# F. Тараканы, вперед! (60 баллов)

Владимир — тренер беговых тараканов. Его воспитанники занимают призовые места на всех соревнованиях. Для качественной тренировки Владимиру нужен надёжный и удобный секундомер.

* У секундомера должно быть две стрелки: секундная и минутная. Cтрелки должны двигаться не постоянно, а только в момент переключения периода, который они отсчитывают. Например, секундная стрелка должна перемещаться на одно деление каждую секунду.
* Должна быть кнопка запуска и остановки секундомера. При первом нажатии секундомер начинает отсчёт времени. При втором — останавливается.
* Должна быть ещё вторая кнопка, с помощью которой можно запоминать текущее значение времени. Она работает в двух режимах: 1) если секундомер запущен, запоминает текущее время (если нажать несколько раз, будет запомнено несколько значений времени) 2) если секундомер остановлен, то сначала передвигает стрелки на 0, а каждое следующее нажатие циклически переключает стрелки между запомненными значениями (включая 0).
* При переводе стрелок можно двигать их по часовой стрелке или против неё (по наикратчайшему пути).
* После переключения в запомненное значение и запуска движения стрелки продолжают идти от этого значения, а все запомненные значения стираются.

Пожалуйста, реализуйте секундомер, используя js-фреймворк для разработки часов под названием Framework, и внесите свой вклад в победу тараканов Владимира.

### Пример кода прошивки секундомера

const ONE\_SECOND\_DEGREES = 6;

const ONE\_SECOND\_FACTOR = 1 / Framework.SPEED \* ONE\_SECOND\_DEGREES;

class MyClock extends Framework.Clock {

constructor() {

super();

this.arrows.push(new Framework.Arrow("seconds", {

color: "red"

}));

this.arrows.push(new Framework.Arrow("minutes", {

weight: 3,

length: 80

}));

this.buttons.push(new Framework.Button("A", () => {

alert("A");

}));

this.tick = 0;

}

onBeforeTick() {

const [arrow] = this.arrows;

this.tick++;

arrow.rotateFactor = this.tick % 10 ? 0 : ONE\_SECOND\_FACTOR;

console.log("before: " + arrow.pos);

}

onAfterTick() {

const [arrow] = this.arrows;

console.log("after: " + arrow.pos);

}

}

## Пример

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| {  "comment": "положение стрелок",  "steps": [  {  "ticks": 10  },  {  "button": 0,  "ticks": 200  },  {  "button": 1,  "ticks": 50  },  {  "button": 1,  "ticks": 300  },  {  "button": 0,  "ticks": 50  },  {  "button": 1,  "ticks": 100  },  {  "button": 1,  "ticks": 100  },  {  "button": 1,  "ticks": 100  },  {  "button": 1,  "ticks": 100  }  ]  } | [  {  "seconds": 0,  "minutes": 0  },  {  "seconds": 120,  "minutes": 0  },  {  "seconds": 150,  "minutes": 0  },  {  "seconds": 330,  "minutes": 0  },  {  "seconds": 330,  "minutes": 0  },  {  "seconds": 0,  "minutes": 0  },  {  "seconds": 120,  "minutes": 0  },  {  "seconds": 150,  "minutes": 0  },  {  "seconds": 0,  "minutes": 0  }  ] |

## Примечания

Откройте HTML-файл тестовой страницы по ссылке «Скачать условие задачи» в конце описания. Вам нужно написать на JavaScript класс с названием MyClock, который реализует поведение, описанное в условии.

class MyClock extends Framework.Clock {

// ваш код

}

При проверке, файл с вашим решением будет подключен на тестовую страницу в место, обозначенное комментарием:

<!-- в качестве решения предоставьте файл solution.js -->

<script src="solution.js"></script>

Идентификаторы стрелок (первый параметр их конструктора) должны быть такими же, как в примере: "seconds", "minutes".

Порядок кнопок должен быть:

* кнопка запуска/остановки секундомера (индекс = 0)
* кнопка запоминания/переключения значений (индекс = 1)

Ваше решение будет тестироваться в браузере Google Chrome 77.

[Скачать условие задачи](https://contest.yandex.ru/contest/14227/download/F/)