

ПРАВИЛА

1. Для задач ниже создать один проект в IntelliJ Idea. Папку этого проекта целиком поместить в репозиторий. Также поместить туда в корень файл .gitignore (выложу его в открытый репозиторий). Далее добавить все файлы (можно с помощью `git add .` (с точкой) - это добавит все папку), закоммитить, запустить
2. (касается задач, в которых присутствует класс с методом `main`) — для каждого такого задания создается отдельный файл `TaskNNN.java`, где `NNN` - трехзначный номер задачи. Файл `TaskNNN` содержит метод `main`. Для вспомогательных классов, используемых в задаче, можно и даже желательно создавать отдельные `java`-файлы.
3. (если задача требует создания `html`-кода) — для каждого задания создать отдельный файл `TaskNNN.html`, где `NNN` - трехзначный номер задачи.
4. `java` код надо подписывать в самом верху следующим образом (привожу пример по себе на примере своей группы 953 и задачи 000):

```
/**
 * @author Mikhail Abramskiy
 * 953a
 * 000 (для вспомогательного класса указывайте для чего используется,
 *      например for 001, 002 and 007)
 */
```

001 Написать регулярное выражения для телефона в формате:

8 (123) 456-78-90

Проверить работу написанного регулярного выражения на примере одного правильного телефона и трех неправильных. Неправильные случаи должны быть разными.

002 Написать регулярное выражения для телефона в формате:

+7-123-456-78-90

Проверить работу написанного регулярного выражения на примере одного правильного телефона и трех неправильных. Неправильные случаи должны быть разными.

003 Генерировать случайные положительные целые числа. Вывести первые 10 сгенерированных четных чисел, остановить генератор, вывести общее количество сгенерированных чисел. Проверку на четность осуществлять регулярным выражением. НЕ использовать математические операции.

004 Генерировать случайные положительные целые числа. Вывести первые 10 сгенерированных чисел, в которых нет трех четных цифр подряд. Остановить генератор, вывести общее количество сгенерированных чисел. Проверку осуществлять регулярным выражением. НЕ использовать математические операции для анализа числа.

005 Генерировать случайные положительные целые числа. Вывести первые 10 сгенерированных чисел, которые содержат более 3 и менее 6 четных цифр и ни одной нечетной. Остановить генератор, вывести общее количество сгенерированных чисел. Проверку осуществлять регулярным выражением. НЕ использовать математические операции для анализа числа.

- 006 Генерировать случайные положительные целые числа. Вывести первые 10 сгенерированных чисел, в которых нет ни двух четных, ни двух нечетных цифр подряд (например, 12345678). Остановить генератор, вывести общее количество сгенерированных чисел. Проверку осуществлять регулярным выражением. НЕ использовать математические операции для анализа числа.
- 007 Генерировать случайные положительные целые числа. Вывести первые 10 сгенерированных чисел, в которых есть как минимум два раза встречается группа из 2 четных цифр. Остановить генератор, вывести общее количество сгенерированных чисел. Проверку осуществлять регулярным выражением. НЕ использовать математические операции для анализа числа. Использовать только matches.
- 008 Сделать задачу 007 с использованием find() и group()

Скачайте какую-нибудь html страницу. Она будет входным файлом для следующих задач. Для чтения этого файла можно использовать, например, Scanner или BufferedReader. Не забудьте добавить эту страницу в репозиторий.

- 009 Вывести все открывающие тэги, в которых нет ни одного параметра, т.е.:

```
<body> - надо выводить  
<body bgcolor="#000000"> - не надо
```

Вывести только уникальные теги, без повторений.

- 010 Вывести все использованные имена картинок: image.jpg, icon.png и т.д.
- 011 Вывести все использованные имена параметров в тегах, например:

```
<p color="#000000" border="1"> - надо вывести слова "color" и "border".
```

Задачи на html и css.

- 012 Сверстать html страницу, содержащую таблицу умножения
- 013 Сверстать страницу, на которой будет отображаться 3 формы поиска - в mail, в yandex и в google. Каждая форма должна содержать текстовое поле для ввода запроса и кнопку с текстом соответственно для каждого "Искать в Mail/Yandex/Google". При нажатии на кнопку открывается страница поиска введенного текста в указанной системе.
- 014 На почту группы придет картинка 014.jpg. Мне необходимо, чтобы вы повторили то, что вы видите на ней, в html и css. **Обязательные вещи:** текст, размер (примерный) и цвет (примерный) шрифта, расположение блоков друг относительно друга, цвет ячеек в календаре. Толщина линий, поля, шрифт произвольные. Картинки, границы, закругления, фон, иконки, вертикальные разделители в меню необязательны. Залить два файла - html и css. Текст содержимого на странице пришлю вместе с картинкой на ящик группы.
- 015 На паре рассматривался пример, когда с помощью js можно динамически выводить количество символов, введенных в textarea. Усовершенствовать код так, чтобы можно было вводить только 140 символов. Выводиться должно количество оставшихся символов. При достижении предела выводить количество оставшихся равное 0, не давать вводить новые символы в textarea.
- 016 Задача с классной работы. Картинка на почте группы. 016.jpg

017 Создать приложение на сервлетах, представляющее собой онлайн калькулятор. На стартовой странице находятся два текстовых поля ввода, выпадающий список (гуглим использование `< select >`) и кнопка. В текстовые поля вводятся два вещественных числа (иногда только одно). Из выпадающего списка выбирается операция (+, −, *, /, *sin*, *cos*, *ln*, *exp*). При нажатии на кнопку осуществляется отправка на сервер введенных данных (чисел и операции). Там происходит их валидация. Если одно из чисел введено неправильно или вообще не введено (там, где нужно два числа), то возвращается страница с начальной формой, в которой данные заполнены так, как были заполнены пользователем до нажатия, а над формой пишется сообщение об том, какая ошибка произошла (например, «не введен первый аргумент», «введено не число», «введено отрицательное число для логарифма»).

Если же все правильно, возвращается страница с результатом, отображаемым крупным шрифтом.

018 Создать приложение с использованием сервлетов, которое содержит следующие страницы:

Страница «Обо мне»

Содержит табличку из двух колонок. В первой жирным шрифтом надписи «Имя», «Никнейм», «Email», «Интересы». Во второй колонке - соответствующие вам значения. Интересов должно быть минимум 3. Каждый интерес - не просто надпись, а ссылка, при нажатии на которую осуществляется поиск в Яндексе, содержимым поиска будет как раз текст вашего интереса. Если вдруг возникают ошибки кодировки, попробуйте писать интересы на английском.

Страница «Мой online-калькулятор»

Содержит форму из задания 017 с тем же (из задание 017) функционалом.

Страница «Моя таблица умножения»

На странице отображается форма с текстовым полем и кнопкой. Вводится число, после нажатия на кнопку вы переходите на страницу, где отображается таблица умножений чисел от 1 до введенного. Например, для числа 3:

1	2	3
2	4	6
3	6	9

Связь между страницами осуществляется тем, что на каждой странице есть меню "Обо мне
Мой online-калькулятор
Моя таблица умножения!"

МЕНЮ ОТОБРАЖАЕТСЯ НА КАЖДОЙ СТРАНИЦЕ!

Дополнительные требования:

1. Каждая страница отображают текущую дату и время, когда вы зашли на страницу (это требование решайте максимально просто, гуглите `java.util.Date`). Все равно, где дату отображать, можно внизу.
2. Каждая страница содержит крупный текст своего заголовка.

019

- 020
- Создать классы для всех таблиц из `virtual-schools`. Хранить внешние ключи и как числа, и как объекты.
 - В отдельном классе в методе `main` создать коллекции, соответствующие таблицам, и с помощью `sql`-запросов, заполнить их данными, сохраняя целостность БД и грамотно выстраивая ссылки
 - Вывести информативно таблицу "Attendance опираясь только на коллекции и не запрашивая БД.

- 021 Создать таблицу users, с полями username и password. Вставить в нее 1-2 строки. Реализовать в проекте с сессиями и cookie (исходники лежат в открытом репозитории в папке 07 session, cookies) проверку пользователя с подключением к БД и проверке наличия пользователя и соответствия имени пользователя и пароля в таблице users.

Пока без хэширования пароля, но для целей семестровой начинайте читать про класс MessageDigest.

- 022 Найдите в сети страницу, на которой лежат прямые ссылки на pdf или mp3 файлы. Например, страницы википедии о каких-нибудь городах странах содержат много pdf-статистики в приложении. Надо с помощью класса URL считать html-код страницы, в нем с помощью регулярных выражений найти ссылки на все указанные на странице файлы, составить на их основе составить новые экземпляры класса URL, с помощью которых требуется скачать все указанные файлы. Имя файла должно совпадать с именем файла в url.

Рассмотрим пример: Пусть есть URL:

`http://ru.wikipedia.org/wiki/Лондон,`

Там в примечаниях лежат ссылки на .pdf файлы. В html-коде страницы (который вы можете считать) это выглядит вот так:

```
<li id="cite_note-43"><b><a href="#cite_ref-43">^</a></b><span
class="reference-text"><span class="citation"><a rel="nofollow
" class="external text" href="http://data.london.gov.uk/datastorefiles/documents/2011-census-first-results.pdf">2011 Census
first results: London boroughs' populations by age and sex</a>
8, 10, 18. Office for National Statistics, Greater London Au
thority (2012).&#160;- Первые результаты переписи 2011 года:
население районов Лондона по возрасту и полу. <small><a rel=
"nofollow" class="external text" href="http://www.webcitation
.org/6B0kk8GjH">Архивировано</a> из первоисточника 14 октября
2012.</small></span></span></li><li id="cite_note-44"><b><a h
ref="#cite_ref-44">^</a></b><span class="reference-text"><spa
n class="citation"><i>Джон Холлис.</i> <a rel="nofollow" clas
s="external text" href="http://data.london.gov.uk/datastorefi
les/documents/FocusOnLondon-PopulationAndMigration.pdf">Focus
on London. Population and migration</a>.&#160;- Лондон: Great
er London Authority, 2010.&#160;- С.&#160;11, 15.</span></spa
n></li>
```

Не обращайте внимания на все то, что там написано, но заметьте в паре строк ссылки на pdf файлы:

* `http://data.london.gov.uk/datastorefiles/documents/2011-census-first-results.pdf`

* `http://data.london.gov.uk/datastorefiles/documents/FocusOnLondon-PopulationAndMigration.pdf`

Их можно найти с помощью регулярных выражений. А дальше имея url-строку файла, вы можете скачать его с помощью `BufferedInputStream` / `BufferedOutputStream`.

Имена в данном случае у скачанных файлов должны быть:

- * `2011-census-first-results.pdf`
- * `FocusOnLondon-PopulationAndMigration.pdf`

Их вы можете извлечь из полученных адресов с помощью регулярных выражений строк банальным `split("/")`, тогда именем файла будет последний элемент полученного split-ом массива.