1. Создайте проект, в котором два класса:

1. Student - класс с 5 private полями фамилия (lastName),имя (firstName), отчество (secondName), номер студенческого (studentId), дата рождения (birthday).

2. Solution, в котором создается один объект - студент с реальными данными и информация о нем выводится на экран.

Дополнительные требования:

Оформление кода должно соответствовать существующим Java Code Conventions.

Не должны использоваться deprecated (устаревшие) классы/методы.

Вывод даты необходимо сделать в виде день месяц год (DDMMYYYY).

Если в тексте java программ встречается текст на русском языке, кодировка этого файла должна быть UTF-8.

Замечания:

1. private int studentId; Используйте тип Integer вместо int
2. private String birthday; Используйте тип Date вместо String
3. Вывод студента на экран с помощью конструкции System.out.println(student).
4. Создание студента с помощью конструктора и значений, не надо считывать их с клавиатуры

2)

1. Student - класс с 5 private полями фамилия (lastName),имя (firstName), отчество (secondName), номер студенческого (studentId), дата рождения (birthday).

2. Group - класс, который хранит группу студентов. Имеет private поля students, id.

3. Solution, в котором создается группа студентов. Данные о группе и студентах считываются из файла group.txt. После ввода информации, выведите группу на экран (проверка корректного считывания) и найдите самого старшего студента в группе, используя метод group.getOldestStudent(), выведите его на экран.

4. Предусмотрите случай когда группа студентов пустая, реализуйте метод isEmpty в группе для проверки пустоты. В случае пустой группы вместо самого старшего студента выводите сообщение "Group of students is empty".

Формат файла group.txt:

groupId

НомерСтуденческого1 Фамилия1 Имя1 Отчество1 ДатаРождения1

НомерСтуденческого2 Фамилия2 Имя2 Отчество2 ДатаРождения2

НомерСтуденческого3 Фамилия3 Имя3 Отчество3 ДатаРождения3

...

Дополнительные требования:

Оформление кода должно соответствовать существующим Java Code Conventions.

Не должны использоваться deprecated (устаревшие) классы/методы. Вывод даты необходимо сделать в виде день месяц год (DDMMYYYY).

Если в тексте java программ встречается текст на русском языке, кодировка этого файла должна быть UTF-8.

3)

1. Базовый класс Person - класс с 4 private полями фамилия (lastName),имя (firstName), отчество (secondName), дата рождения (birthday).

2. Student - класс, который наследуется от Person, также содержит private поля - номер студенческого (studentId). Класс должен реализовывать interface Comparable.

3. Teacher - класс, который наследуется от Person, также содержит private поля - должность (jobTitle), Класс должен реализовывать interface Comparable.

4. GroupWithTeachers - класс, который хранит группу студентов и всех преподавателей, кто ведет занятия для этой группы. Имеет private поля id, students, teachers.

5. Solution, в котором создается [группа студентов](http://programming-technologies.ru/courses/mod/assign/view.php?id=15). Данные о группе, студентах и преподавателях считываются из файла group2.txt. После ввода информации, выведите группу на экран (проверка корретного считывания) и список студентов отсортированных по возрасту в порядке возрастания, далее список преподавателей, отсортированных по фамилии в порядке возрастания. Для сортировки используйте метод Collections.sort().

Формат файла group2.txt:

id

Student НомерСтуденческого1 Фамилия1 Имя1 Отчество1 ДатаРождения1

Student НомерСтуденческого2 Фамилия2 Имя2 Отчество2 ДатаРождения2

...

Teacher ФамилияN1 ИмяN1 ОтчествоN1 ДатаРожденияN2 ДолжностьN1

Teacher ФамилияN2 ИмяN2 ОтчествоN2 ДатаРожденияN2 ДолжностьN2

Дополнительные требования:

Оформление кода должно соответствовать существующим Java Code Conventions.

Не должны использоваться deprecated (устаревшие) классы/методы.

Вывод даты необходимо сделать в виде день месяц год (DDMMYYYY).

Если в тексте java программ и входных/выходных файлах встречается текст на русском языке, кодировка этого файла должна быть UTF-8.

4)

Используя разработанный ранее класс Student, напишите класс Solution, который создает студента, сериализует его в файл student.bin, а затем десериализует его из этого файла. Выведите информацию о студенте на экран до сериализации и после. Сравните два объекта (студент и десериализованный студент) и выведите результат сравнения.