



Разрабатываем систему управления университетом.

Сущности:

- студенты **Postgraduate** и **Undergraduate**;
- преподаватели **Academic**, которые могут быть руководителями **Postgraduate** или преподавателями **Undergraduate**.

Класс **Student** реализует интерфейс **Notifiable**, который предоставляет метод **notify()** – для уведомления студентов о важных событиях по электронной почте.

В соответствии с диаграммой классов реализовать:

1. Класс **Postgraduate**.
2. Класс **Course**.
3. Класс **Notifier**. Поле класса **notifiables** хранит множество объектов, подлежащих уведомлению. Параметр конструктора инициализирует поле **notifiables**. Класс предоставляет метод **doNotifyAll()**, который вызывает **notify()** для всех объектов множества **notifiables**.
4. Класс **ProgrammingTest** с методом **main()**, который

- Создает **Course** с двумя **Undergraduate** и тремя **Postgraduate** на основе информации, приведенной в таблице ниже. Добавьте любую информацию, которая отсутствует в таблице.
- Выводит всех **Postgraduate** курса, у которых руководителем является Cay Horstmann.
- Использует **Notifier**, чтобы уведомить со следующим сообщением "Hello, Students!"

Login	Type	Supervisor/Tutor
gg4	Undergraduate	Paul Deitel
pr3	Undergraduate	Cay Horstmann
te2	Postgraduate	Paul Deitel
yj34	Postgraduate	Cay Horstmann
jj8	Postgraduate	Cay Horstmann

5. Продвигаем класс **Course**. Добавить:

- метод добавления студента;
- итератор для перебора студентов;
- метод, который возвращает множество студентов, удовлетворяющих предикату;
- продемонстрировать в методе **main()**.

6. Класс **Writer**, а в него – метод для сохранения Курса в текстовый файл. Использовать оператор **try with resources**. Продемонстрировать в методе **main()**.

7. Оконный интерфейс. Классы **Frame** и **CoursePanel** – для отображения данных о курсе.