Лабораторная работа. Классы. Интерфейсы

Задание: Предметная область - программное обеспечение для университета

1 Класс Student

Создайте публичный класс Student.

1) В классе должны быть следующие ***поля***:

- имя;
- фамилия;
- год поступления;
- уникальный шестизначный номер зачетной книжки.

2) В классе должны быть следующие ***конструкторы***:

а) С двумя параметрами: имя и фамилия. В номер зачетной книжки записывается 0;

б) С тремя параметрами: имя, фамилия, номер зачетной книжки.

3) В классе должны быть следующие ***методы***:

а) геттер, возвращающий имя студента;

б) сеттер, изменяющий имя студента;

в) геттер, возвращающий фамилию студента;

г) сеттер, изменяющий фамилию студента;

д) геттер, возвращающий номер зачетной книжки;

е) сеттер, изменяющий номер зачетной книжки;

ж) геттер, возвращающий год поступления;

з) сеттер, изменяющий год поступления.

2 Класс Group

Создайте публичный класс Group – студенческой группы.

1) В классе должны быть следующие ***поля***:

- уникальный номер (в пределах специальности);
- массив студентов.

2) В классе должны быть следующие ***конструкторы***:

а) С одним параметром: номер группы (в этом случае, количество студентов записывается как 0);

б) С двумя параметрами: номер группы, количество студентов (инициализация массива, но его элементы остаются пустыми);

в) принимает на вход массив студентов.

3) В классе должны быть следующие ***методы***:

а) геттер, возвращающий номер группы;

б) сеттер, изменяющий номер группы;

в) геттер, возвращающий общее число студентов группы;

г) метод, возвращающий ссылку на студента по номеру зачетной книжки;

д) метод, удаляющий студента из группы по номеру зачетной книжки (помните про корректное удаление элемента из массива);

е) метод, добавляющий нового студента в группу (принимает на вход ссылку на объект Student, если массив уже полностью заполнен - реализуйте расширение массива);

ж) метод, возвращающий массив студентов;

з) метод, возвращающий массив студентов, отсортированный по фамилиям.

3 Класс Payment

Создайте класс Payment - платы за обучение.

1) В классе должны быть следующие ***поля***:

- дата (экземпляр класса java.time);
- размер суммы, переведенной студентом на счет университета.

2) В классе должны быть следующие ***конструкторы***:

а) без параметров (дата содержит null, сумма - 0);

б) с двумя параметрами - датой оплаты и суммой.

3) В классе должны быть следующие ***методы***: геттеры и сеттеры для приватных полей.

4 Класс ContractStudent

Создайте класс ContractStudent, расширяющий класс Student.

Данный класс добавляет следующие члены:

1) Поля:

- приватное поле - массив платежей;
- приватное поле - стоимость обучения за семестр (предполагается, что эта стоимость в течение всего периода обучения меняться не будет).

2) Конструкторы:

а) с двумя параметрами: имя и фамилия студента (номер зачетной книжки - 0, создает массив платежей нулевой длины);

б) с тремя параметрами: имя, фамилия, номер книжки (создает массив платежей нулевой длины);

3) Методы:

а) геттер для массива платежей и стоимости обучения за семестр;

б) метод, возвращающий размер задолженности студента на текущий момент. Исходите из того, что плата за обучение не меняется. Зная текущую дату и год поступления, можно выяснить, сколько семестров студент проучился и определить сумму, которую он должен внести за проведенные в университете семестры. Сравните эту сумму с тем, что студент оплатил по факту;

в) метод, добавляющий новый платеж в конец списка платежей;

5 Интерфейс Event

Создайте интерфейс Event - мероприятия, в котором принимает участие студент.

Интерфейс определяет следующие ***методы***:

а) геттеры и сеттеры для даты проведения мероприятия (методы работают с объектами типа java.util.Date);

б) геттеры и сеттеры для названия города, в котором проводилось мероприятие.

6 Класс Olympics

Определите класс Olympics - студенческие олимпиады. Класс реализующий интерфейс Event.

1) В классе должны быть следующие ***поля***:
   - дата проведения олимпиады;
   - название города, где проходила олимпиада;
   - место, которое занял студент на олимпиаде.
2) В классе должны быть следующие ***конструкторы***:

а) конструктор с тремя параметрами (дата, город, место).

3) В классе должны быть следующие ***методы***:

а) геттеры и сеттеры для приватных полей.

7 Класс Conference

Определите класс Conference - студенческие конференции. Класс должен реализовывать интерфейс Event.

1) В классе должны быть следующие ***поля***:
   - дата проведения конференции;
   - город, где проходила конференция;
   - название доклада, с которым студент выступал на конференции.
2) В классе должны быть следующие ***конструкторы***:

а) конструктор с тремя параметрами (дата, город, название доклада).

3) В классе должны быть следующие ***методы***:

а) геттеры и сеттеры для приватных полей.

8 Класс Competition

Определите класс Competition - студенческие соревнования. Класс должен реализовывать интерфейс Event.

1) В классе должны быть следующие ***поля***:

– дата проведения соревнования;
– город, где проходило соревнование;
– название студенческого проекта;
– выигранная сумма (0, если студент ничего не выиграл).

2) В классе должны быть следующие ***конструкторы***:

а) конструктор с четырьмя параметрами (дата, город, название проекта, сумма).

3) В классе должны быть следующие ***методы***:

а) геттеры и сеттеры для приватных полей.

9 Интерфейс Activist

Определите интерфейс Activist - участники различных конкурсов, олимпиад и так далее.

Интерфейс определяет следующие методы:

а) метод возвращает общее количество мероприятий, в которых принимал участие студент;

б) метод возвращает число призовых мест (с 1 по 3), занятых на олимпиадах;

в) метод возвращает число докладов на конференциях;

г) метод возвращает строку, состоящую из названий проектов (названия разделены переходом на новую строку), за которые студент получил какую-то сумму на соревнованиях.

10 Модификация класса Student

Измените класс Student. Он теперь должен реализовывать интерфейс Activist.

Реализуйте методы интерфейса.

Помимо этого, в класс необходимо добавить следующие члены класса:

1) Поля:

– массив событий, в которых принимал участие студент.

2) Методы:

а) добавление нового события в массив событий;

б) поиск событий по дате (метод возвращает ссылку на объект события);

в) удаление события по дате.

3) Внесите изменения в **конструктор(-ы)** класса Student, чтобы они инициализировали массив событий.

11 Модификация класса Group

Измените класс Group.

Добавьте следующие ***методы***:

а) метод возвращает массив студентов-активистов (студенты, которые принимали участие хотя бы в одном событии);

б) метод возвращает список студентов-призеров (которые хотя бы раз занимали призовое место на олимпиаде или выигрывали какую-то сумму в соревновании);

в) метод возвращает количество студентов-активистов;

г) метод возвращает число "бюджетников" в группе;

д) метод возвращает число контрактных студентов в группе;

е) метод возвращает количество студентов-должников (вовремя не оплативших контракт).

12 Класс Main

В классе Main напишите код, чтобы протестировать функциональность созданных классов и реализованных методов

P.S. Работа с датами [Java | Работа с датами. LocalDate (metanit.com)]