

# Лабораторная работа. UML моделирование

## Предусловия

1. Установите Visual Paradigm ([Download Visual Paradigm. Try it FREE \(visual-paradigm.com\)](https://www.visual-paradigm.com/))
2. Ознакомьтесь с инструкциями:
  - [Use Case Diagram Tutorial \(visual-paradigm.com\)](https://www.visual-paradigm.com/tutorials/use-case-diagram-tutorial.html)
  - [Class Diagram Tutorial \(visual-paradigm.com\)](https://www.visual-paradigm.com/tutorials/class-diagram-tutorial.html)
  - [Sequence Diagram Tutorial \(visual-paradigm.com\)](https://www.visual-paradigm.com/tutorials/sequence-diagram-tutorial.html)
  - [Activity Diagram Tutorial \(visual-paradigm.com\)](https://www.visual-paradigm.com/tutorials/activity-diagram-tutorial.html)
  - [State Machine Diagram Tutorial \(visual-paradigm.com\)](https://www.visual-paradigm.com/tutorials/state-machine-diagram-tutorial.html)
  - [UML/Code Generation Software \(visual-paradigm.com\)](https://www.visual-paradigm.com/tutorials/uml-code-generation-software.html)
    - [How to Use Java Round-Trip Engineering? \(visual-paradigm.com\)](https://www.visual-paradigm.com/tutorials/how-to-use-java-round-trip-engineering.html)
    - [How to Use C++ Round-Trip Engineering? \(visual-paradigm.com\)](https://www.visual-paradigm.com/tutorials/how-to-use-cplusplus-round-trip-engineering.html)
    - [How to Generate Code from State Machine Diagram? \(visual-paradigm.com\)](https://www.visual-paradigm.com/tutorials/how-to-generate-code-from-state-machine-diagram.html)
    - [How to Generate Sequence Diagram from Java? \(visual-paradigm.com\)](https://www.visual-paradigm.com/tutorials/how-to-generate-sequence-diagram-from-java.html)
    - [C# Round-trip Engineering \(visual-paradigm.com\)](https://www.visual-paradigm.com/tutorials/csharp-round-trip-engineering.html)

**Замечание:** вы можете использовать любое другое средство UML моделирования, допускающее генерацию кода и выполнение модели.

**Задание:** постройте модель следующей системы, выполнив все шаги.

«Университетская информационная система»

Университет (**university**) состоит из множества факультетов (**faculty**), которые в свою очередь разделены на кафедры (**institute**). Каждая кафедра и факультет имеют название (**name**). Для каждой кафедры известен адрес (**address**).

Руководителем каждого факультета является декан (**dean**), который является сотрудником (**employee**) университета.

Общее число сотрудников (**number of employees**) известно. Сотрудники имеют социальный код (**social security number**), имя (**name**) и электронный адрес (**e-mail**). Существует различие между научным (**research**) и административным персоналом (**administrative personal**).

Каждый научный сотрудник прикреплен по крайней мере к одной из кафедр. Область исследований для каждого из них известна. Кроме того, каждый научный сотрудник может быть вовлечен в проекты (**project**) на определенное количество часов (**hours**), с названием проекта (**name**), датой начала (**starting date**) и окончания (**end date**) известных заранее. Некоторые научные сотрудники могут вести учебные курсы (**course**). Тогда они называются преподавателями (**lecturer**).

Каждый курс имеет уникальный номер (**unique number**) (ID), название (**name**) и количество часов в неделю (**weekly duration**).

Шаг 1: Идентификация классов : проводим текстовый анализ и выделяем существительные — кандидаты на классы (**red**).

Шаг 2: Идентификация атрибутов: проводим текстовый анализ и выделяем существительные — кандидаты на атрибуты (**green**) классов.

Шаг 3: Идентификация отношений

Существует четыре типа отношений:

Ассоциация - Association

Обобщение - Generalization

Агрегация — Aggregation

Композиция - Composition

- *Индикаторы обобщения:*

1. «Существует различие между научным (**research**) и административным персоналом (**administrative personal**).»
2. «Некоторые научные сотрудники могут вести учебные курсы (**course**). Тогда они называются преподавателями (**lecturer**).»

- *Идентифицируйте типы отношений для следующих предложений*

1. “Университет (**university**) состоит из множества факультетов (**faculty**), которые в свою очередь разделены на кафедры (**institute**).”
2. “Руководителем каждого факультета является декан (**dean**), который является сотрудником (**employee**) университета.”
3. “Каждый научный сотрудник прикреплен по крайней мере к одной из кафедр.”
4. “Кроме того, каждый научный сотрудник может быть вовлечен в проекты (**project**) на определенное количество часов ...”

5. “Некоторые научные сотрудники могут вести учебные курсы (**course**). Тогда они называются преподавателями (**lecturer**).”

Шаг 4. Идентифицируйте методы классов.

Шаг 5. Сгенерируйте код на каком-либо объектно-ориентированном языке программирования и добавьте ввод/вывод данных.

Шаг 6. Компилируйте и выполните код.

Шаг 7. Примените обратный инжинеринг и сгенерируйте модель (диаграмму классов) в точности соответствующую вашему коду.

Шаг 8. Конкретизируйте модель, добавив следующие поведенческие диаграммы:

- диаграмма прецедентов
- диаграмма последовательностей
- диаграмма состояний (или диаграмма активностей)

Шаг 7. Представьте отчёт (pdf файл), включающий окончательную модель и код, а также скриншоты результатов компиляции и выполнения.

### Приложение: *Class Diagram*

