**Задача 6**

В этой работе нужно чуть больше поработать самостоятельно. Необходимо написать программу, которая соответствует требованиям. Способ реализации тех или иных требований выберите сами, основываясь на пройденном материале.

Цели домашнего задания

Научиться:

* объявлять собственные классы;
* создавать экземпляры классов, вызывать их методы, обращаться к свойствам;
* создавать объекты и работать с ними.

Что нужно сделать

Создайте собственный класс TV, который описывает множество телевизоров и обладает всеми перечисленными характеристиками и действиями.

1. У класса следующие свойства:
   * **марка, модель, размер диагонали** — эти свойства доступны для чтения, есть возможность создавать телевизоры разных моделей и размеров;
   * **включён/выключен** — свойство доступно для чтения и недоступно для записи;
   * **список каналов** — внутреннее свойство телевизора, недоступно для чтения;
   * **максимальная громкость** — внутреннее свойство, значение константное, поэтому располагается в объекте-компаньоне этого класса.
2. Класс обладает следующими функциями:
   * включение/выключение;
   * увеличение громкости на заданную величину;
   * уменьшение громкости на заданную величину;
   * переключение канала на заданный (аналог нажатия кнопки с цифрой канала на пульте); если телевизор выключен, то использование этой функции включает телевизор и переключает канал на заданный;
   * переключение каналов в порядке возрастания и убывания (аналог нажатия кнопок переключения каналов «Вверх» и «Вниз» на пульте); если телевизор выключен, то использование этих функций включает телевизор на канале, который был до выключения; переключение каналов непрерывно: после последнего канала включается первый из списка;
   * представление списка каналов в виде $порядковый\_номер — $название\_канала для отображения.
3. При создании объекта класса «Телевизор» проводится настройка каналов. Список каналов заполняется случайными неповторяющимися каналами. Для получения случайных каналов объявите собственный object Channels. Он должен содержать список всех возможных каналов и функцию getRandomChannels, которая будет возвращать список случайных каналов. Можете реализовать эту функцию самостоятельно. Один из вариантов реализации:
   * с помощью [функции shuffled](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/shuffled.html) перемешайте список каналов;
   * выберите оттуда необходимое для каждого телевизора количество.
4. Списки каналов у разных телевизоров различаются. Величина списка каналов у телевизоров может быть разной.
5. Каналы представлены строками с названием. По желанию создайте дополнительный класс Channel для описания канала.
6. Напишите программу, которая создаёт несколько экземпляров телевизоров.
   * Для каждого экземпляра выведите информацию об этом телевизоре.
   * Для каждого экземпляра вызовите все доступные функции и убедитесь, что они работают корректно. Для этого в каждую вызываемую функцию добавьте вывод в консоль информации о текущих значениях полей экземпляра. Например, в функцию увеличения громкости добавьте вывод обновлённого значения текущей громкости.
   * Используйте циклы для проверки функций работы с громкостью и функций последовательного переключения каналов.
   * Все вызовы функций корректно обрабатываются, если телевизор выключен (если не сказано обратного).

**Задания по желанию**

1. Создайте свой класс, который описывает выбранный вами объект или явление.
   * Определите, какой объект или явление будет описывать этот класс.
   * Установите, какие характеристики и свойства есть у описываемого объекта.
   * Определите, какие действия должны быть доступны для этого объекта.
2. Опишите в этом классе необходимые поля и реализуйте функции.
3. Создайте экземпляр этого класса и убедитесь, что всё функционирует корректно.
4. Перечислите, что описывает созданный вами класс и какие функции он выполняет. Приложите текст через форму для сдачи домашнего задания, чтобы проверяющий преподаватель смог оценить, насколько созданный вами класс соответствует описанию.

Для контроля корректности выполнения программы используйте функцию Thread.sleep(millis: Long). Вызов этой функции приостанавливает выполнение программы на переданное количество миллисекунд. Спустя это время программа продолжает своё выполнение с того места, на котором остановилась. С помощью этой функции можно регулировать скорость выполнения программы и скорость вывода информации в консоль. Это позволит вам контролировать ход выполнения программы.

* Thread.sleep(1000) останавливает программу на одну секунду.
* Thread.sleep(5000) останавливает программу на пять секунд.

Рекомендации по выполнению

Воспользуйтесь документацией:

* [классы](https://kotlinlang.org/docs/classes.html);
* [объект-синглтон](https://kotlinlang.org/docs/object-declarations.html#object-declarations-overview) и [объект-компаньон](https://kotlinlang.org/docs/object-declarations.html#companion-objects);
* [функция shuffled](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/shuffled.html);
* [модификаторы доступа](https://kotlinlang.org/docs/visibility-modifiers.html#classes-and-interfaces).

Что оценивается

* Программа запускается, работает корректно и выводит всю необходимую информацию.
* Выполнены все обязательные пункты задания.
* Функции соответствуют принципам чёрного ящика, выполняют атомарные операции и независимы друг от друга.
* Функции возвращают неизменяемые коллекции.
* Названия переменных отражают суть данных, на которые они ссылаются.
* Переменные, которые не изменяются в программе, объявлены неизменяемыми.
* Внутренние члены класса — с модификатором private.
* Различные выводы в консоль начинаются с новой строки.

*Плохо:* «Вывод информации 1. Вывод информации 2».

*Хорошо:*

«Вывод информации 1.

Вывод информации 2».

Как отправить задание на проверку

* Используйте репозиторий dpo\_kotlin.
* Скачайте изменения в репозитории на локальную машину.
* Выполните домашнее задание в папке 07\_ClassesAndObjects. Создайте проект внутри папки, сделайте коммит или несколько коммитов. Отправьте коммиты в удалённый репозиторий.