**Задание 1**

Исходный объект:

|  |
| --- |
| let obj = {  num: 1.24,  str: “not very long string”,  f() {  return this.str.split(“ ”)  },  arr: [“some”, “array”, {someProp: “value”}],  prop: { key: 1 },  empty: null,  last: 0  }; |

Написать функцию, передав в которую исходный объект, мы получим новый объект вида:

|  |
| --- |
| {  “количество собственных свойств”: 7,  propTypes: [“number”, “string”, “function”, …, “number”],  propNames: [“num”, “str”, …, “last”]  } |

Исходный объект после отрабатывания функции изменяется:

1. все числовые значения свойств *первого уровня (\*)* преобразуются в числа с плавающей точкой с двумя знаками после точки;
2. все строковые значения *первого уровня* преобразуются к верхнему регистру;
3. объект становится не расширяемым (нельзя добавлять новые свойства).

Исходный объект является примером. Функция может принимать любой объект и соответственно менять возвращаемое значение.

**Задание 2**

Создать объект, который будет описывать человека. А именно, студента LevelUp. У Объекта есть следующие свойства:

* имя студента (строка)
* фамилия (строка)
* возраст (число)
* изучаемые предметы/курсы (массив объектов, где каждый объект - это описание отдельного курса. См. вложенный список)
  + Название курса (строка)
  + Имя преподавателя (строка)
  + Продолжительность курса в часах (число)
  + Пройдено курса (число от 0 до 1)
  + Оценки (массив чисел от 1 до 5)

Кроме этих свойств, объект содержит ряд методов:

1. getFullName - метод возвращает строку, в которой указаны имя и фамилия студента
2. getAge - метод возвращает строку с возрастом (например, “31 years old”)
3. getCourses - возвращает строку с курсами/предметами, перечисленными через запятую (например, “frontend, python, php”)
4. addNewCourse - принимает параметры: *teacherName* (строка), *course* (строка), *duration* (число). Добавляет новый курс в массив курсов студента.
5. getAvarageMarkByCourse - принимает параметр *course* (строка), и возвращает среднее арифметическое из оценок в курсе. Или *null*, если в курсе нет оценок.
6. getAvarageMark - возвращает среднее арифметическое из всех оценок во всех курсах
7. addMark - принимает параметры *course* (строка), *mark* (число) и добавляет оценку в массив оценок курса. Комментарий
8. addProgress - принимает параметры *course* (строка) и *hours* (число) и возвращает прогресс прохождения курса (число от 0 до 1). Комментарий
9. getProgress - возвращает строку с информацией о том, какая часть курса уже пройдена (например, “33%”)