

Task 1

1. На вход подаётся строка (с консоли) вычисляется сумма чисел.

Пример:

Input: 1w4tt!7

Output: 12

2. На вход подаётся строка (с консоли) и вычисляется частота вхождения каждой буквы.

Input: 1w4tt!7

Output:

1. *t - 2*

2. *w - 1*

3. В некоторой школе занятия начинаются в 9:00. Продолжительность урока — 45 минут, после 1-го, 3-го, 5-го и т.д. уроков перемена 5 минут, а после 2-го, 4-го, 6-го и т.д. — 15 минут.

Дан номер урока (число от 1 до 10). Определите, когда заканчивается указанный урок.

Выведите два целых числа: время окончания урока в часах и минутах. (решить задачу без циклов и условий)

Input: Дан номер урока (число от 1 до 10).

Output данные: Выведите два целых числа: время окончания урока в часах и минутах.

Input: 3

Output: 11 35

Input: 2

Output: 10 35

Task 2

Реализуйте библиотеку *reversestring.jar*

Основная цель библиотеки - расширить метод **reverse** классов *StringBuilder/Buffer*.

Набор методов:

- обычный реверс
- реверс по указанной подстроке в строке
- весь этот функционал + пользователь может указать, с какого символа, индекса, строки или *CharSequence* начать/закончить

примеры:

```
public static String reverse(String src);
```

```
-> hello world
```

```
<- olleh dlrow
```

```
public static String reverse(String src, String dest);
```

```
-> hello world, worl
```

```
<- hello lrowd
```

```
public static String reverse(String src, int firstIndex, int  
lastIndex);
```

```
-> hello world, 3, 7
```

```
<- helol owrld
```

Использование коробочного **reverse** запрещено!

Task 3

Реализуйте консольное CRUD приложение. Выберите для себя *entity* на ваш вкус и организуйте *dao*, *service* и *controller* слои для CRUD операций! Запрещено использование коробочных коллекций, того же LIST! Вы должны использовать для вашей DB_IN_MEMORY массив ваших entity. Например, из нашего примера по UserDB, вместо List<User> вы используете User[]. Помним, что массивы не резиновые, соответственно, продумайте, как увеличить размер массива, если он заполнен и наоборот, если вы удаляете, то подумайте, как заполнять дыры в массиве, например, вместо users[6] = null, я хотел бы увидеть смещение users[7] в users[6].

Task 4

Цель: создание приложения, в котором осуществлен CRUD минимум двух дата классов и организации связей между ними. Например: Book & Author, Department & Employee, Student & Course etc.

CRUD должен учитывать релейшн (на курсе может быть много студентов так же как и студент может быть на разных курсах).

Хорошо продумайте удаление: это специфическая операция и не всегда удаление – это физическое удаление (из тех же таблиц БД).

Программа должна логироваться, продумайте разделение логов по уровням и файлам!

Программа должна быть покрыта тестами, которые бы действительно отображали тестирование критических узлов. Если я попробую внести вредоносный код в ваши сервисы, тесты должны на это отреагировать.

Продумайте юзер интерфейс – пользователь не должен испытывать трудностей при работе с вашей программой!

WARN!!! При проектировании своей DB in memory, запрещено пользоваться Java Collection.

Task 5

Реализация generic set

Реализовать класс MathSet<Number>

класс должен иметь следующие конструкторы:

1. MathSet()
2. MathSet(int capacity)
3. MathSet(Number[] numbers)
4. MathSet(Number[] ... numbers)
5. MathSet(MathSet numbers)
6. MathSet(MathSet ... numbers)

класс должен иметь следующие методы:

1. void add(Number n)
2. void add(Number ... n)
3. void join(MathSet ms)
4. void join(MathSet ... ms)
5. void intersection(MathSet ms)
6. void intersection (MathSet ... ms)
7. void sortDesc()
8. void sortDesc(int firstIndex, int lastIndex)
9. void sortDesc(Number value)
10. void sortAsc()
11. void sortAsc(int firstIndex, int lastIndex)
12. void sortAsc(Number value)
13. Number get(int index)
14. Number getMax()
15. Number getMin()
16. Number getAverage()
17. Number getMedian()
18. Number[] toArray();
19. Number[] toArray(int firstIndex, int lastIndex);
20. MathSet cut(int firstIndex, int lastIndex)
21. void clear()
22. void clear(Number[] numbers)

Warning!

Использование классов и методов Collection и его наследников, Arrays и Stream запрещено!

Task 6

Реализовать консольное приложение Calendar

Вы должны реализовать приложение, которое предоставит пользователю следующие возможности:

- находить разницу между датами в миллисекундах, секундах, минутах, часах, днях и годах;
- прибавлять к дате миллисекунды, секунды, минуты, часы, дни и года;
- вычитать из даты миллисекунды, секунды, минуты, часы, дни и года;
- сравнивать перечень дат по убыванию и возрастанию;

Отсчет времени начинать с 1 января 0 года! Даты можно вводить в следующих строковых форматах:

- 1/10/34 - система преобразует в 1 октября 34 года 0 часов 0 минут 0 секунд 0 миллисекунд;
- /5/47 00:24:00:000 - система преобразует в 1 мая 47 года 0 часов 24 минуты 0 секунд 0 миллисекунд;
- /2/ :2 - система преобразует в 1 февраля 0 года 2 часа 0 минут 0 секунд 0 миллисекунд;
- 1256 14:59 - система преобразует в 1 января 1256 года 14 часов 59 минут 0 секунд 0 миллисекунд

Также пользователь может указать формат даты для ввода и вывода:

- dd/mm/yy - 01/12/21
- m/d/yyyy - 3/4/2021
- mmm-d-yy - Март 4 21
- dd-mmm-yyuu 00:00 - 09 Апрель 789 45:23

Как видите, мы имеем комбинации из знаков / и -, а также индикаторов d, dd, m, mm, mmm, yy, yyy. Вы должны предоставить пользователю возможность менять формат работы с датой на любом этапе жизненного цикла программы. Что касается формата 00:00:00:000 – миллисекунды всегда округляются до 3! Если пользователь вводит 00:00, то программа понимает это как часы:минуты, если 00:00:00 – часы:минуты:секунды! Между блоком день/месяц/год и часы:минуты:секунды:миллисекунды обязан быть пробел! Итак, пользователь может выбрать формат ввода, удобный для него, например - dd/mm/yy 00:00 – это значит, что он должен вводить даты только как 03/03/2021 18:39. Тоже касается и формата вывода, вы должны выводить формат в том виде, в каком попросил пользователь! Форматы не должны зависеть друг от друга, пользователь может выбрать один формат вывода и менять сколько угодно раз формат ввода и наоборот! Вы должны предусмотреть все исключительные ситуации при вводе, расчетах и выводе! Если пользователь ввел некорректный формат даты, вы должны это предусмотреть и обработать возможностями Java и попросить пользователя ввести корректный формат! Если во время расчетов произошла ошибка, программа не должна аварийно завершаться, а - попросить извинения у пользователя и предложить снова выполнить необходимые ему действия

Внимание! При разработке программы, использование классов Date, Calendar, DateTime и прочих реализованных возможностей, - категорически запрещено. Вы также должны учесть в своих расчетах тот простой факт, что существует високосный год!

Task 7

Реализовать data-table web-application on spring/mysql, что такое data-table можете увидеть здесь: <https://datatables.net/> за основу можете взять пример здесь: https://github.com/Iegor-Funtusov/nix_10/tree/master/unit_12_web_data_table_jdbc вы должны реализовать связь на уровне many to many бд, сущности выбираете сами: employee-department, book-author, student-course обязательно создайте скрипты инициализации бд, схема.скл и дата.скл должна быть реализована пагинация и сортировка по каждому полю таблицы + полный круд для обеих сущностей

Task 8-9

Реализовать CRUD web application

1. Вы обязаны взять две любые сущности (преподаватель-студент, клиент-банк и т.д) и реализовать релейшн многие-ко-многим
2. Вы обязаны реализовать веб UI слой для взаимодействия с вашей БД. Должна быть представлена возможность просмотра всех сущностей, создание новой сущности, обновления существующей сущности, с удалением будьте внимательны (мы все равно берем ситуации из жизни, поэтому если мы удаляем студента из университета, то мы его и удаляем из всех курсов, а вот книгу удалить не так то и просто))))
3. В вашем приложении должна быть возможность просмотра всех сущностей А по сущности В, так и сущности В по сущности А
4. Обязательно реализуйте data table concept – должна быть представлена пагинация и сортировка по полям таблицы
5. Создайте пожалуйста пользователя root с паролем rootroot (дабы мне не пришлось под каждую домашку делать лишние телодвижения)
6. В качестве источника хранения данных берем MySQL or Postgre
7. Обязательно предусмотрите скрипт инициализации вашей БД, я не буду под каждого студента ручками заполнять таблички
8. В качестве источника общения с БД используете JPA (hw_8_jpa) and Hibernate (hw_9_hibernate) пример смотрим здесь https://github.com/Iegor-Funtusov/nix_8/tree/master/unit_13_web_jpa