

## Тема: DateTime API + Exception + Junit

Дорабатываем интерфейс Animal + всех наследников, добавляя новое поле:

- LocalDate birthDate; // День рождения животного в формате dd-MM-yyyy. Генерируются любые даты, а не только текущие

Создать SearchService (интерфейс) и его имплементацию. Интерфейс должен содержать метод **checkLeapYearAnimal**, который на вход принимает любое животное и делает проверку, родилось ли животное в високосный год. Результат работы методы - вывод на экран сообщения ИМЯ + был рожден в високосный год / не был рожден в високосный год.

- Если был переданный на вход объект пустой, то необходимо выбросить unchecked exception - InvalidAnimalException (необходимо самим реализовать) с сообщением “на вход пришло некорректный объект животного” + текущая дата.
- Если был передан на вход не пустой объект, но у него не заполнена дата, то необходимо выбросить checked exception - InvalidAnimalBirthDateException (необходимо самим реализовать) с сообщением “у животного ТИП\_ЖИВОТНОГО не указана дата его рождения”. Сам метод не должен обрабатывать такое исключение. Обработку необходимо сделать в main, где результатом будет проброс InvalidAnimalException, который будет передавать сообщение “Работа метода завершилась ошибкой” и пойманный exception в cause.

На метод написать unit-тест, проверив все кейсы (с ошибками и без них). Для удобства использовать @DisplayName с описанием какой кейс проверяется. Плюсом будет использование ParameterizedTest