**Лабораторные работы ПнаЯВУ**



1. ~~Intro   
   1. Создать и запустить локально простой веб/REST сервис, используя любой открытый (например, в открытом доступе в сети  интернет) пример с использованием Java stack: Spring (Spring Boot)/maven/gradle/Jersey/ Spring MVC. 2. Добавить GET ендпоинт, принимающий входные параметры в качестве queryParams в URL и возвращающий результат в виде JSON согласно варианту.~~
2. ~~Error logging/handling   
   1. Добавить валидацию входных параметров с возвращением 400 ошибки, 2. Добавить обработку внутренних unchecked ошибок с возвратом 500 ошибки 3. Добавить логирование действий и ошибок 4. Написать unit test~~
3. ~~Collections intro, project structure    
   1. Добавить простейший кэш в виде in-memory Map для сервиса. Map должна содержаться в отдельном бине/классе, который должен добавляться в основной сервис с помощью dependency injection механизм Spring~~
4. ~~Concurrency    
   1. Добавить сервис для подсчёта обращений к основному сервису. Счётчик должен быть реализован в виде отдельного класса, доступ к которому должен быть синхронизирован. 2. Используя jmeter/postman или любые другие средвста сконфигурировать нагрузочный тест и убедиться, что счётчик обращений работает правильно при большой нагрузке.~~
5. ~~Functional programming with Java 8   
   1. Преобразовать исходный сервис для работы со списком параметров для bulk операций используя Java 8 лямбда выражения. 2. Добавить POST метод для вызова bulk операции и передачи списка параметров в виде JSON~~
6. ~~Functional filtering and mapping   
   1. Добавить агрегирующий функционал (подсчёт макс, мин, средних значений) для входных параметров и результатов с использованием Java 8 map/filters функций. Расширить результат POST соотвественно.~~
7. ~~Data persistence   
   1. Добавить возможность сохранения всех результатов вычислений в базе данных или файле, используя стандартные persistence фреймворки Java (Spring Data/Hibernate/MyBatis)~~
8. ~~Asynchronous calls   
   1. Добавить возможность асинхронного вызова сервиса используя future, возвращать статус вызова REST сервиса, не дожидаясь результатов подсчётов. Результаты подсчётов должны быть представлены в БД по предопределённой ID~~

   
   
   
   
   
   
   
**Варианты исполнения сервисов:**   
1. Сервис должен принимать 4 параметра (числа) и вернуть среднее и медианное значение для предоставленных параметров   
2. Сервис должен принимать два параметра (длина и высота) и вернуть периметр и площадь прямоугольника для предоставленных параметров   
3. Сервис должен принимать три параметра (сторона А, сторона Б, сторона В) и вернуть периметр и площадь треугольника для предоставленных параметров   
4. Сервис должен принимать три параметра (сторона А, сторона Б, сторона В) и вернуть свойства треугольника (является ли треугольник равносторонним, равнобедренным, прямоугольным) для предоставленных параметров.   
5. Сервис должен принимать три параметра (значение 1, значение 2, операция («+», «-», «\*», «/»)) и вернуть результат операции для предоставленных параметров.   
6. Сервис должен принимать один параметр (число) и вернуть результат в виде одного случайного числа больше предоставленного параметра и одно случайное число меньше предоставленного параметра   
7. Сервис должен принимать два параметра (значение 1, значение 2) и вернуть результаты сложения, вычитания, произведения и деления предоставленных параметров.   
8. Сервис должен принимать один параметр (число) и вернуть результаты передода предоставленного числа из метров в дюймы и из дюймов в метры.   
9. Сервис должен принимать один параметр (строка) и вернуть результат инвертирования предоставленной параметром строки.   
10. Сервис должен принимать один параметр (строка) и вернуть результат проверки предоставленной параметром строки на полином («полином»/»не полином»).   
11. Сервис должен принимать четыре параметра (значение a, значение b, начало отрезка, конец отрезка) и вернуть результат решения уравнения «х + a = b», где a – значение a, b – знаение b, методом подбора, выбирая корень из значений отрезка предоставленного параметрами «начало» и «конец отрезка»   
12. Сервис должен принимать два параметра (начало отрезка, конец отрезка) и вернуть результат интегрирования sin(x)dx методом трапеций для предоставленного отрезка   
13. Сервис должен принимать один параметр (десятичное число) и вернуть результат конвертирования предоставленного числа из десятичной в двоичную форму   
14. Сервис должен принимать один параметр (год) и вернуть результат определения количества дней в предоставленном году и результат определения, является ли предоставленный год високосным («високосный»/ «не високосный»)   
15. Сервис должен принимать два параметра (год и номер дня в году) и вернуть результат определения дня недели («понедельник», «вторник» и т.д.) для предоставленного дня в предоставленном году   
16. Сервис должен принимать один параметр (число) и вернуть пять случайных чисел от 0 до предоставленного в параметре числа   
17. Сервис должен принимать два параметра (строку, символ (буква или цифра)) и вернуть результат подсчёта количество вхождений символа в предоставленной строке   
18. Сервис должен принимать один параметр (число) и вернуть результат проверки предоставленного числа на простоту («простое» - «не простое») и чётность («чётное» - «не чётное»)   
19. Сервис должен принимать один параметр (порядковый номер в ряду Фибоначчи) и вернуть результат нахождения в ряду Фибоначчи числа с предоставленным порядковым номером   
20. Сервис должен принимать четыре параметра (масса тела А, скорость тела А, масса тела Б, скорость тела Б) и вернуть результат вычисления скорости после неупругого столкновения тел с предоставленными параметрами   
21. Сервис должен принимать два параметра (расстояние, скорость) и вернуть время прямолинейного движения для прохождения предоставленного расстояния с предоставленной скоростью   
22. Сервис должен принимать три параметра (значение 1, значение 2, значние 3) и вернуть максимальное из предоставленных значений   
23. Сервис должен принимать три параметра (координаты x1, y1, x2, y2) и вернуть значения нормы вектора и проекций на оси x и y для предоставленных координат   
24. Сервис должен принимать два параметра (строку, операцию («зашифровать» или «расшифоровать»)) и вернуть результат шифрования (или расшифрования) предоставленной строки   
25.  Сервис должен принимать два параметра (масса раствора, процент сухого вещества) и вернуть массу сухого вещества в растворе согласно предоставленным параметрам   
26. Сервис должен принимать два параметра (высота цилиндра, радиус) и вернуть результат подсчёта объёма цилиндра   
27. Сервис должен принимать два параметра (действительная часть числа, мнимая часть числа) и вернуть результат конвертации комплексного числа из алгебраической формы в показательную (модуль и аргумент(фаза)) для предоставленных параметров   
28. Сервис должен принимать один параметр (количество блужданий) и вернуть результат подсчёта случайного блуждания   
29. Сервис должен принимать один параметр (угол в градусах) и вернуть результат перевода градусов угла в радианы   
30. Сервис должен принимать два параметра (делимое и делитель) и вернуть результат деления предоставленных параметров (частное, остаток)