**Цели домашнего задания**

* Научиться работать со всеми типами данных в Dart.
* Научиться создавать собственные типы.
* Получить практический опыт работы с наследованием, миксинами, методами расширения и исключениями.
* Научиться логически мыслить и интерполировать логику на программный код.

**Что нужно сделать**

Реализуйте пакет (библиотеку классов), в которую поместите методы и классы, предлагаемые в заданиях ниже. Для тестирования методов можно создать функцию void main(). Каждая задача данного задания познакомит вас с определёнными возможностями языка Dart и научит работе с различными типами данных.

1. Реализуйте методы вычисления НОД и НОК целых чисел. Реализуйте метод, который разбивает число на простые множители и возвращает их.
2. Реализуйте методы для преобразования целых чисел из десятичной системы в двоичную и обратно.
3. Реализуйте метод, который принимает строку слов, разделённых пробелами. Задача — найти в данной строке числа и вернуть коллекцию num этих чисел.
4. Есть коллекция слов. Реализуйте метод, возвращающий Map. Данный Map должен соотносить слово и количество его вхождений в данную коллекцию.
5. Есть коллекция строк вида ‘one, two, three, cat, dog’ или любого другого. Реализуйте метод, возвращающий цифры без повторений, которые встречаются в данной строке. Однако цифры встречаются в виде английских слов от zero до nine. Например, в результате строки ‘one, two, zero, zero’ мы получим следующий результат: [1, 2, 0]. Если в строке есть слова, не являющиеся цифрами от 0 до 9, пропускайте их.
6. Реализуйте класс Point, который описывает точку в трёхмерном пространстве. У данного класса реализуйте метод distanceTo(Point another), который возвращает расстояние от данной точки до точки в параметре. По желанию можете реализовать метод, принимающий три точки в трёхмерном пространстве и возвращающий площадь треугольника, который образуют данные точки. Реализуйте factory-конструкторы для данного класса, возвращающие начало координат, и ещё несколько на своё усмотрение (например, конструктор, возвращающий точку с координатами [1,1,1], которая определяет единичный вектор).
7. Реализуйте метод, который вычисляет корень любой заданной степени из числа. Реализуйте данный метод как extension-метод для num. Алгоритм можете взять на википедии как «Алгоритм нахождения корня n-й степени». Запрещается использовать методы math. В случае когда значение вернуть невозможно, необходимо бросать исключение с описанием ошибки.
8. Создайте класс User, у которого есть поле email. Реализуйте два наследника данного класса AdminUser и GeneralUser. Реализуйте mixin над классом User, у которого будет метод getMailSystem, который возвращает значение из email, которое находится после @. Например, если email пользователя user@mail.ru, то данный метод вернёт mail.ru. Используйте данный миксин на AdminUser. Далее реализуйте класс UserManager<T extends User>, у которого будет храниться список пользователей и будут методы добавления или удаления их. Также в UserManager реализуйте метод, который выведет почту всех пользователей, однако если пользователь admin, будет выведено значение после @. Проверьте реализованные методы на практике.
9. (Усложнённое). Реализуйте метод, который вычисляет значение определённого интеграла Римана в заданных значениях и с заданной точностью разбиения. По возможности учтите случаи, когда функция имеет точки разрыва. Реализованный метод должен принимать функцию, точки x1 и x2, точность разбиения.
10. (Очень сложно). Реализуйте метод, который принимает коэффициенты системы линейных уравнений (для любого числа неизвестных). Данный метод создаёт матрицу коэффициентов, приводит её к треугольному виду и находит все неизвестные данной системы.

**Советы и рекомендации**

* Каждое задание можно реализовывать в отдельном классе, при этом все задания могут находиться в одном проекте. К примеру, для вычисления НОД и НОК можно реализовать класс DelimetersCalculator и добавить в него необходимые методы.
* Разрешается использовать вспомогательные методы. Главное, не забывать делать их приватными.
* Изучите статьи, которые помогут вам выполнить задание:
  1. [«НОД. НОК. Алгоритм Евклида»](https://brestprog.by/topics/gcd/);
  2. [«Алгоритм нахождения корня n-й степени»](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%BD%D0%B0%D1%85%D0%BE%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BA%D0%BE%D1%80%D0%BD%D1%8F_n-%D0%BD%D0%BE%D0%B9_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%B8).
* Воспользуйтесь сервисом [перевода чисел в двоичную систему счисления и обратно](https://math.semestr.ru/inf/index.php).

**Что оценивается**

* Выполнено минимум восемь заданий.
* Качество кода. Правильно именованы переменные и классы.
* Соблюдён принцип инкапсуляции.
* Оптимизированы реализованные алгоритмы и скорость их работы.
* Длина методов. Методы краткие и помещаются на экран (в идеале).
* Покрытие border cases. Чем сложнее сломать программу, тем лучше.