1. Разработать приложение, предоставляющее пользователю возможность генерации валидаторов и мапперов.
   Требования:
   1. Приложение должно предоставлять графический интерфейс со следующими элементами:
      1. Кнопка для добавления в список рабочих библиотек папки с Java классами или jar файла.
      2. Кнопка для удаления элементов из списка рабочих библиотек.
      3. Выбор типа создаваемого компонента.
   2. При добавлении нового элемента в список рабочих библиотек, приложение сканирует его содержимое и кэширует классы, помеченные аннотацией @Api. Классы используемые в качестве входных и выходных типов в методах обнаруженных классов должны быть доступны для выбора в качестве входных данных для работы генераторов.

   3. Приложение должно реализовывать работу следующих генераторов:
      1. Генератор валидаторов.
         Входные данные: класс, для которого генерируется валидатор
         Выходные данные: Java код валидатора
         Требования:
         1. Генератор должен предоставлять графический интерфейс со следующими элементами:
            1. Выпадающий список для выбора класса из п. 1
            2. Динамически создаваемый список полей указанного класса и его родителей.
               Если поле не является примитивным, то список вложенных полей и т.д. рекурсивно.
               Для каждого поля класса должна быть возможность указать обязательное оно или нет и, в зависимости от типа поля, указать ограничения, накладываемые на его значение.
               Для числовых полей - верхние и нижние пределы диапазона значений, запрет на определенные значения, список разрешенных значений.
               Для строковых полей - регулярное выражение, которому должно соответствовать поле
               Для коллекций - минимальное и максимальное количество элементов.
            3. Кнопки для сохранения и загрузки в файл.
            4. Кнопку для генерации Java кода.
         2. В соответствии с заданными в GUI настройками, генератор должен сформировать Java код класса-валидатора. Метод, выполняющий валидацию должен принимать экземпляр указанного класса и возвращать строку, содержащую список нарушений или null, если ни одно условие не было нарушено.
            Сгенерированный валидатор не должен останавливаться на первом обнаруженном нарушении, а должен выполнить максимальное количество возможных проверок.
      2. Генератор мапперов
         Входные данные: один класс, поля которого требуется заполнить, 0 или более классов, используемых для заполнения полей.
         Выходные данные: Java код маппера.
         Требования:
         1. Генератор должен предоставлять графический интерфейс со следующими элементами:
            1. Выпадающий список для выбора класса для заполнения из п. 1

    2. Динамический набор выпадающих списков для выбора классов, используемых для заполнения полей. Изначально выводится один список. Если пользователь выбрал в нем какой-то класс, добавляется еще один список для выбора еще одного класса, и т.д.
    3. Динамически создаваемый список полей указанного класса и его родителей, аналогичный списку создаваемому в генераторе валидаторов.
       Для каждого поля заполняемого класса должен быть предоставлен выпадающий список с полями прочих классов, подходящих для заполнения данного поля. Если из списка не выбрано ни одно поле, должно быть предоставлено поле для указания константного значения.
    4. При создании динамического списка полей, генератор должен автоматически выбирать значения полей, используемых для заполнения, если они совпадают по типу и имени с заполняемыми полями, в т.ч. вложенных полей.
    5. Кнопки для сохранения и загрузки в файл.
    6. Кнопка для генерации Java кода.
2. В соответствии с заданными в GUI настройками, генератор должен сформировать Java код класса-маппера. Метод, выполняющий маппинг должен принимать экземпляр указанного класса, поля которого нужно заполнить, 0 или более экземпляров классов, используемых для заполнения, и возвращать заполненный экземпляр класса.