**Министерство образования и науки Российской Федерации**

**федеральное государственное бюджетное образовательное учреждение высшего профессионального образования**

**«Российский экономический университет имени Г.В. Плеханова»**

Институт цифровой экономики и информационных технологий

Кафедра информатики

**Практическая работа**

по дисциплине

«Объектно-ориентированные технологии в программной инженерии»

Выполнили

студенты 3 курса

группы 15.27Д-БИ19/22Б

очной формы обучения

высшая школа ВШКМиС

Фамилии студентов Нгуен Као Бач,

Москва, 2025

**Задание 1. Диаграмма состояний «Таймер»**

A diagram of a diagram

AI-generated content may be incorrect.

|  |  |  |
| --- | --- | --- |
| Элемент UML | Пример в диаграмме | Обозначение |
| ● | Начальное состояние | Таймер включается (вход в систему) |
| Прямоугольник с закруглёнными углами | Контроль времени, Установка времени | Отдельные состояния таймера |
| Вложенное состояние | Изменение часов, Изменение минут | Подсостояния в «Установка времени» |
| do / Проверка() | В состоянии «Контроль времени» | Действие, выполняемое в процессе |
| ➝ Стрелки перехода | включение, установка, режим, выключение/сброс | События, вызывающие переходы |

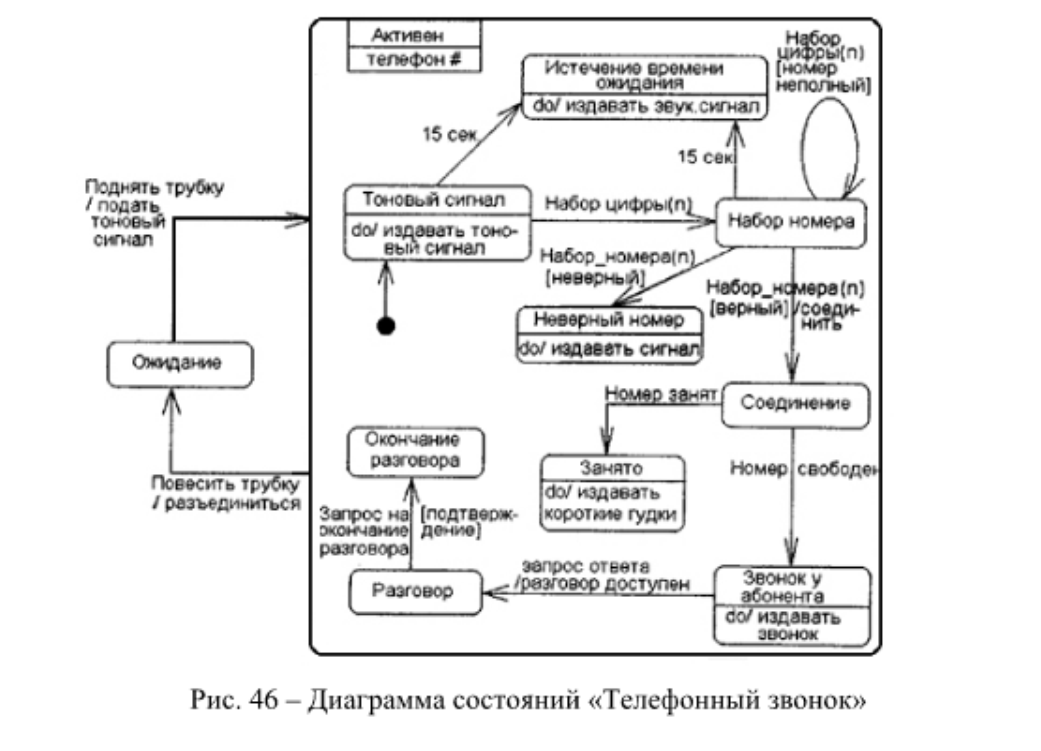
Диаграмма состояний начинается с начального состояния, которое обозначено чёрной точкой. При наступлении события «включение», система переходит в состояние «Контроль времени». В этом состоянии выполняется внутреннее действие do / Проверка(), то есть осуществляется периодическая проверка времени.

Если пользователь выбирает действие «установка», таймер переходит в составное состояние «Установка времени». Это составное состояние включает два вложенных состояния: сначала «Изменение часов», которое является состоянием по умолчанию. При повторном наступлении события «установка» система переходит во вложенное состояние «Изменение минут».

После завершения процесса установки времени, при наступлении события «режим», система возвращается в состояние «Контроль времени».

В любой момент времени, при наступлении события «выключение/сброс», система выходит из текущего процесса и возвращается в начальное состояние, то есть происходит возврат к исходной точке.

**Задание 2. Диаграмма состояний «Телефонный звонок»**



|  |  |  |
| --- | --- | --- |
| Элемент UML | Пример | Назначение |
| ● | Начальное состояние | Старт работы телефона |
| Прямоугольники | Ожидание, Тоновый сигнал и др. | Состояния |
| do / действие | do/издавать сигнал | Внутреннее поведение |
| ➝ переход | События: набор, соединить и др. | Переход между состояниями |
| Ветвление (if/else) | Верный/неверный номер | Условия перехода |

Телефон начинает в состоянии **«Ожидание»**. Если пользователь **поднимает трубку**, телефон переходит в **«Тоновый сигнал»** и воспроизводит тоновый сигнал (do/издавать тоновый сигнал). Затем начинается **набор цифр** – если номер **неверный**, система переходит в **«Неверный номер»** и издаёт сигнал ошибки. Если номер **верный**, телефон переходит в **«Соединение»**.

В зависимости от доступности:

* Если **номер занят**, система воспроизводит короткие гудки (Занято).
* Если **номер свободен**, телефон переходит в **«Звонок у абонента»** (do/издавать звонок).

Если абонент отвечает, начинается **«Разговор»**, который может завершиться по событию **«окончание разговора»**. После завершения возвращается в **«Ожидание»**.

Аналогично, если происходит **входящий звонок**, телефон начинает звонить. При снятии трубки – переход в **«Разговор»**.

**Задание 3. Диаграмма состояний «Игра «Жизнь»»**

A diagram of a diagram

AI-generated content may be incorrect.

На диаграмме представлены два возможных состояния клетки: «Мёртвая» и «Живая».  
Переход между этими состояниями осуществляется в соответствии с правилами игры «Жизнь» Джона Конуэя.

Состояние «Мёртвая» связано с условием, при котором клетка может ожить: если ровно три живые соседки, то клетка переходит в состояние «Живая». В этом случае происходит событие «Оживает» и выполняется поведение entry / Очистить клетку.

Состояние «Живая» сохраняется, если у клетки две или три живые соседки. В этом случае выполняется внутреннее действие do / Обновить цвет.

Если у живой клетки меньше двух или больше трёх живых соседок, она умирает, возвращаясь в состояние «Мёртвая». Это сопровождается событием «Умирает».

Начальная точка (Start) указывает, что по умолчанию клетка стартует в состоянии «Мёртвая». Конечного состояния нет, так как модель предполагает цикличную, бесконечную смену поколений.

**Задание 4**

A diagram of a company

AI-generated content may be incorrect.

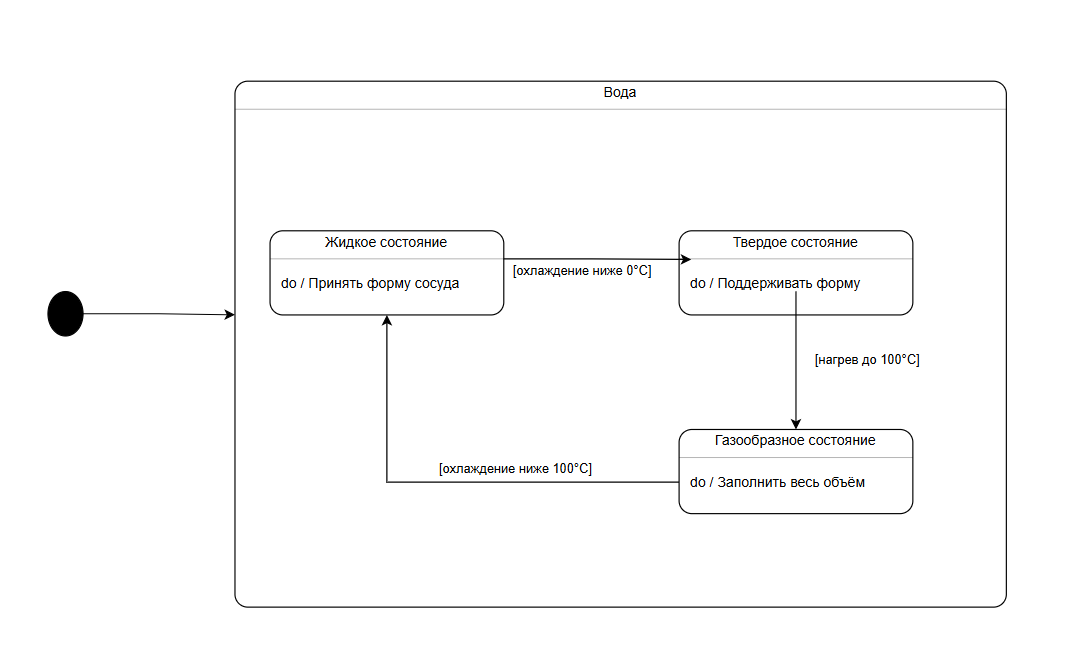
Реактор изначально находится в неактивном состоянии, что обозначено стрелкой от начального узла. При событии «включить реактор» он переходит в активное состояние, где происходит накопление энергии (действие do / Накопление энергии).

Когда достигается критическая масса, реактор переходит в критическое состояние, в котором осуществляется стабильная реакция (do / Стабильная реакция).  
Из критического состояния возможны два сценария:

* Если энергия снижается, реактор переходит в подкритическое состояние (do / Остывание), после чего остывает и выключается, возвращаясь в неактивное состояние.
* Если энергия превышена, реактор переходит в надкритическое состояние (do / Риск аварии). Чтобы избежать аварии, осуществляется аварийное отключение, возвращающее систему в неактивное состояние.

Из неактивного состояния система может быть полностью завершена, переходя в состояние «Остановка» (entry / Завершить работу), за которым следует конечный узел.

**Задание 5**



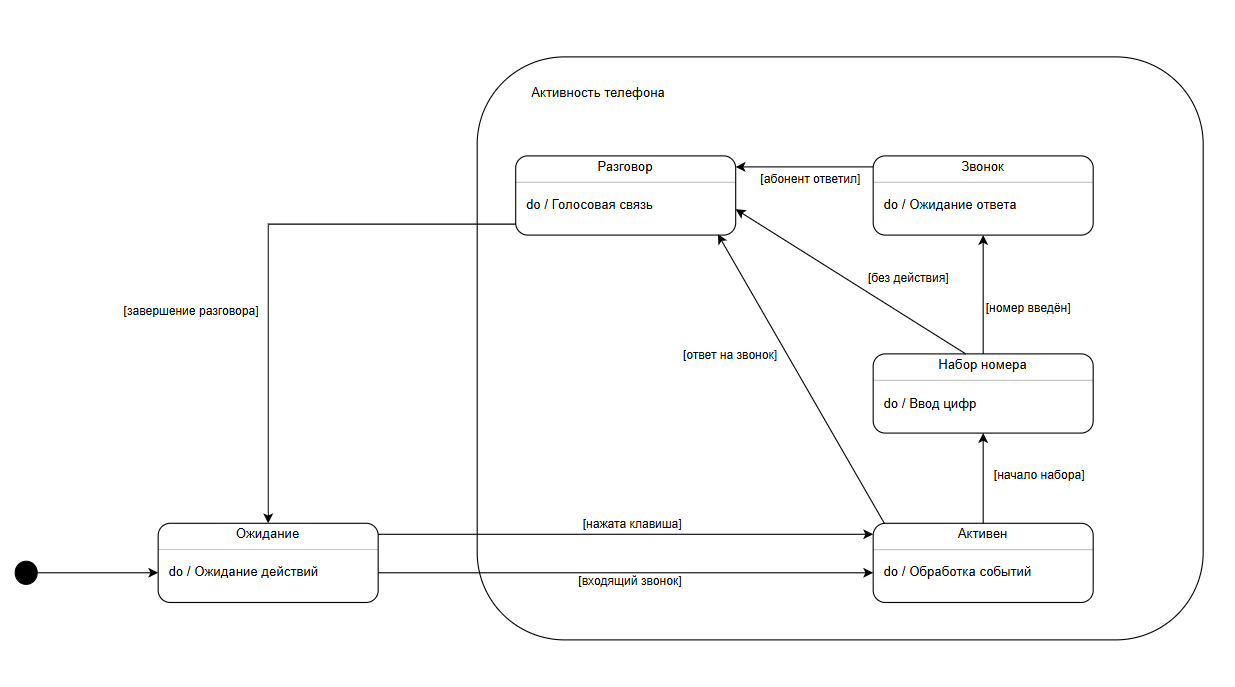
Система начинается с **жидкого состояния**, обозначенного стрелкой от начального узла. В этом состоянии вода принимает форму сосуда (do / Принять форму сосуда).

При **охлаждении ниже 0°C** вода переходит в **твёрдое состояние**, в котором сохраняет свою форму (do / Поддерживать форму).

При **нагреве до 100°C** вода переходит в **газообразное состояние**, где заполняет весь доступный объём (do / Заполнить весь объём).

Если температура понижается **ниже 100°C**, пар конденсируется обратно в **жидкое состояние**.

**Задание 6**



Начальное состояние системы — «Ожидание», где телефон ожидает действий пользователя или входящего звонка. В этом состоянии выполняется поведение do / Ожидание действий. При нажатии клавиши телефон переходит в состояние «Активен», в котором осуществляется do / Обработка событий. Если пользователь начинает набирать номер ([начало набора]), телефон переходит в состояние «Набор номера» (do / Ввод цифр).

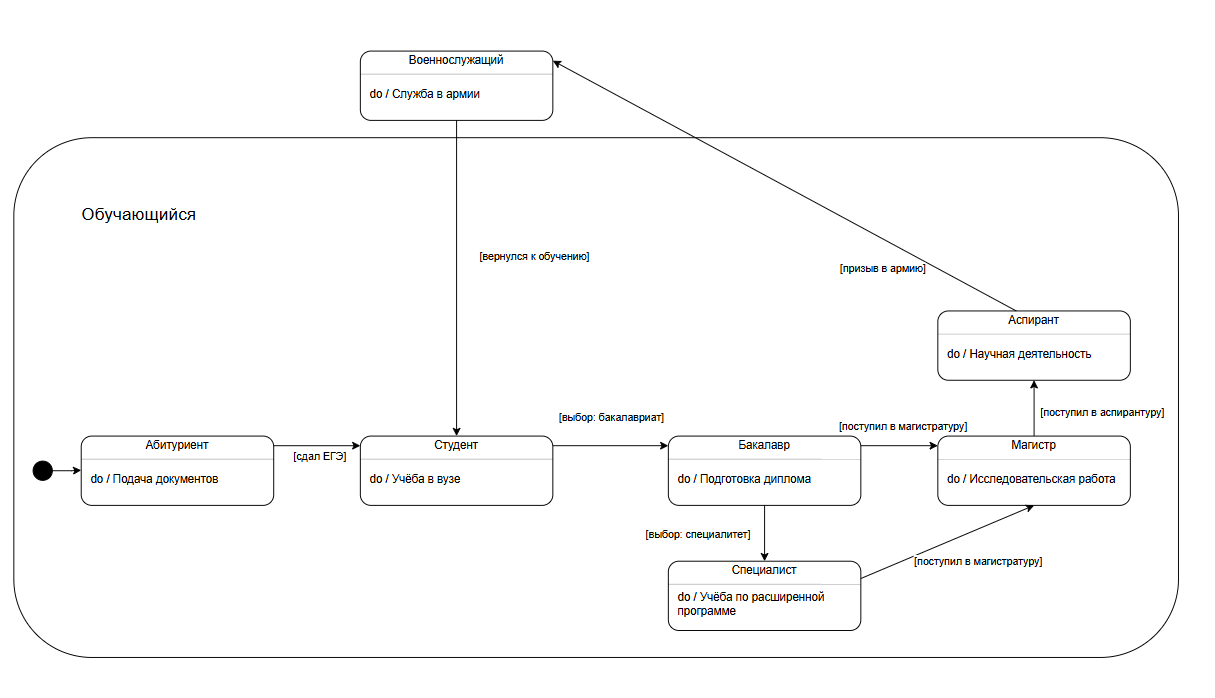
Когда номер введён ([номер введён]), вызывается состояние «Звонок» — ожидание ответа абонента (do / Ожидание ответа). Если абонент ответил ([абонент ответил]), телефон переходит в состояние «Разговор» (do / Голосовая связь). После завершения разговора ([завершение разговора]), телефон возвращается в состояние ожидания.

Система также обрабатывает входящие вызовы: при входящем звонке ([входящий звонок]) телефон переходит в состояние «Активен», и при ответе ([ответ на звонок]) — сразу в «Разговор».

Если после активации не происходит никаких действий ([без действия]), телефон автоматически возвращается в режим ожидания.

Все активные состояния (Активен, Набор номера, Звонок, Разговор) находятся внутри обобщенного состояния «Активность телефона», что подчёркивает логическую связанность этих этапов.

**Задание 7**



На диаграмме представлены возможные состояния человека в системе образования, начиная с момента поступления в университет.

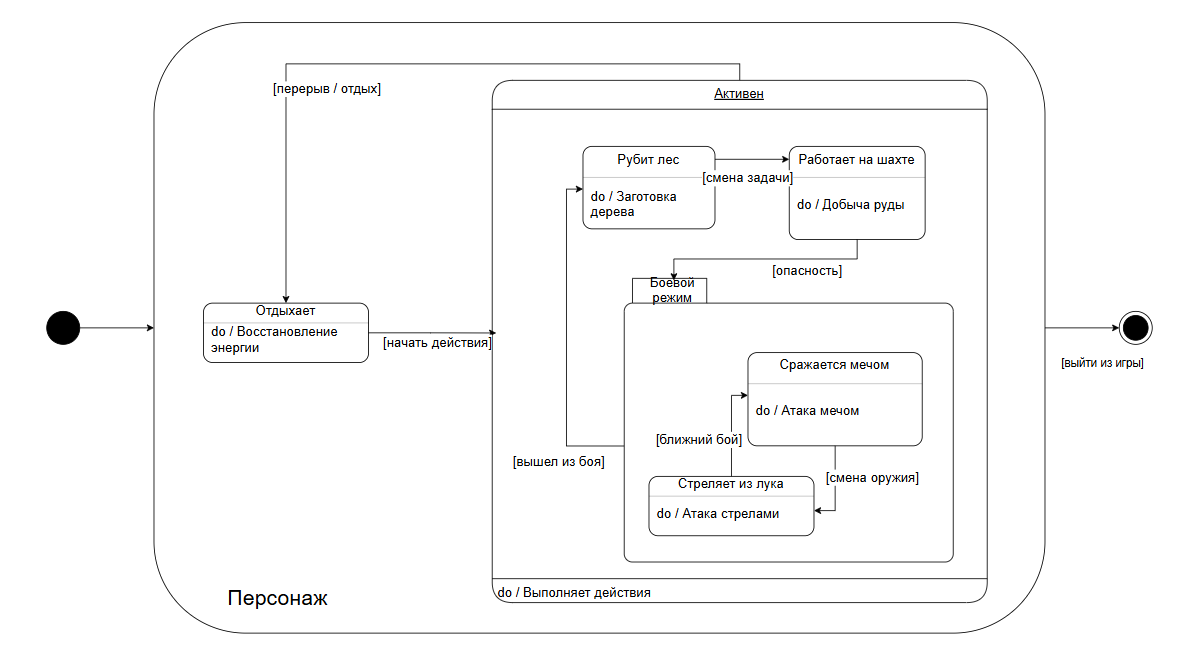
Начальное состояние — «Абитуриент», отражающее этап подачи документов. При успешной сдаче ЕГЭ ([сдал ЕГЭ]) человек становится «Студентом».

Будучи студентом, человек может выбрать траекторию:

* «Бакалавр» ([выбор: бакалавриат]),
* или «Специалист» ([выбор: специалитет]).

После окончания бакалавриата или специалитета возможен переход в магистратуру ([поступил в магистратуру]), а затем — в аспирантуру ([поступил в аспирантуру]). Для мужчин возможен вызов на военную службу после окончания специалитета ([призыв в армию]). По окончании службы человек может вернуться в систему образования ([вернулся к обучению]) на уровень студента.

**Задание 8**



На данной диаграмме представлены возможные состояния персонажа в онлайн-игре. Диаграмма начинается с начального состояния, после чего персонаж переходит в состояние «Отдыхает», в котором он восстанавливает энергию (do / Восстановление энергии).

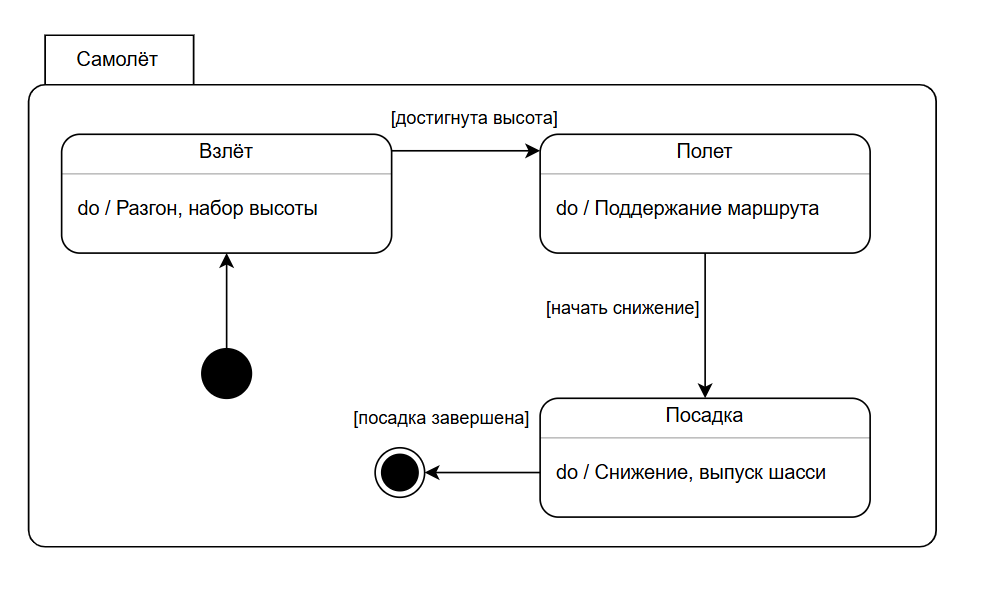
По событию [начать действия], персонаж активизируется и переходит в состояние «Активен», содержащее несколько под состояний:

* Рубит лес – заготовка дерева
* Работает на шахте – добыча руды
* Боевой режим – суперсостояние, включающее:
  + Сражается мечом – ближний бой
  + Стреляет из лука – дальняя атака

Между под-состояниями внутри «Активен» и «Боевого режима» предусмотрены переходы, отражающие смену задач, опасность или смену оружия.

По событию [перерыв / отдых] персонаж возвращается в состояние «Отдыхает».  
Если игрок выходит из игры, осуществляется переход к завершающему состоянию по условию [выйти из игры].

**Задание 9**

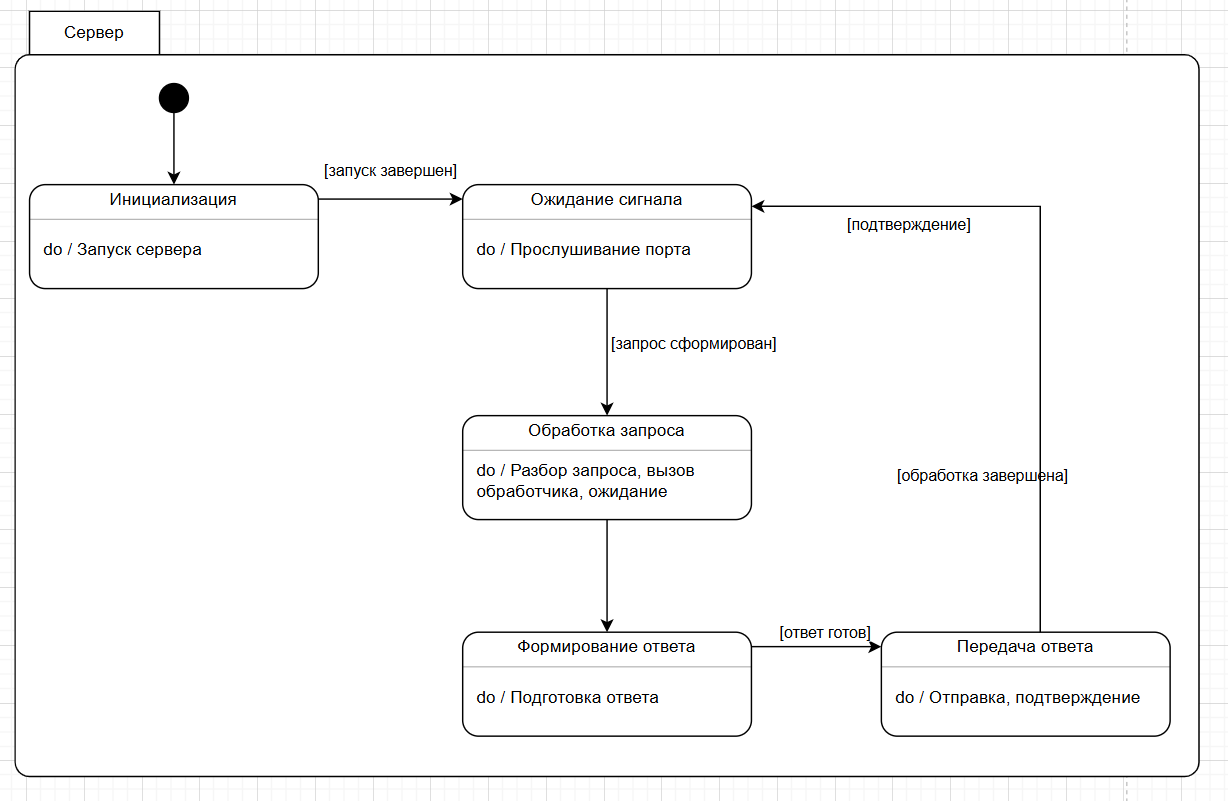
Начальное состояние переходит в состояние **«Взлёт»**, где происходит разгон и набор высоты (do / Разгон, набор высоты).

После достижения необходимой высоты ([достигнута высота]) самолёт переходит в состояние **«Полет»**, в котором осуществляется поддержание маршрута с включённым автопилотом (do / Поддержание маршрута).

Когда начинается снижение ([начать снижение]), самолёт переходит в состояние **«Посадка»**, где он снижает скорость и высоту, выпускает шасси и закрылки (do / Снижение, выпуск шасси).

После завершения посадки ([посадка завершена]), происходит переход в завершающее состояние , завершая весь цикл полёта.

**Задание 10**



Начальное состояние ведёт к состоянию «Инициализация», где происходит запуск сервера (do / Запуск сервера).

После завершения инициализации ([запуск завершен]) сервер переходит в состояние «Ожидание сигнала», в котором он прослушивает порт и ожидает сигналы от клиента (do / Прослушивание порта).

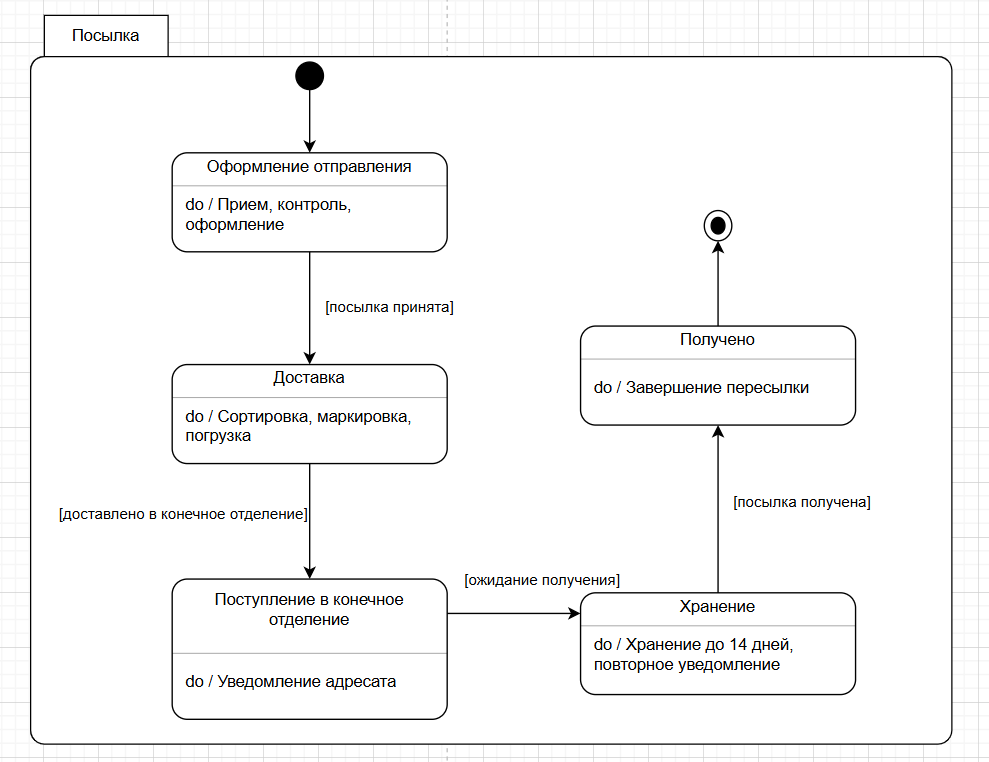
Как только поступает сигнал и формируется запрос ([запрос сформирован]), сервер переходит в состояние «Обработка запроса», где анализирует запрос, вызывает обработчик и ожидает результат (do / Разбор, вызов, ожидание).

Когда обработка завершена ([обработка завершена]), формируется ответ в состоянии «Формирование ответа» (do / Подготовка ответа).

Далее происходит «Передача ответа», в которой осуществляется отправка и подтверждение получения (do / Отправка, подтверждение).

После успешной передачи сервер возвращается в состояние «Ожидание сигнала», готов к обработке следующего запроса.

**Задание 11**



Начальное состояние ведёт к «Оформление отправления», где посылка принимается сотрудником почты, производится контроль веса и адреса, осуществляется оплата (do / Прием, контроль, оформление).

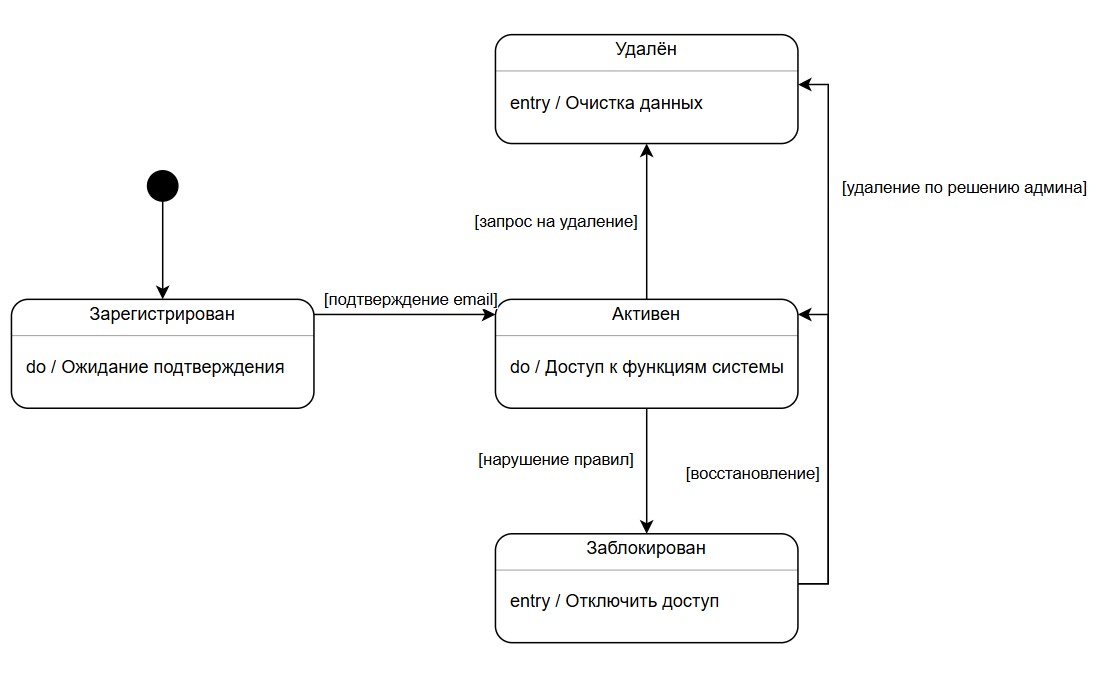
После завершения оформления ([посылка принята]) посылка поступает в состояние «Доставка», где она проходит сортировку, маркировку и погрузку на транспортное средство (do / Сортировка, маркировка, погрузка).

После доставки в пункт назначения ([доставлено в конечное отделение]) посылка переходит в состояние «Поступление в конечное отделение», где формируется уведомление для получателя (do / Уведомление адресата).

Если получатель не забирает посылку сразу, она переходит в состояние «Хранение», где может находиться до двух недель, и при необходимости отправляется повторное уведомление (do / Хранение до 14 дней, повторное уведомление).

После получения ([посылка получена]) посылка переходит в финальное состояние «Получено», после чего диаграмма завершается.

**Задание 12**



Начальное состояние — пользователь находится в состоянии “Зарегистрирован”, где ожидает подтверждения email (do / Ожидание подтверждения). При подтверждении email ([подтверждение email]) пользователь переходит в состояние “Активен”, где имеет полный доступ к функциям системы (do / Доступ к функциям системы). При нарушении правил пользователь блокируется и переходит в состояние “Заблокирован”, в котором отключается доступ (entry / Отключить доступ).

Из состояния блокировки возможен переход обратно в Активен при восстановлении доступа ([восстановление]). При запросе на удаление пользователь переходит в состояние “Удалён”, где происходит очистка данных (entry / Очистка данных). Также возможно удаление по решению администратора напрямую из состояния “Заблокирован”.

**Контрольные вопросы**

1. Перечислите особенности диаграммы состояний.

* Отображает все возможные состояния объекта.
* Показывает переходы между ними.
* Моделирует поведение системы во времени.
* Может содержать вложенные состояния.

2. С какой целью создаются диаграммы состояний?

Для моделирования изменений объекта в ответ на внешние или внутренние события.

3. Перечислите элементы нотации диаграммы состояний.

* Начальное состояние (●)
* Конечное состояние (◎)
* Состояние (прямоугольник с закруглёнными углами)
* Переход (стрелка с условием/действием)

4. Какие стереотипы используются для описания внутреннего поведения объекта?

* entry / действие — при входе в состояние
* exit / действие — при выходе
* do / действие — во время нахождения

5. Как указываются события, приводящие к смене состояния объекта?

Указываются рядом со стрелкой:

[условие] / действие

6. Обязательны ли диаграммы состояний для проектирования?

Нет, только для элементов с нетривиальным поведением.

7. Перечислите условия к автомату, необходимые для соблюдения.

* Переход зависит только от текущего состояния.
* В каждый момент — одно активное состояние.
* Все состояния должны быть связаны.
* Конечное число состояний.
* Нет конфликтов переходов.