

1. Отчет по работе	3
2. ИЗУЧЕНИЕ командной строки	3
Задания и Вопросы	3
3. Установка subversion на локальную машину.	4
Задания и Вопросы	4
4. Изучение VIM	4
Задания и Вопросы	5
5. Установка Java	5
6. Изучение Java	5
Задания и Вопросы	5
7. Установка IDEA	6
8. Установка Tomcat	6
9. Создать простой Servlet	6
Задания и Вопросы	7
10. Выкладка на тестовый Tomcat	7
Подключение на ssh (security shell тестовый сервер)	7
Обзор каталогов на удалённом сервере	7
Копирование файла на ssh	8
Активирование изменения	8
Посмотреть log исполнения	8
Подключение к БД из Servlet	8
11. Создание репозитория	9
12. Структура META (база данных)	9
Подключение к БД	10
Получение объекта	10
Получение значения атрибута	10
Получение значения атрибута по id объекта	10
Фильтр. Применение условия выборки	11
Получение набора объектов	11
Изменить значение атрибута	13
Создать объект	13
Задания и Вопросы	13
13. Работа с редисом	14
Задания и Вопросы	15
14. Дополнительные возможности системы	15
Просмотр событий	15

Запуск работа	16
Закрытие задачи в системе	16
15. Итоговые задания	16

## 1. Отчет по работе

В конце рабочего дня заполнять так называемые почасовки вот в этой системе:

<https://bg.market/т/>

Логин и пароль такой же, как для авторизации в систему задач.

Выбираете раздел Почасовки, нажимаете ссылку Добавить, заполняете дату, количество отработанных часов и в поле Описание пишете, что конкретно вы делали, какую задачу решали, каких результатов достигли.

## 2. ИЗУЧЕНИЕ командной строки

[https://disk.yandex.ru/d/SXo\\_j0C3c1J2Jg](https://disk.yandex.ru/d/SXo_j0C3c1J2Jg)

Прочитать книгу: Интерфейс командной строки UNIX. В. Е. Сорокин.

В какой папке находитесь: pwd

Содержимое текущего каталога: ls

Подробное содержание текущего каталога: ls -l

Копировать файлы: scp "что" "куда"

Создание каталога: mkdir "название"

Переход в каталог: cd "куда"

Создать новый файл: touch "имя файла с расширением" (touch test.txt)

Изменение прав: chmod XXX name

Удалить папку со всем содержимым: rm -rf "Название папки"

Исполняемые процессы: ps

Прерывание: ctrl+C

Скомпилировать: javac name.java

Запуск: java name

---

### Задания и Вопросы

1. Узнать, где находитесь.
2. Добавить новую папку с вашим именем.
3. Создать в ней несколько файлов с разным расширением
4. Вывести подробное содержание папки
5. Поменять права всех файлов в вашей папке на 664
6. Вернуться на одну папку выше

7. Удалить папку с вашим именем

### 3. Установка subversion на локальную машину.

Прочитать, что такое subversion (svn - система контроля версий)

<http://svnbook.red-bean.com/nightly/ru/svn-book.html#svn.tour.initial>

<http://blog.alwawee.com/>

[2012/01/30/%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0-%D1%81-subversion-%D0%B2-mac-os-x/](http://2012/01/30/%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0-%D1%81-subversion-%D0%B2-mac-os-x/)

Установить: `sudo apt install subversion`

---

Задания и Вопросы

1. Установить на локальную машину Subversion
2. Создать репозиторий
3. Создать рабочую копию репозитория
4. Выполнить простейший рабочий цикл:
  - обновление рабочей копии
  - внесение изменений
  - анализ изменений
  - слияние изменений
  - фиксация изменений
5. Зачем нужна рабочая копия?
6. Посмотреть, кто правил файл.

### 4. Изучение VIM

Редактор VIM позволяет переписывать файл из терминала.

Установить VIM: `sudo apt install VIM`

Нажатие на клавишу "i" позволяет войти в режим правки файла.

Клавиша Esc позволяет выйти из режима правки.

Нажатие на / начинает поиск по файлу.

Ввод :w позволяет сохранить правленный файл.

При вводе :q произойдет выход из файла.

---

## Задания и Вопросы

1. Исправить файл и сохранить изменения.
2. Исправить файл, сохранить изменения и выйти, нажав на "Закрытие окна", а не путём ввода :q
3. Разобраться, как работает поиск по файлу.

## 5. Установка Java

Требуется установить Java версии 7.

Для установки необходимо распаковать архив в папку.

[https://disk.yandex.ru/d/SXo\\_j0C3c1J2Jg](https://disk.yandex.ru/d/SXo_j0C3c1J2Jg)

## 6. Изучение Java

(Книга содержится во вложении), для первого дня изучения - прочитать 1 - 3 главы.

Необходимо усвоить: (1)

- как выводить информацию в терминал
- стр. 51 - рассказать все ключевые слова

---

## Задания и Вопросы

1. Какие бывают типы данных?(1)
2. Чем отличается ссылочный тип данных от элементарных? (1)
3. Написать функцию, которая будет выводить в консоль "Hello, World!". (1)
4. Сделать Exception в функции. (1)
5. Написать функцию, в которую при запуске передаём значения, а она их перемножает, складывает и т.д. (1)
6. Сделать комментарии в коде. (1)
7. Написать функцию, в которую передаём значения, а она показывает большее из них. (1)
8. Сделать проверку try catch. (2)
9. Зачем нужен throws Exception ? (2)
10. Чем ArrayList отличается от массива? (2)

11. Расскажите про TreeSet, HashSet, HashMap, LinkedHashMap и т.д. (2)
12. Зачем использовать константы в большой системе? Приведите примеры. (2)
13. Чем & и | отличаются от && и || ? (2)
14. Что такое StringBuffer, StringBuilder? (2)
15. Написать функцию, которая будет выбирать меньшее число из нескольких введённых. (2)
16. Написать функцию, которая будет считывать вводимые параметры, и будет выводить сумму чётных позиций введённых чисел (0+2+4+6..) и разность нечётных позиций введённых чисел (1-3-5-7..) (2)
17. Остановить предыдущую функцию, когда чётные станут больше 100 или нечётные станут меньше 0. (2)
18. Написать функцию, в которую передаём дату рождения, а она показывает сколько до ближайшего (проверка в обе стороны). Можно использовать библиотеки из интернета. (2)
19. Добавьте в ArrayList и LinkedList по N элементов. А потом засекуте выборку M раз из этих листов. Вывести разницу. N и M задаем.(2)

## 7. Установка IDEA

Скачать и распаковать архив в папку.

Настроить проект bg из локальной копии репозитория, создав новый проект в IDEA из исходников и подключив необходимые библиотеки. (меню File/Project Structure)

Настроить параметры Subversion. (меню VCS/Subversion)

## 8. Установка Tomcat

Прочитать 4-ую главу книги JAVA

Скачать и распаковать архив в папку

Потренироваться запустить и остановить Tomcat

Запустить Tomcat и проверить его работоспособность: <http://localhost:8080/>

## 9. Создать простой Servlet

В простом проекте IDEA создать класс сервлета, скомпилировав его в .class

Скомпилированный файл перенести в соответствующую папку TOMCAT\_HOME\webapps  
Настроить web.xml  
Запустить и остановить tomcat, проверить работоспособность созданного servlet.  
Изучить более подробно: <http://java-course.ru/student/book1/servlet/>

---

## Задания и Вопросы

1. Что такое Servlet?
2. Зачем нужен Servlet?
3. Чем post-запрос отличается от get-запроса?
4. Написать функцию, которая будет выводить на странице "Hello, World!".
5. Написать функцию, которая из url считывает значения и выводит их на странице.
6. Написать функцию, которая из url считывает дату, проверяет её и выводит ошибку, если не корректна, так же для даты со временем.
7. Написать функцию, с формой на странице, которую можно заполнить и сохранить значения в url. Например: Имя, Фамилия, Дата рождения. Должны быть поля ввода и кнопка сохранить.

## **10. Выкладка на тестовый Tomcat**

### **Подключение на ssh (security shell тестовый сервер)**

Команда на подключение: ssh login@studentn.isasoft.ru

Ввести пароль

### **Обзор каталогов на удалённом сервере**

В каталоге /w2/srv/bin найти свою папку

Каталог /w2/srv/tomcat21/lib содержит общие библиотеки и файлы properties.

Каталог /w2/srv/tomcat21/webapps/bg содержит локальную копию проекта.

Файл /w2/srv/tomcat21/webapps/bg/WEB-INF/web.xml содержит настройки servlet.

Каталог /w2/srv/tomcat21/webapps/bg/WEB-INF/classes/bg содержит классы, пакета site.

### **Копирование файла на ssh**

```
scp откуда/что login@student.isasoft.ru:/куда  
scp /Home/IdeaProjects/Bg/out/production/Bg/bg/*.class  
login@studentn.isasoft.ru:/w2/srv/tomcat21/webapps/bg/WEB-  
INF/classes/bg
```

Ввести пароль на девелоперский сервер.

Важно! Копирование должно запускаться с локального терминала.

### **Активирование изменения**

Для вступления изменений в силу, tomcat должен перепроверить содержимое классов и настроек. Данная команда сообщает tomcat об этом.

```
touch /w2/srv/tomcat21/webapps/bg/WEB-INF/web.xml
```

Можно полностью перезапустить томкат, например, если перезаписать все классы:

```
/etc/init.d/atomcat21 stop - выключение
```

```
/etc/init.d/atomcat21 start - включение
```

### **Посмотреть log исполнения**

(последние строки)

```
tail -f /w2/srv/tomcat21/logs/catalina.out
```

### **Подключение к БД из Servlet**

Строка подключения:

```
ResourceBundle mains =
```

```
ResourceBundle.getBundle(getServletConfig().getIni-  
tParameter("main_resource"));
```

Важно! В web.xml сервлета, использующего подключение к БД, нужно прописать параметр main\_resource со значением bg\_main

```
<init-param>
```

```
    <param-name>main_resource</param-name>
```

```
    <param-value>bg_main</param-value>
```



<init-param>

Важно! При создании вспомогательных классов в сервлете, нужно копировать автоматически созданный файл вашего класса на tomcat. Файл имеет следующее название: Имя Сервлета \$ИмяКласса.class

## 11. Создание репозитория

Настроить проект bg из локальной копии репозитория, создав новый проект в IDEA из исходников и подключив необходимые библиотеки.(меню File/Project Structure)Настроить параметры subversion. (меню VCS/Subversion)

Создать рабочую (локальную/рабочую) копию репозитория site:

svn checkout [svn://svn.bibliki.ru/bgrep/site](http://svn.bibliki.ru/bgrep/site)

Или через IDEA:

VCS -> Checkout from Version Control -> Subversion

Repositories: + svn://svn.bibliki.ru/bgrep/

login, password

папки: lib, site, dock, smpp, dict

## 12. Структура МЕТА (база данных)

у объекта есть тип, id, атрибуты

4 Страна:

Название	N	Тип
Код страны	1001118	Список
Карта	1000457	Рисунок
Въезд по визе	1001264	Да/Нет
Города вылета	1004101237	Ссылка

5 Город:

Название	N	Тип
Город-агенство	1000102	Да/нет
Город-аэропорт	1000100	Да/Нет

Город-консульство	1000103	Да/ Нет
-------------------	---------	------------

26 Отель:

Название	N	Тип
Юридический адрес(англ)	1026100050	Строка
Город	1000117	Ссылка
Валюта	1000118	Список
Категория	1000119	Список

### Подключение к БД

Учитывая разнообразие используемых серверов, применяется единый метод подключения, через инициализацию ResourceBundle, где описываются параметры подключения:

```
ResourceBundle mains =
ResourceBundle.getBundle(getServletConfig().getInitParameter("main_resource"));
getServletConfig().getInitParameter("main_resource") возвращает название файла с параметрами подключения. Берется это название из параметров в описании Servlet (файл web.xml)
```

### Получение объекта

Используется метод getObject, класса Ob0, где idOb это id объекта. Obb row = Ob0.getObject(mains, idOb);

### Получение значения атрибута

Получив объект

```
Obb ob = Ob0.getObject(mains, idOb);
```

чтобы получить значение атрибута, используется метод getAttribute, Ob0.getAttribute(объект ob, номер атрибута);

### Получение значения атрибута по id объекта

Для получения значения атрибута, требуется id объекта и id\_oba атрибута, с указанием типа атрибута.

```
Ob0.getZn(<ResourceBuilder>, id, id_oba,  
type_oba);
```

id: id объекта

id\_oba: id атрибута

type\_oba: тип атрибута

Например:

```
Ob0.getZn(mains, id26, атрибут26 Город,  
DatabaseSettings.idTypeObaLink);
```

```
или db.getAtById(id26, атрибут26 Город,  
DatabaseSettings.AtrType.LINK);
```

### **Фильтр. Применение условия выборки**

Для применения условий, требуется определить фильтр:

```
Obb fltr = Ob0.createFilter(тип26);
```

В определении фильтра указывается тип, по которому будет производиться выборка.

После этого определяются условия выборки.

```
Ob0.addCondition(fltr, атрибут26 Валюта, EQ,  
«RUR»);
```

В условии указывается атрибут, по которому будет произведена выборка, условие выборки (EQ) и значение выборки («RUR»).

Если выборку требуется произвести по атрибуту, доступному через ссылку на другой

объект, то нужно указать последовательность доступа к этому атрибуту, через ссылочный

элемент в виде последовательного массива атрибутов. Этот массив может иметь

больше элементов, если требуется обратиться к атрибуту с большей вложенностью.

```
Ob0.addCondition(fltr, new int[] {атр1, атр2},  
LED, dEnd);
```

И выполнить запрос на получение набора объектов.

```
Ob0.getSrcObs(mains, ...)
```

### **Получение набора объектов**

Для примера возьмем тип отель.

Определяем фильтр на тип:

```
Obb fltr = Ob0.createFilter(ТИП26);
```

Получаем набор объектов:

```
Obb[] myObs =
```

```
Ob0.getSrcObs(mains, Ob0.metaconnname, false,  
fltr, 0, 0, 2);
```

Рассмотрим полный набор возможных параметров запроса:

```
Obb[] myRecs = Ob0.getSrcObs(<par1>, <par2>,  
<par3>, <par4>, <par5>, <par6>, <par7>, <par8>,  
<par9>, <par10>, <par11>, <par12>, <par13>,  
<par14>, <par15>, <par16>);
```

Описание параметров:

par1: ResourceBundle (обязательный) Название файла с параметрами подключения

par2: Ob0.metaconnname(обязательный) Имя соединения, определяется администратором

par3: debug (false) Требуется ли отладочная информация по обработке запроса

par4: Filter(обязательный. Фильтр. Указывает на нужный тип и возможные условия выборки)

par5: m Порядковый номер первой записи

par6: ent Количество выводимых записей

par7: us Пользователь (записи созданные определенным пользователем)

par8: lid Возвращает в массив ArrayList<Long> ID объектов

par9: datafis <дата время> состояние базы на заданный момент времени

par10: DATANS массив String[] из двух дат, для выборки данных введенных за период

par11: new Int[] {0} Определяет набор блоков для поиска данных

par12: StringBuffer (обязательный) возвращает кол-во объектов. Если установлен, то без самих объектов

par13: ArrayList<Long> Возвращает id объектов

par14: new Int[] {0} Определяет id атрибутов, данные по которым требуется вернуть

par15: тип сортировки 2:по мере добавления; ID атрибута: по атрибуту; (отрицательные): в обратном порядке

par16: isOb Boolean, Определяет требуется ли возвращать дату создания объектов

### Изменить значение атрибута

```
Obb ob = Ob0.getOb(mains, id);
Ob0.addAt(ob, атр, знач);
ob.id_user = whoami.id;
Ob0.edtOb(mains, ob);
```

### Создать объект

```
Obb ob = new Obb(тип26);
// добавить значение
Ob0.addAt(ob, атр, знач);
// добавить массив значений
String[] vResultIds = Ob0.addAts(ob, атр, знач[]);
Ob0.addOb(mains, ob); // сохранить изменения
```

---

### Задания и Вопросы

1. Выписать на странице в таблице все id + Города 5 Московской области.

1005101368 территория - Мос. Область 100518301512

2. Выбрать все города (тип 5 1000004) России 100410000050, где 1000101 Город-вылета = Да. На экран вывести таблице, в первой колонке по алфавиту идут территории, а во втором столбце по алфавиту города в этих территориях: Название области 1005101368, Название города 1000098.

3. Выбрать всех Туристов (тип 23), которые родились позже 01.01.1980. На экран для каждого туриста вывести в формате JSON: номер паспорта, ФИО туриста, дату рождения.

1000152 дата рождения

1001446 1001447 серия номер паспорта

1000144 1000146 1000147 ФИО

4. Отправить письмо (sendLetterToSomebody) с информацией о трансферах (id трансфера, название трансфера, и все транспортные средства):

1000269 тип трансфера - индивидуальный,

1001189 страна - Россия,  
1038922658 Партнер - БГ ЮГ 115836399146,  
1000265 Название,  
1001187 Список ТС(ссылка на 143 тип), 1001179 название  
ТС(143 тип).

5. Найти в коде, где обрабатываются запросы:

[https://student.isasoft.ru/site?](https://student.isasoft.ru/site?flt=100410000050&action=price&tid=211&id_price=-1&xml=11)

[flt=100410000050&action=price&tid=211&id\\_price=-1&xml=11](https://student.isasoft.ru/site?flt=100410000050&action=price&tid=211&id_price=-1&xml=11)

[https://student.isasoft.ru/small.shtml?](https://student.isasoft.ru/small.shtml?action=transfer&tid=38&data=25.09.2019&flt=100410000049)

[action=transfer&tid=38&data=25.09.2019&flt=100410000049](https://student.isasoft.ru/small.shtml?action=transfer&tid=38&data=25.09.2019&flt=100410000049)

### 13. Работа с редисом

Redis - это хранилище данных (кэш). Работает очень быстро. В redis складывается информация, которую надо часто и быстро использовать, но эта информация может меняться, поэтому нельзя положить ее в константу.

Подробнее про редис: <https://aws.amazon.com/ru/redis/>

Для работы с редисом потребуются следующие команды:

`byte[] bKey = (key).getBytes();` - кодирует данную строку в последовательность байтов

`byte[] VL2 = Ob3.get(mains, pref, pref, bKey);` - в pref передаем название редиса. Функция возвращает значение в редисе, которое лежит по ключу bKey.

`Util.B2L, Ob0.fromBytes, Util.B2I` - перевод из байт.

`Ob0.toBytes` - так же перевод в байты

`Ob3.puts(mains, pref, pref, bKey, Ob0.toBytes(ob) );` - кладем данные в редис.

Полная версия функции, которая кладет данные в редис:

```
puts(ResourceBundle mains, String schem, byte[] key, byte[] val, long time, boolean nx, int expire, String oper);
```

Описание параметров:

par1: ResourceBundle (обязательный) Название файла с параметрами подключения

par2: schem - название редиса

par3: key - ключ

par4: val - значение

par5: time - радиус времени охвата машин, обычно 1000000000L

par6: nx - вызываем функцию setnx или нет. По умолчанию ставим false

par7: expire - время жизни ключа, потом он и его значение удаляются

par8: oper - C - используем редис, как хранилище данных. L и др. - для других операций.

---

### Задания и Вопросы

1. Какие данные, по вашему мнению, должны храниться в редисе? Как обезопасить себя от возможных сбоев редиса?
2. Написать интерфейс, чтобы отображать данные из редиса. Зашли в интерфейс, написать откуда, если в редисе нет ключа, написать, что заново положили ключ+значение(объект46) в редис и положить этот ключ+значение, или просто достали из редиса - написали об этом. В интерфейсе по id 46 надо отображать следующую информацию: 1000348 Название, 1000350 id номера, 1046222729 типа стоимости (0/1/2 - С/БНС/НС).Все кладем в редис lprice, время жизни ключа = 3 минуты.(id для теста 104610284458, 104610284457, 104610284456)
3. Добавить любой ключ в редис и научиться его удалять.

## 14. Дополнительные возможности системы

### Просмотр событий

<https://student.isasoft.ru/rservice?>

[pdatan2=01.01.2019+00:00:00&pdatak2=31.12.2019+23:59:59&submit=](https://student.isasoft.ru/rservice?pdatan2=01.01.2019+00:00:00&pdatak2=31.12.2019+23:59:59&submit=Выбрать&action=preview&task=ListOfEvents&ptype=)

[Выбрать&action=preview&task=ListOfEvents&ptype=](https://student.isasoft.ru/rservice?pdatan2=01.01.2019+00:00:00&pdatak2=31.12.2019+23:59:59&submit=Выбрать&action=preview&task=ListOfEvents&ptype=)

ptype: тип просматриваемых событий

pdatan2: начало событий

pdatak2: окончание событий

## **Запуск работа**

```
/usr/java/latest/bin/java -cp /w2/srv/tomcat21/webapps/bg/WEB-INF/classes:/w2/srv/tomcat21/lib:/w2/srv/tomcat21/lib/* bg.Robot  
bg_main co_bg RobotName
```

## **Заккрытие задачи в системе**

При закрытии задачи, в комментарии нужно указать id задачи (например):

#123456+название

Правила прописывания урлов:

1) для site нужно указывать action, например, site? action=title

2) для rservice нужно указывать task, например, rservice?task=tourInfo

3) для роботов, запускающихся через bg.Robot, нужно указывать app, например, app=createClassCache

4) для событий нужно указывать номер события, например, E21

5) для роботов, запускающихся не через bg.Robot, нужно указывать строку запуска, не включающую переменные, например, bg.AvangardPayments gpb

6) статика (для сайтов, открывающаяся на www.bgoperator.ru, bg-optics.ru и т.д.)

выкладывается автоматически и пути берутся из коммитов

7) если нужно выложить статический файл, доступ к которому нужен приложению, нужно

указывать урл этого приложения, например, нужно выложить templates/hotel-register.txt,

чтобы его мог прочитать RegisterServlet на /reg, в урле нужно указать /reg

## **15. Итоговые задания**

100410000050 - Россия

100574745019 -Анапа

124331253701 - Библио-Глобус

если формат даты не указан, то формат 30.07.2019

Во всех таблицах делать нумерацию строк в левом столбце.



1. Вывести последние 50 агентств 112, у которых страна 1000844 - Россия, город 1000845 - Анапа, 1112101718 Туроператор - Библио-Глобус. В таблице вывести отсортированные по дате создания агентства, их названия 1112100000, город вылета 1112100366, номер телефона 1001501.
2. Написать функцию, которая достанет все рейсы 31 типа. Пересохраните массив в ArrayList, LinkedList. Засеките по каждому листу отдельно, сколько времени необходимо для вывода id каждого объекта. В конце выведите каждое время и разницу между ними.
3. Для существующих сейчас цен страховок 24, дата начала существования 1024700003 (30.07.2019 13:52:12), дата конца существования 1024700004 (формат 30.07.2019 13:52:12). Вывести на экран все цены дата начала действия будет распространяться на сегодня 1024100142 и дата окончания минимум через 5 недель 1024100143. В выводимой таблице необходимо указать id цены, даты существования и даты действия, страну 1024100033 + ее название 1000000, зная id валюты 1024400040, вывести ее название, используя BgAdminServlet.VALUTNUMBER\_INTEGER и BgAdminServlet.VALUTNAME. Засечь все время работы метода и отрисовать под таблицей.
4. Сгенерировать xml файл (создать файл с расширением xml, можно без интерфейса), который будет сохраняться в вашей папке /w2/srv/bin/name. В файле должна быть информация о шаблонах 211, в которых страна 1211100003 - Россия, дата начала 1211100142 и дата окончания 1211100143 такие, чтобы выехать 1го января следующего года, а вернуться 1го февраля следующего года. Выбрать только такие шаблоны, в которых город вылета 1211100059 и город возврата 1211101224 совпадают. В xml вывести id, описание 1211100063, город вылета/возврата, ВСЕ рейсы туда 1211922717 + их номера 1000213, и ВСЕ рейсы обратно + их номера.
5. Отрисовать в браузере pdf файл, в котором написать «форму/билет» для экскурсии. Все необходимые атрибуты (110 - описание экскурсии, 36 - стоимость экскурсии):

1000427 город 36, 1000790 ссылка на описание экскурсии (36 на 110), 1036922797 ссылка на контрагента (36 на 317), 1000782 полное описание экскурсии 110, 1000801 страна 110, 1110200042 Тип (экскурсия - 0, билет - 1, спорт - 2, прокат - 3, услуга - 4, СПА - 5, авиация - 6) (нам нужна именно экскурсия), расписание 32: 1000226 дата начала, 1000227 дата окончания, 1032100376 ссылка на 36, 1000311 - время начала (формат 30.07.2019 13:52:12), время окончания 1000312 (формат 30.07.2019 13:52:12) .В форме должно быть место для заполнения фио, описание экскурсии, время начала, в какой стране и в каком городе экскурсия. На каждой странице напечатать свой билет, для всех экскурсий по России, существующих на данный момент времени.

6. Создать страницу для заполнения данных, из нее считать фио 1207100838, фамилия 1207100085, имя 1207100087, отчество 1207100089, дата рождения 1207100094. Для частного лица (как пример 120700000810000095, можно взять и другого частника) изменять в форме данные, после изменения по кнопке - сохранить (в фио и по отдельности должны совпадать значения).
7. Достать все объекты 26 типа, засечь, сколько понадобится времени на заполнение hashSet id отеля, и по очереди доставать значения. Сравнить со временем работы TreeSet.
8. Достать все объекты 83 типа и положить каждый из них в редис на час. Засечь время, за которое по 1 объекту достанете все из редиса. И Время для получения из базы тех же объектов по 1.