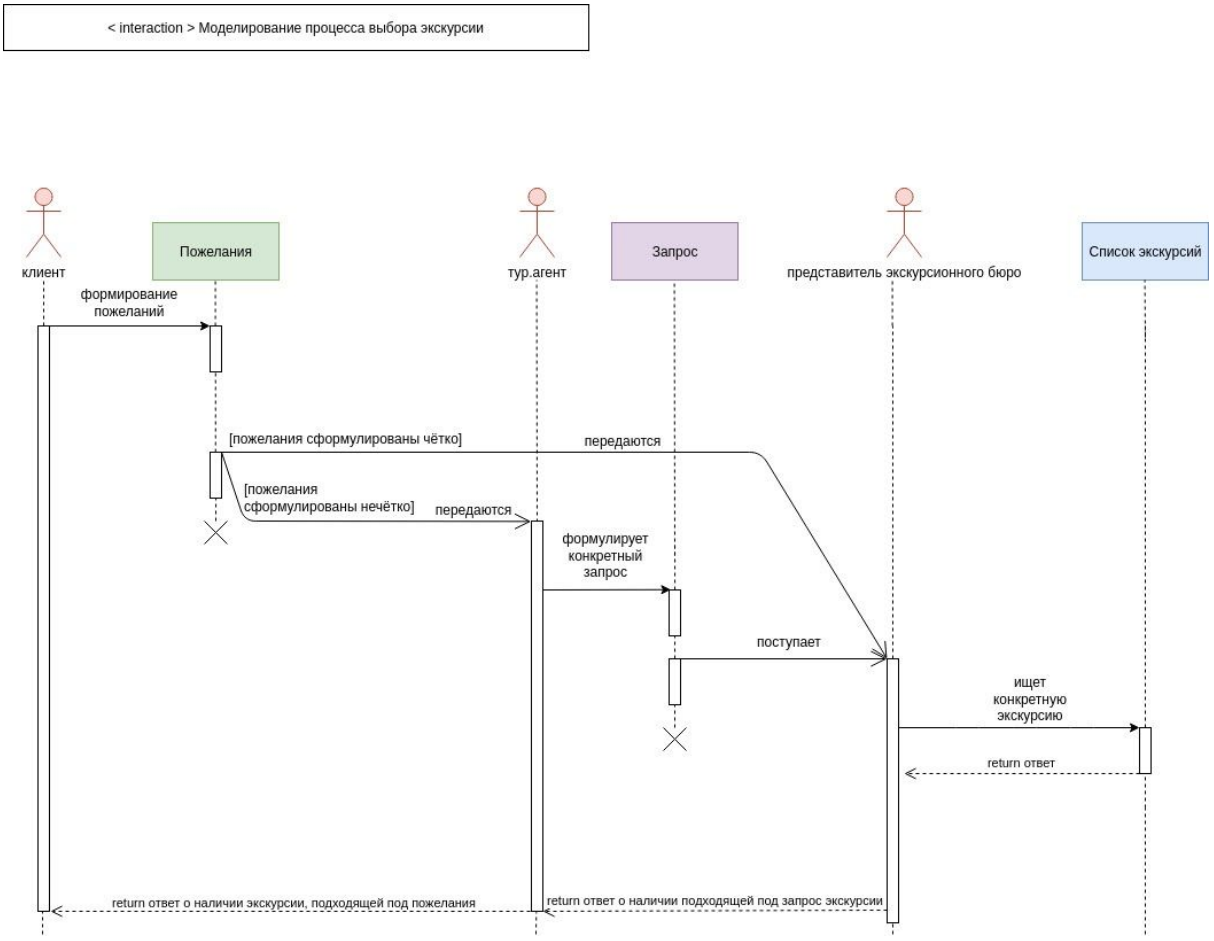
Диаграмма последовательности №1:



Описание сущностей и передачи сообщений между ними

Клиент

При покупке тура нужно выбрать множество его составляющих, одна из которых - экскурсии, которые будут включены в тур. Поэтому в данной диаграмме первая сущность - “клиент”. Она создаётся посредством использования линии жизни, а далее - активации, т.е. на линии жизни появляется фокус управления. Т.к. клиент существует на протяжении всего существования диаграммы, то фокус управления имеет длину, равную длине диаграммы. Клиент придумывает пожелания посредством создания сообщения с подписью “формирование пожеланий”. Оно выходит из правого верхнего угла фокуса управления клиента и приходит в левый верхний угол фокуса управления сущности “пожелания”.

Клиент передаёт сообщение с помощью стрелки procedure call, что означает, что клиент посылает данное сообщение объекту-приемнику “пожелания” - и, чтобы

продолжить свою работу, ожидает, пока вся дальнейшая вложенная последовательность сообщений не будет обработана приемником.

Пожелания

Сущность пожелания начинается с линии жизни. Когда фокус управления передаётся из “клиента” в “пожелания”, то появляется фокус управления.

Пожелания могут быть сформулированы клиентом вполне конкретно, а могут быть крайне туманными. Следовательно, в данном месте диаграммы появляется ветвление: если пожелания чёткие, значит, они не требуют дополнительной обработки и передаются сразу представителю экскурсионного бюро, а если пожелания сформулированы расплывчато, то сначала их обрабатывает и структурирует тур.агент, создаёт запрос, запрос передаётся представителю экскурсионного бюро. Как это выглядит на диаграмме: Из фокуса управления сущности “пожелания” выходит две ветки. Из них выполняется только одна в соответствии со сторожевым условием (как сформулированы пожелания - чётко/нечётко).

Сообщения передаются из “пожеланий” с помощью стрелки simple, что означает, что данные сообщения выполняются в одном потоке управления. Это свойство задается добавляемым на диаграмму сообщениям по умолчанию.

Фокус управления больше никогда не будет отдан сущности “пожелания”, поэтому данная сущность уничтожается после передачи фокуса управления другой сущности.

Тур.агент

Сущность “тур.агент” начинается с линии жизни. Когда фокус управления передаётся из сущности “пожелания” в “тур.агента”, то появляется фокус управления.

“Тур. агент” получает фокус управления только в том случае, если пожелания сформулированы некорректно. Тогда тур.агент обрабатывает их, структурирует, выделяет главное. Далее - составляет запрос. “Тур.агент” передаёт сообщение с помощью стрелки procedure call, что означает, что “тур.агент” посылает данное сообщение объекту-приемнику “запрос” - и, чтобы продолжить свою работу, ожидает, пока вся дальнейшая вложенная последовательность сообщений не будет обработана приемником.

Запрос

Сущность “запрос” начинается с линии жизни. Когда фокус управления передаётся из сущности “тур.агент” в “запрос”, то появляется фокус управления.

Запрос поступает представителю экскурсионного бюро посредством использования стрелки simple, что означает, что данное сообщение выполняется в одном потоке

управления. Это свойство задается добавляемому на диаграмму сообщению по умолчанию.

Фокус управления больше никогда не будет отдан сущности “запрос”, поэтому данная сущность уничтожается после передачи фокуса управления другой сущности.

Представитель экскурсионного бюро

Сущность “представитель экскурсионного бюро” начинается с линии жизни. Когда фокус управления передаётся из сущности “запрос” в “представитель экскурсионного бюро”, то появляется фокус управления.

“Представитель экскурсионного бюро” принимает сообщение от сущности “пожелания” или “запрос”, в зависимости от того, как развилось ветвление в сущности “пожелания”. Также он находит название предположительно подходящей под критерии экскурсии и передаёт сообщение в сущность “список экскурсий”. Сообщение передаётся с помощью стрелки procedure call, что означает, что представитель экскурсионного бюро посылает данное сообщение объекту-приемнику “список экскурсий” - и, чтобы продолжить свою работу, ожидает, пока вся дальнейшая вложенная последовательность сообщений не будет обработана приемником.

Список экскурсий

Сущность “список экскурсий” начинается с линии жизни. Когда фокус управления передаётся из сущности “представитель экскурсионного бюро” в “список экскурсий”, то появляется фокус управления.

“Список экскурсий” возвращает ответ о наличии запрошенной экскурсии с помощью стрелки с типом сообщения return.

Ремарка:

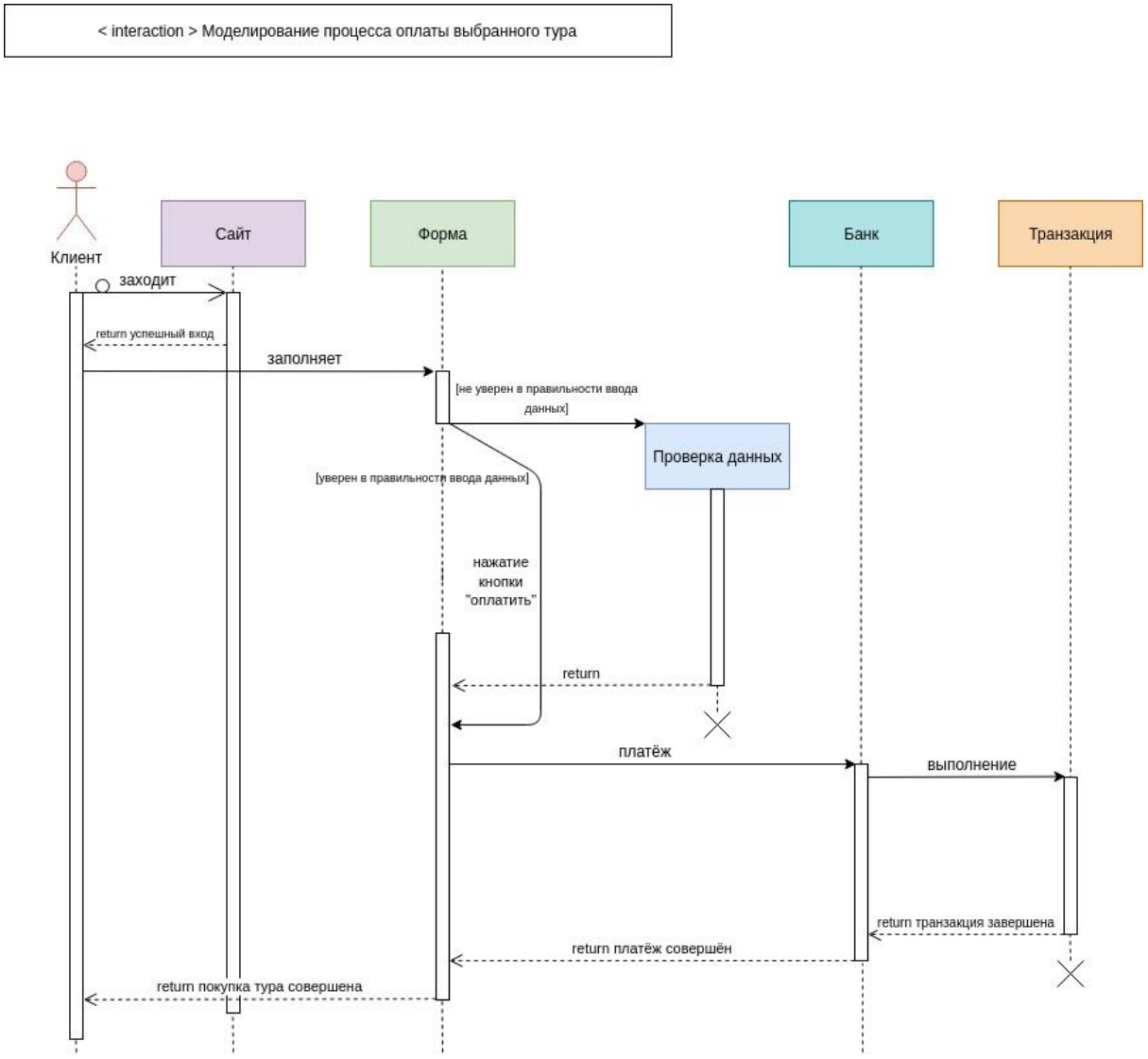
Как описано выше, некоторые сущности отправляли сообщения с помощью стрелки procedure call, использование которой говорит о том, что сущности ждут, когда вложенные операции выполнятся. Ниже будут описаны оставшиеся у сущностей операции, которые они выполняют, когда вложенные действия заканчиваются.

Представитель экскурсионного бюро: посылает сущности “тур.агент” сообщение о наличии запрошенной экскурсии с помощью стрелки с типом сообщения return.

Тур.агент: посылает сущности “клиент” сообщение о наличии экскурсии, походящей под заданные критерия с помощью стрелки с типом сообщения return.

Клиент: принимает от “тур.агента” сообщение о завершении операции.

Диаграмма последовательности №2:



Описание сущностей и передачи сообщений между ними

Клиент

Когда клиент определился с выбором тура, возникает необходимость его оплаты. Поэтому диаграмма под названием “моделирование процесса оплаты выбранного тура” начинается с сущности “клиент”. Она создаётся посредством использования линии жизни, а далее - активации, т.е. на линии жизни появляется фокус управления. Т.к. клиент существует на протяжении всего существования диаграммы, то фокус управления имеет длину, равную длине диаграммы.

Для того, чтобы оплатить тур, клиенту нужно войти на сайт. Для этого клиент передаёт сообщение сайту с помощью стрелки procedure call и типом сообщения timeout, что означает, что клиент посылает данное сообщение объекту-приемнику

“сайт” - и, чтобы продолжить свою работу, ожидает, пока вся дальнейшая вложенная последовательность сообщений не будет обработана приемником. Тип сообщения timeout означает, что после передачи данного сообщения объект-приемник может поместить данное сообщение в очередь с ограниченным временем ожидания, если он занят выполнением других операций.

Сайт

Сущность “сайт” начинается с линии жизни. Когда фокус управления передаётся из “клиента” в “сайт”, то появляется фокус управления.

Данная сущность принимает сообщение типа “с ожиданием” от клиента. Т.к. сайт существует на протяжении всего существования диаграммы, то фокус управления имеет длину, равную длине диаграммы. “Сайт” посылает клиенту сообщение типа return, что означает, что вход на сайт выполнен.

Форма & Клиент

Сущность “форма” начинается с линии жизни. Когда фокус управления передаётся из “клиента” в “форму”, то появляется фокус управления.

Клиент получил ответ от сайта, что вход выполнен. Теперь клиент может заполнить форму, необходимую для оплаты тура. Для этого клиент посылает в сущность “форма” сообщение с типом стрелки procedure call, что означает, что клиент посылает данное сообщение объекту-приемнику “форма” - и, чтобы продолжить свою работу, ожидает, пока вся дальнейшая вложенная последовательность сообщений не будет обработана приемником.

“Форма” принимает сообщение от клиента. Из правого нижнего угла фокуса управления выходят две стрелки - что означает, что в этом месте диаграммы происходит ветвление. Устроено оно так: срабатывает только одна стрелка из двух в соответствии со сторожевым условием. Условие состоит в следующем: если клиент уверен в правильности введённых данных, то он переходит сразу к нажатию кнопки “оплатить”, если клиент не уверен в правильности введённых данных, то происходит создание ещё одной сущности, которая называется “проверка данных”.

Если клиент уверен в правильности введённых данных, то из сущности “форма” в неё же посылается сообщение, в котором говорится о том, что нужно нажать на кнопку “оплатить”. После этого с помощью стрелки procedure call (что означает, что клиент посылает данное сообщение объекту-приемнику “банк” - и, чтобы продолжить свою работу, ожидает, пока вся дальнейшая вложенная последовательность сообщений не будет обработана приемником) происходит передача сообщения “платёж”.

Если клиент не уверен в правильности введённых данных, то из сущности “форма” выходит стрелка procedure call, что означает, что клиент посылает данное сообщение объекту-приемнику “проверка данных” - и, чтобы продолжить свою работу, ожидает,

пока вся дальнейшая вложенная последовательность сообщений не будет обработана приемником.

Проверка данных

Сущность “проверка данных” начинается с фокуса управления. Данная сущность появляется в том случае, если клиент не был уверен в правильности введённых данных, тогда фокус управления передаётся из “формы” в “проверку данных” и данная сущность создаётся.

Происходит проверка данных, после чего из левого нижнего угла фокуса управления выходит стрелка типа return, которая сообщает сущности “форма” о том, что проверка данных завершена и можно продолжать выполнение следующих операций.

Фокус управления больше никогда не будет отдан сущности “проверка данных”, поэтому данная сущность уничтожается после передачи фокуса управления другой сущности.

Форма

Из “формы” было отправлено сообщение с помощью стрелки procedure call, что означает, что фокус управления формы ожидает, когда данная операция выполнится, чтобы можно было выполнять следующие операции. Теперь это возможно - так как от сущности “проверка данных” пришло сообщение типа return о том, что проверка данных закончена. После этого происходит нажатие кнопки “оплатить” в этой же самой сущности “форма”. Отдельно стоит заметить, что это происходит и в том случае, если клиент уверен в правильности данных, и если не уверен. Далее происходит отправка сообщения “платёж” в сущность “банк”, используя стрелку procedure call, что означает, что из сущности “форма” посылается данное сообщение объекту-приемнику “банк” - и, чтобы продолжить свою работу, “форма” ожидает, пока вся дальнейшая вложенная последовательность сообщений не будет обработана приемником.

Банк

Сущность ”банк” начинается с линии жизни. Когда фокус управления передаётся из сущности “форма” в “банк”, то появляется фокус управления.

Банк принимает сообщение “платёж” и обрабатывает его. Для его обработки необходимо выполнить транзакцию - это отдельная сущность в данной диаграмме. Для этого из “банка” в “транзакцию” посылается сообщение о выполнении транзакции со стрелкой procedure call, что означает, что “банк” посылает данное сообщение объекту-приемнику “транзакция” - и, чтобы продолжить свою работу, ожидает, пока вся дальнейшая вложенная последовательность сообщений не будет обработана приемником.

Транзакция

Сущность ”транзакция” начинается с линии жизни. Когда фокус управления передаётся из сущности “банк” в “транзакцию”, то появляется фокус управления.
Сущность “транзакция” совершает непосредственно транзакцию. После завершения данной операции из неё выходит стрелка типа return, передающая сообщение о том, что транзакция успешно завершена.
Фокус управления больше никогда не будет отдан сущности “транзакция”, поэтому данная сущность уничтожается после передачи фокуса управления другой сущности.

Ремарка:
Как описано выше, некоторые сущности отправляли сообщения с помощью стрелки procedure call, использование которой говорит о том, что сущности ждут, когда вложенные операции выполнятся. Ниже будут описаны оставшиеся у сущностей операции, которые они выполняют, когда вложенные действия заканчиваются.

Банк: после того, как дождался ответа от сущности “транзакция”, отправляет сообщение типа return сущности “форма”, говорящее о том, что платёж совершён.
Форма: после того, как дождался ответа от сущности “банк”, отправляет сообщение типа return сущности “клиент”, говорящее о том, что покупка тура совершена.
Клиент: принимает от “формы” сообщение о том, что тур куплен.