

Задача 1

Реализовать модуль управления банком.

```
1  const bank = new Bank();
2
3  const personId = bank.register({
4    name: 'Pitter Black',
5    balance: 100
6  });
7
8  bank.emit('add', personId, 20);
9  bank.emit('get', personId, (balance) => {
10     console.log(`I have ${balance}€`); // I have 120€
11  });
12  bank.emit('withdraw', personId, 50);
13  bank.emit('get', personId, (balance) => {
14     console.log(`I have ${balance}€`); // I have 70€
15  });
```

1. Реализовать метод **register** который будет регистрировать нового контрагента. Метод возвращает идентификатор контрагента (генерируется автоматически модулем Bank).

```
1  {
2    name*: String, // имя контрагента
3    balance*: Number // начальный баланс
4  }
```

2. Реализовать поддержку события **add** которое принимает идентификатор контрагента в качестве второго аргумента и сумму зачисления в качестве третьего аргумента. Событие должно добавить деньги на счёт клиенту.
3. Реализовать поддержку события **get** которое принимает идентификатор контрагента в качестве второго аргумента и колбек функцию в качестве третьего аргумента. Callback принимает один аргумент **balance**, который указывает на количество денег на момент генерации события.
4. Реализовать поддержку события **withdraw** которое принимает идентификатор контрагента в качестве второго аргумента и сумму списания в качестве третьего аргумента. Событие должно списать деьги со счёта контрагента.

Обратите внимание!

1. Метод **register** должен вызывать ошибку при попытке добавления двух контрагентов с одинаковыми именами.
2. Метод **register** должен вызывать ошибку при попытке добавления контрагента со значением **balance** меньше или равным нулю.

3. Событие **add** должно генерировать ошибку при попытке зачислить отрицательную или нулевую сумму на счёт контрагента.
4. Событие **withdraw** должно генерировать ошибку при попытке списать отрицательную сумму со счёта контрагента.
5. Событие **withdraw** должно генерировать ошибку при попытке списать сумму со счёта контрагента если после списания счёт контрагента будет меньше нуля.
6. Все события должны генерировать ошибку если им передан не существующий идентификатор контрагента.
7. Генерирование и обработку ошибок необходимо реализовать при помощи события **error**.

```
1 | .emit('error');  
2 | .on('error')
```