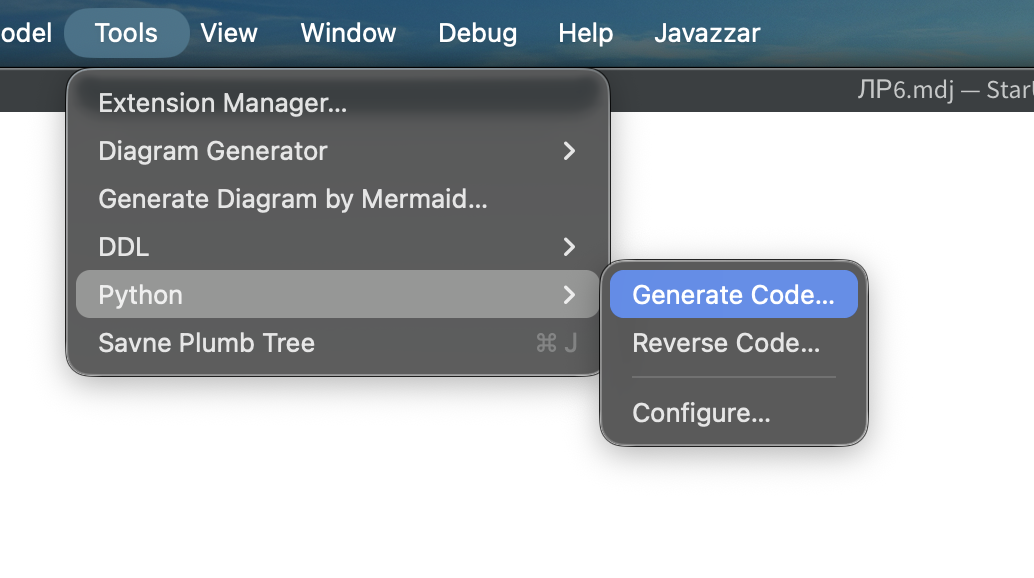
**Создать диаграмму трассировки пакетов анализа в подсистемы**



**Создать подсистемы (package) и переместить туда полученные классы. В подсистему данных внести сущности инфологической модели.**

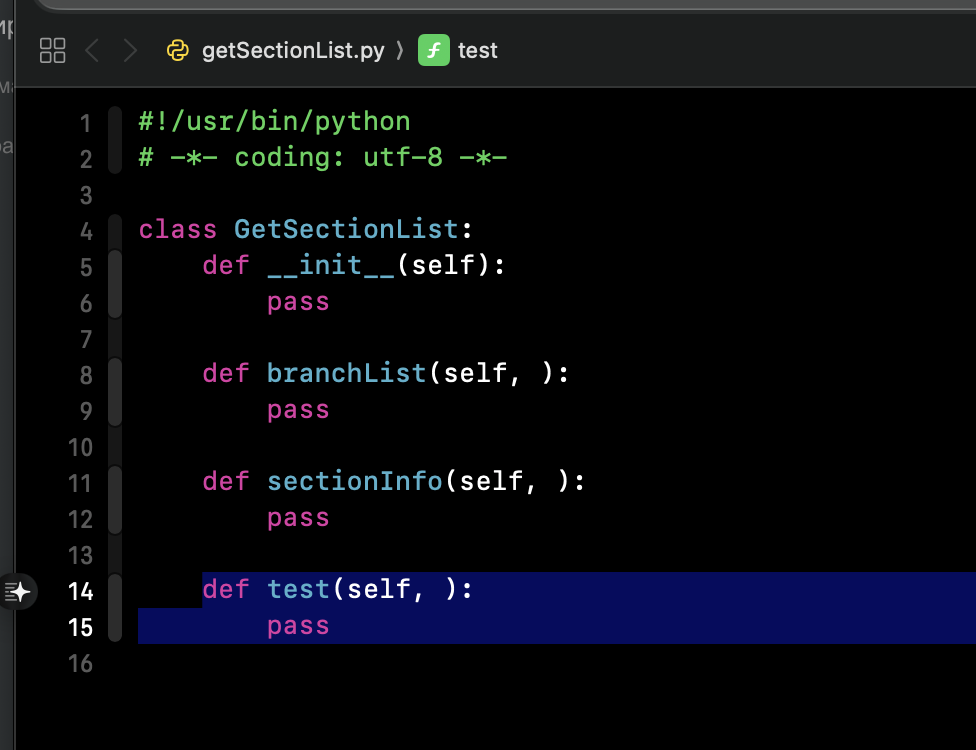
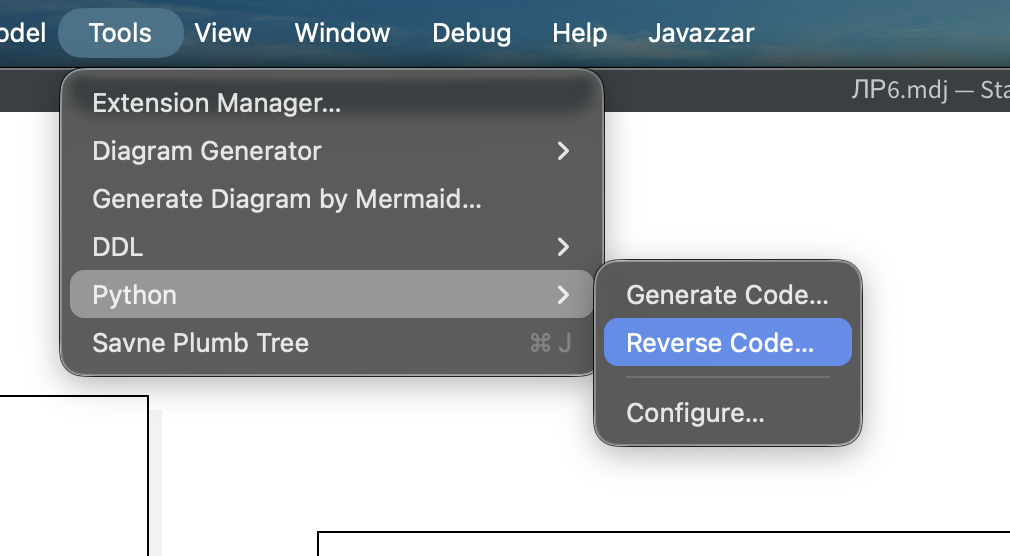


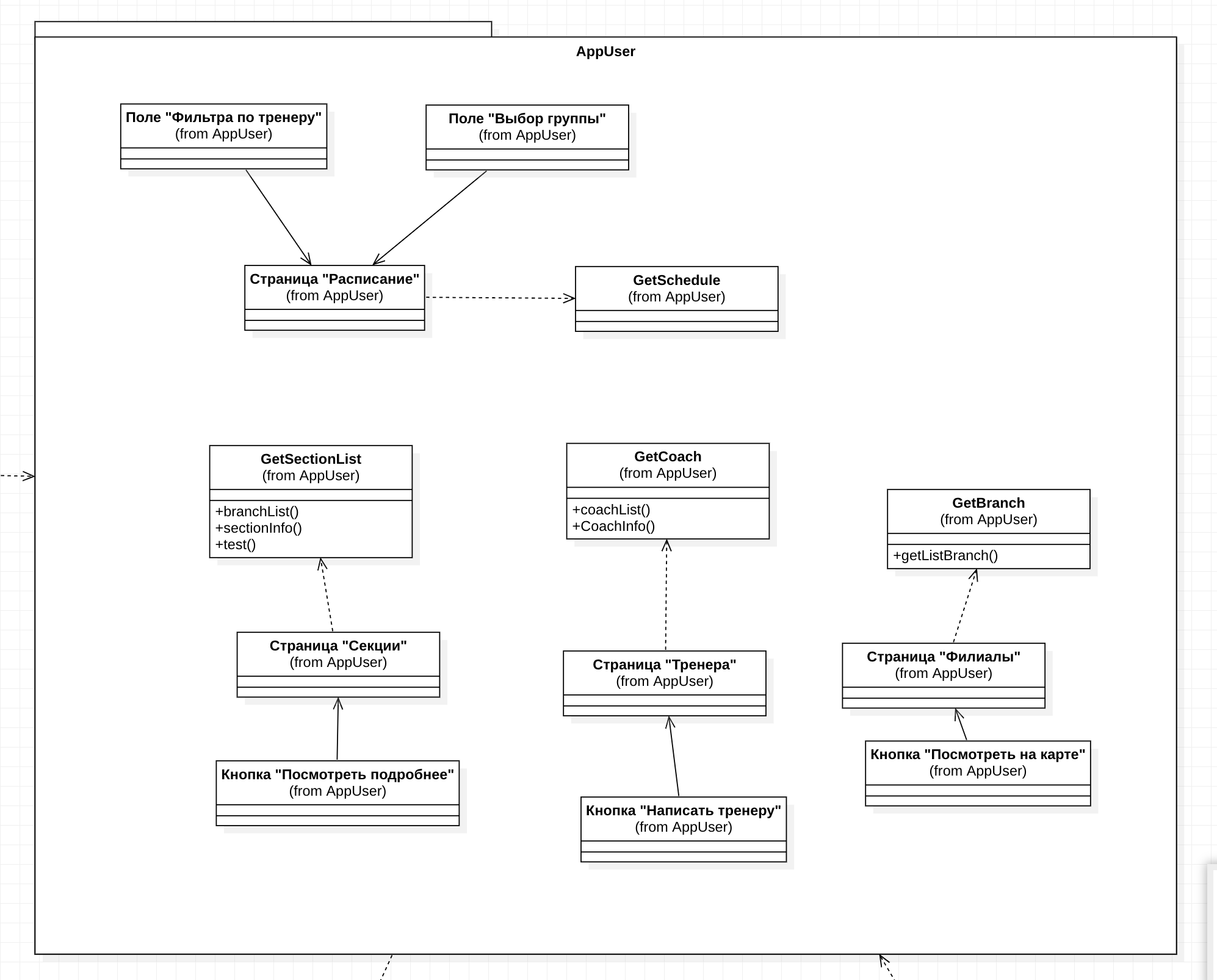
**Выполнить прямое и обратное проектирование package подсистемы (сгенерировать код, изменить код, синхронизировать).**

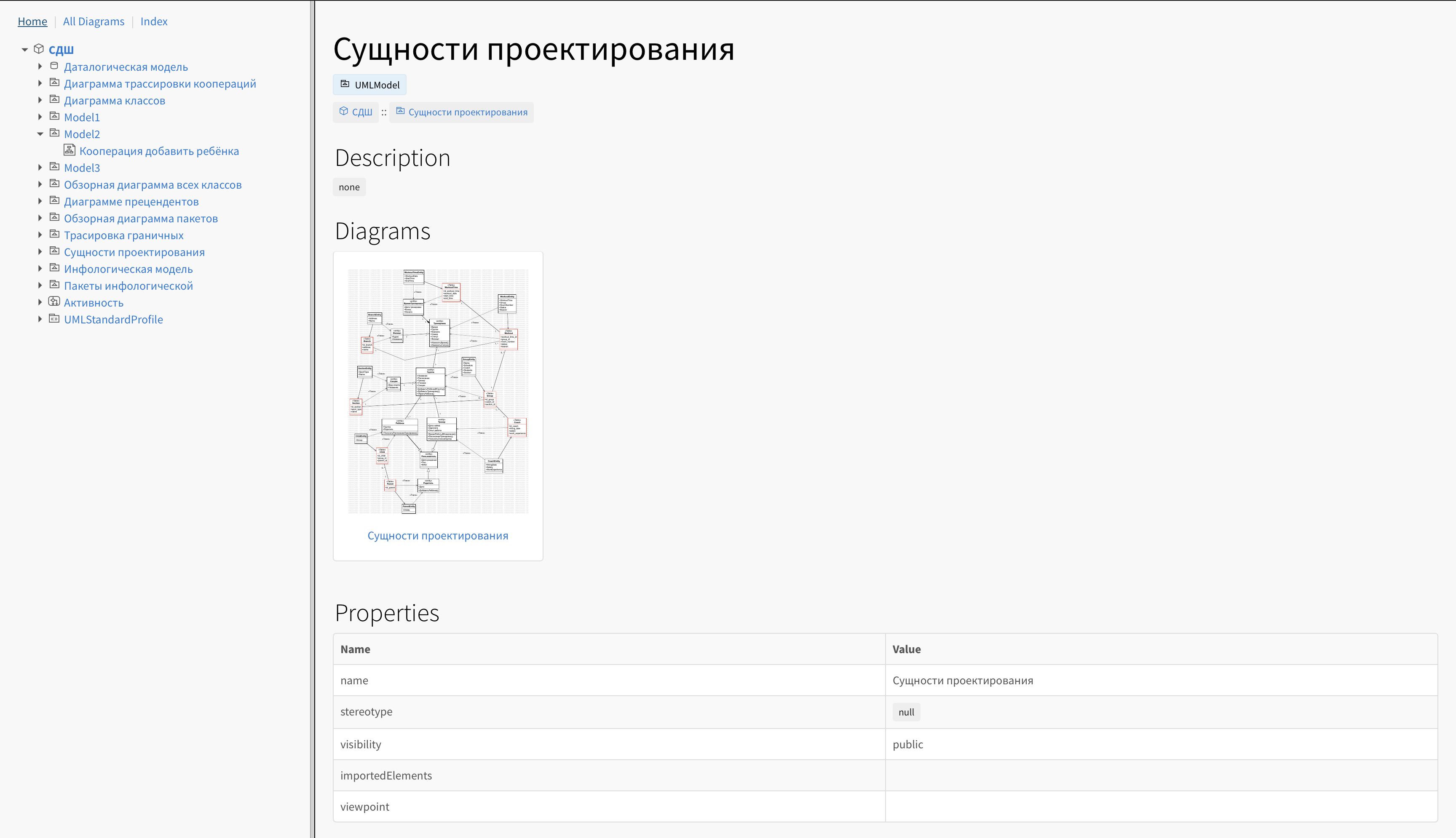


Откроем файл

Сгенерировали документацию проекта и отредактируем

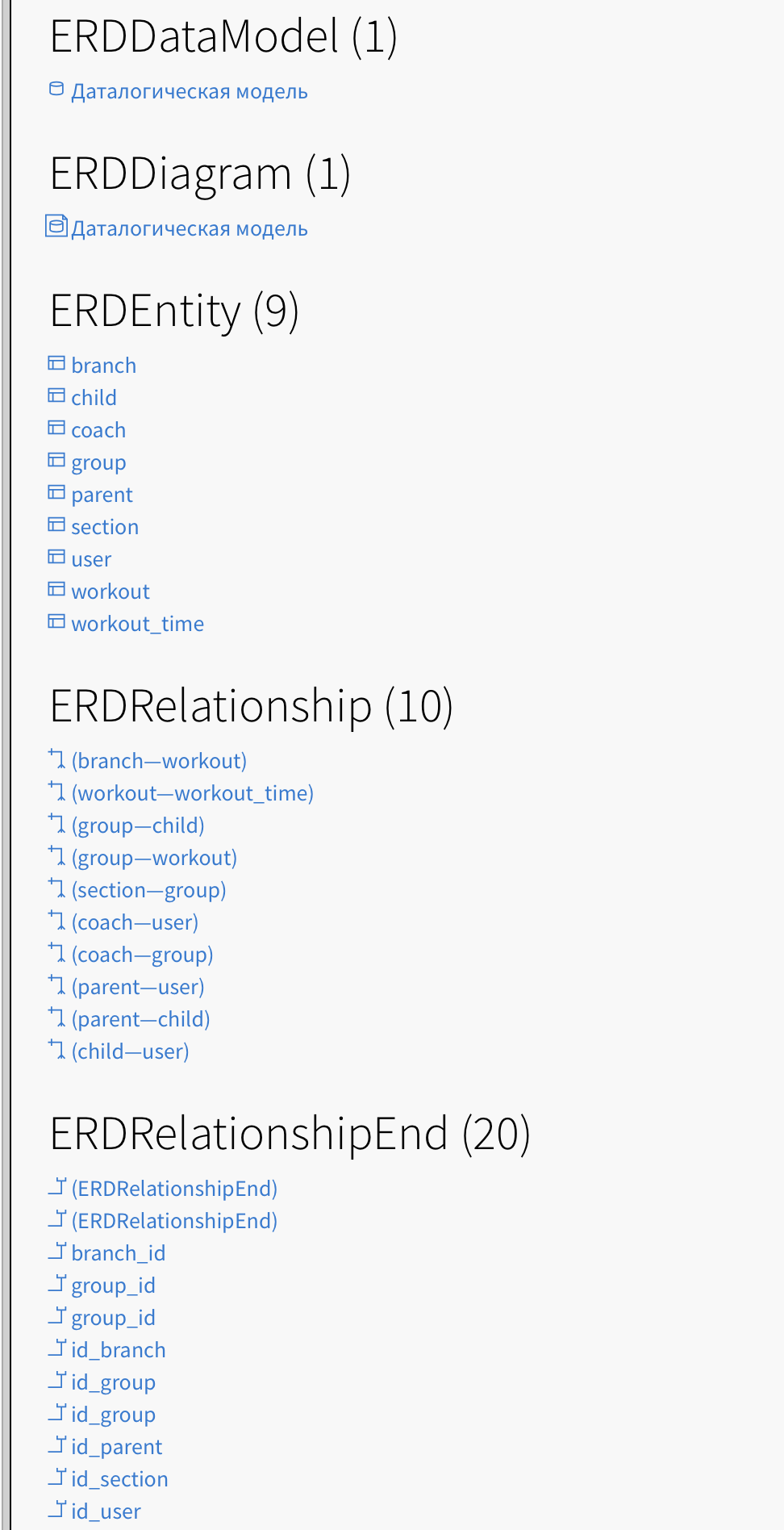




**Выполнить проверку целостности проекта**



**Получить краткую статистику проекта.**



### Вывод.

Мы изучили унифицированный процесс разработки (RUP), приобрели умения построения модели проектирования и процессов прямого и обратного проектирования, получили навыки построения модели проектирования и процессов прямого и обратного проектирования в среде Sparx Enterprise Architect.

### Список литературы.

1. Виноградова М.В., Белоусова В.И. Унифицированный процесс разработки программного обеспечения: учебное пособие / Виноградова М.В., Белоусова В.И. – М.: МГТУ им.Н.Э. Баумана. – 2015 г. – 82 с. - Режим доступа: http://ebooks.bmstu.ru/catalog/193/book1303.html (дата обращения: 17.12.2017). — ISBN: 978-5-7038-4265-2
2. Орлов С.А., Цилькер Б.Я. Технологии разработки программного обеспечения, 4-е изд. — СПб.: Питер, 2012 г. — 608 с. — ISBN: 9785459011012