

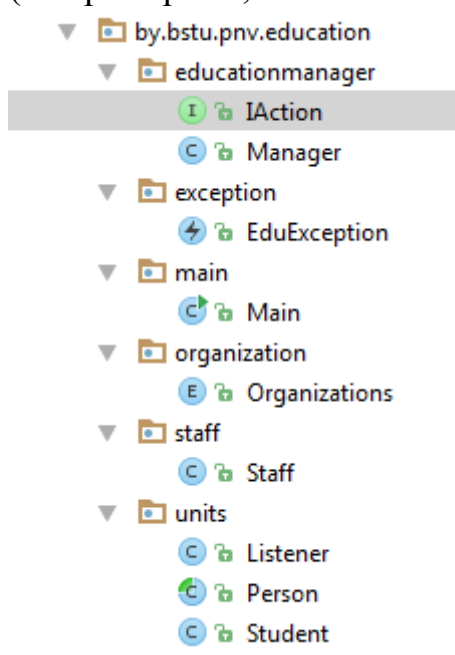
№ 2 Core JAVA

Классы, интерфейсы и наследование, коллекции и исключения, лямбды

Задание

- 1) Сейчас все студенты стремятся пройти какие-нибудь специализированные курсы. Разработайте систему регистрации, набора и управления слушателями курсов. Создается курс (Stuff), которым управляет преподаватель (Manager). На курсе могут быть два типа учащихся: студенты (они оцениваются) и слушатели (работники этой же организаций). В приложении создайте два определенных специализированных курса и набирите на него участников двумя способами: чтение данных из файла (JSON) и генерация участников со случайными данными. Затем преподаватель (Manager) определяет тройку лучших студентов по рейтингу/оценкам и выполняет сортировку студентов по двум критериям (оценка, возраст и т.п.). Для сортировки используйте компараторы.
- 2) Требования:
 - ✓ Каждый класс должен иметь отражающее смысл название и информативный состав (поля, set, get, конструкторы, toString).
 - ✓ При кодировании должны быть использованы соглашения java code convention.
 - ✓ Используйте лямбда выражения.
 - ✓ Используйте generic типы и стандартные интерфейсы
 - ✓ Классы должны быть разложены по пакетам.

(Например так, но названия классов и состав может быть другой)



- 3) Должен быть минимум один интерфейс со статическим(и) методами и реализацией метода(ов) по default (здесь IAction), абстрактный класс (здесь это Person) и один внутренний класс, а также перечисление для организаций или заведений (Organizations) из которых пришли студенты (в перечислении определите поля – например рейтинг, методы и конструкторы). Перечисление должно помимо названия организации содержать полное название организации и количество принятых студентов из этой организации.
- 4) Для сбора объектов использовать Concurrent Collections Java.
- 5) Продемонстрируйте возможности использования Optional
- 6) Для обработки исключений создать свой тип (например EduException)
- 7) Используйте тип Log и весь консольный вывод направляйте туда. Информировать о создании объектов, исключениях, выводить результаты запросов.

Проектирование класса-контейнера

Класс –контейнер, в данном случае Stuff может содержать коллекцию созданных объектов, конструкторы, set и get, добавления и удаления :

```
public class Staff {

    private ArrayList<Person> studlist;

    public Staff() { studlist = new ArrayList<Person>(); }
    public Staff(ArrayList<Person> persons) {...}

    public List<Person> getStudlist() { return studlist; }

    public void setStudlist(ArrayList<Person> studlist) { this.studlist = studlist; }

    public boolean add (Person item) throws EduException {...}
    public boolean remove (Person item){...}

    @Override
    public String toString() {...}

    public boolean compByYear (Person a, Person b) {...}
}
```

Проектирование управляющего класса

Класс должен содержать следующие методы: чтения, генерации объектов и запросы.

Например:

```

// Класс управления
public class Manager implements IAction {
    // считать с источника-----
    public Staff createGroup(Staff someCourse, int maxStudent, int maxListener, String filename)
    // сгенерировать курс-----
    public Staff generateCourse(Staff someCourse, int maxstudCount, int maxlistCount) throws Exception {
    // считать из файла
    // сумма рангов-----
    public int sumRanges (Staff anyCourse) {...}
    // количество слушателей -----
    public int countListener (Staff anyCourse) {...}
    Comparator<Person> compare = (o1, o2) -> {
        return Integer.compare(o1.getYear(), o2.getYear());
    };

    @TargetApi(Build.VERSION_CODES.N)
    public Staff sortByYear (Staff anyCourse) {...}
}

```

Вопросы

1. Какие “строковые” классы вы знаете?
2. Какие основные свойства “строковых” классов (их особенности)?
3. Чем отличаются и что общего у классов String, StringBuffer и StringBuilder?
4. Перечислите состав класса.
5. Где и как могут использоваться [static] [abstract] [final] в контексте класса?
6. Для чего используется модификатор native?
7. Что такое логический блок? Статический блок?
8. Определите параметризованный класс.
9. Как используется метасимвол «?»
10. Когда можно опускать типы параметров в лямбда выражении?
11. Как задать ссылку на метод?
12. Что могут содержать перечисления? Приведите пример
13. Какие существуют ограничения для перечисления?
14. Что такое методы подстановки?
15. Состав класса Object.
16. Что такое внутренние классы (inner)? Правила использования.
17. Что такое вложенные (nested) классы? Правила использования.
18. Что такое анонимные (anonymous) классы?
19. Можно ли перегрузить static метод?
20. Для чего используются статические методы в интерфейсе?
21. Как определить методы default в интерфейсе? Для чего?
22. Какова иерархия коллекций?
23. Каково назначение коллекции java.util.Optional<T> ?
24. Перечислите встроенные аннотации.