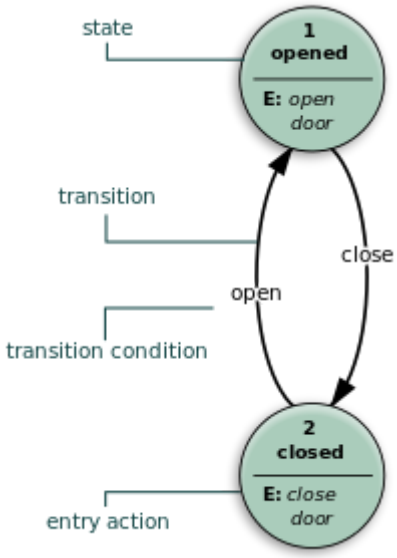
Принцип работы стейт-машины

Стейт-машина состоит из двух вещей: **состояний** и **переходов**. Вот простейший пример

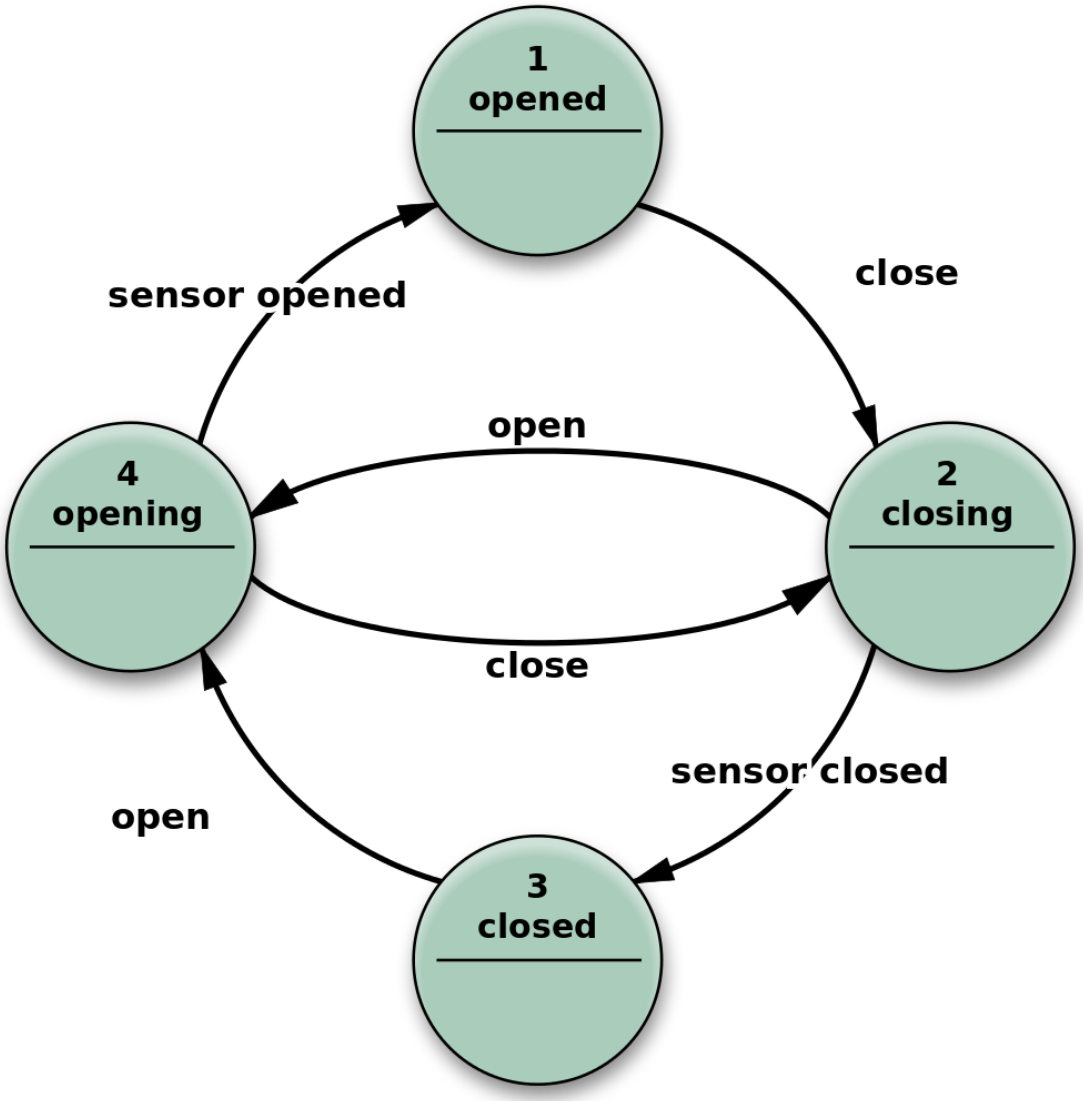


стейт-машины, **дверь**:

У двери есть два состояния: *открыта* и *закрыта*. И два перехода между ними: *открытие двери* и *закрытие двери*. Как вы видите, почти всё что угодно можно описать как стейт-машину.

Стейт-машина — это паттерн проектирования, шаблон, в который можно подставлять разный смысл. Состояниями и переходами можно описать дверь, человека, магазин или компьютерную программу. Написав стейт-машину для одной программы, её можно практически без изменений скопировать в другую, похожую.

Сильное преимущество этого паттерна — расширяемость. Например, дверь можно описать подробнее:



Появились два новых состояния, теперь дверь может быть:

- Открыта
- Закрыта
- В процессе открытия (тянем за ручку, уже не закрыта, но ещё не открылась полностью)
- В процессе закрытия (уже не открыта, но почти захлопнулась)

Как выглядит в коде?

Вот пример эхобота на стейт-машине. Все запросы к нему попадают в функцию `handle_users_reply`.

Функция получает `chat_id` пользователяи фразу, которую он сказал:

```
if update.message:
    users_reply = update.message.text
    chat_id = update.message.chat_id
else:
    users_reply = update.callback_query.data
    chat_id = update.callback_query.message.chat_id
```

Затем, если пользователь впервые, она выставляет ему стейт `START`:

```
if users_reply == '/start':
    users_state = 'START'
```

Если же пользователь уже работал с ботом, его стейт хранится в базе данных:

```
else:
    users_state = DATABASE.get(chat_id).decode("utf-8")
```

Далее идёт словарь с состояниями и их обработчиками. Состояние `START` обрабатывает функция `start`, а состояние `ECHO` обрабатывает функция `echo`:

```
states_functions = {
    'START': start,
    'ECHO': echo
}
```

Далее получаем функцию, которая обрабатывает состояние пользователя:

```
state_handler = states_functions[users_state]
```

Запускаем её, получаем в ответ следующее состояние:

```
next_state = state_handler(bot, update)
```

И записываем за пользователем следующее состояние:

```
DATABASE.set(chat_id, next_state)
```