

Bigtrip

Сперва разберёмся с кодом, который скопился у нас в `TripController` и отвечает за смену точки маршрута на форму редактирования. Для этого создадим `PointController`:

1. В нем нужно описать конструктор и метод `render`
2. Конструктор должен принимать `container` — элемент, в который контроллер будет всё отрисовывать.
3. Метод `render` должен принимать данные одной точки маршрута. Также в него должен переехать код, который отвечает за отрисовку точки маршрута, ее замену на форму редактирования и наоборот, а также установка связанных с этим обработчиков событий.

Настроим частичный датабиндинг

Реализуем обработку кликов на кнопке «Favorite» (звёздочка) в форме редактирования. Обработчики должны изменять данные точки маршрута — добавлять или удалять из избранного, и на основе изменённых данных перерисовывать компонент.

Для этого:

1. В компоненте формы редактирования добавьте метод для установки обработчика клика для звёздочки.
2. В конструкторе `PointController` добавьте новый аргумент — функцию `onDataChange`. Эта функция должна вызываться в обработчике клика и получать на вход старые и новые данные (точку маршрута и измененную точку маршрута).

Обратите внимание, пока что мы занимаемся обновлением только признаков точки маршрута, поэтому при сохранении у нас по-прежнему форма просто заменяется на точку маршрута, а удаление не работает вовсе. С этим мы разберёмся в следующих заданиях.

3. В `TripController` опишите метод `_onDataChange` с точно таким же интерфейсом, как функция `onDataChange`. Задача метода — обновить моки и вызывать метод `render` у конкретного экземпляра `PointController` с обновлёнными данными.
4. Чтобы всё заработало, передайте метод `_onDataChange` в `PointController` при создании его экземпляра.

Добавим интерактивности

Форма редактирования довольно сложный интерактивный компонент. Но это поведение — не часть бизнес-логики приложения. Это бизнес-логика самого компонента. Поэтому для реализации этой логики введем понятие `SmartComponent` — компонент, который может себя перерисовывать.

1. Создайте абстрактный класс `AbstractSmartComponent`, унаследовав его от `AbstractComponent`, с двумя методами:

- абстрактный метод `recoveryListeners`, его нужно будет реализовать в наследнике. Его задача — восстанавливать обработчики событий после перерисовки;
- обычный метод `rerender`, его задачи:
 - удалить старый DOM-элемент компонента;
 - создать новый DOM-элемент;
 - поместить новый элемент вместо старого;
 - восстановить обработчики событий, вызвав `recoveryListeners`.

2. Унаследуйте компонент формы редактирования от `AbstractSmartComponent` и объявите в компоненте метод `recoveryListeners`, пока пустым.
3. Теперь нужно реализовать перерисовку формы редактирования после взаимодействия с пользователем:
 - при смене типа точки маршрута нужно менять плейсхолдер у пункта назначения и показывать соответствующий типу набор дополнительных опций;
 - при выборе пункта назначения нужно менять описание.

Обратите внимание, что при смене дополнительных опций выбранные значения не сохраняются, а ещё выбор дополнительных опций не влияет на стоимость, указанную в соответствующем поле ввода.

4. При перерисовке компонента все обработчики событий будут утеряны, поэтому их нужно навесить заново в методе `recoveryListeners`.

Отообразим только одну форму

Мы научились обновлять данные и создавать интерактивные компоненты. Осталось запретить открывать несколько форм редактирования одновременно. Мы реализуем это простым способом: в `TripController` мы «прикажем» всем экземплярам `PointController` вернуться в исходное

состояние, когда пользователь открывает форму редактирования.

1. Добавьте метод `setDefaultView` в `PointController` для отображения точки маршрута вместо формы редактирования.
2. Добавьте в конструктор `PointController` новый аргумент — функцию `onViewChange` и вызывайте её перед тем, как сменить точку маршрута на форму редактирования.
3. В `TripController` опишите метод `_onViewChange`, где вызовете у всех экземпляров `PointController` метод `setDefaultView`.
4. В `TripController` при создании экземпляров `PointController` передайте и метод `_onViewChange`.