

# Конечный автомат

Урок 2  
Диаграммы состояний





# План курса

1

Основы диаграмм состояний

2




Конечный автомат

3

Моделирование  
жизненного цикла объекта  
& введение в UML



## Что будет на уроке сегодня

-  Что такое конечный автомат
-  Вариации конечных автоматов
-  Много примеров для детального понимания процесса! :)





**PlantUML - это свободный и открытый язык и инструмент для создания UML-диаграмм, который позволяет создавать графические представления различных типов моделей, включая диаграммы классов, диаграммы последовательностей, диаграммы состояний и многие другие.**



# Примеры

Автомат, который проверяет, является ли входная строка четным числом.

- ✓ В этой таблице переходов, если автомат находится в стартовом состоянии и получает входной символ "0", он переходит в состояние **"Состояние 1"**.
- ✓ Если автомат находится в состоянии **"Состояние 1"** и получает входной символ "1", он переходит в состояние **"Состояние 2"**.
- ✓ И так далее.

Текущее состояние	Введенный символ	Следующее состояние
Старт	0	Состояние 1
Состояние 1	1	Состояние 2
Состояние 2	0	Состояние 1

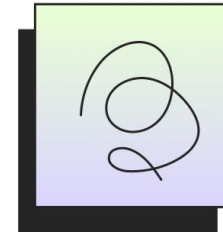


# Конечный автомат и диаграмма состояний



## Конечный автомат

Это математическая модель, которая описывает поведение автомата с помощью состояний, событий и переходов между состояниями.



## Диаграмма состояний

Это графическое представление конечного автомата, которое показывает состояния в виде вершин и переходы между состояниями в виде ребер.



# Элементы конечного автомата

1

## Таблицы переходов

Это таблица, используемая  
в конечных автоматах, для  
описания переходов между  
состояниями автомата  
в зависимости от входных  
символов.





# Элементы конечного автомата

1

## Таблицы переходов

Это таблица, используемая в конечных автоматах, для описания переходов между состояниями автомата в зависимости от входных символов.

2

## Текущее состояние автомата

Это состояние, в котором находится автомат в данный момент.





# Элементы конечного автомата

1

## Таблицы переходов

Это таблица, используемая в конечных автоматах, для описания переходов между состояниями автомата в зависимости от входных символов.

2

## Текущее состояние автомата

Это состояние, в котором находится автомат в данный момент.

3

## Стартовое состояние

Это начальное состояние, в котором находится конечный автомат в начале работы.



# Элементы конечного автомата

1

## Таблицы переходов

Это таблица, используемая в конечных автоматах, для описания переходов между состояниями автомата в зависимости от входных символов.

2

## Текущее состояние автомата

Это состояние, в котором находится автомат в данный момент.

3

## Стартовое состояние

Это начальное состояние, в котором находится конечный автомат в начале работы.

4

## Заключительное состояние

Это состояние, в котором конечный автомат завершает работу после прохождения всех состояний и входных символов



# Таблица переходов состояний

Текущее состояние	Введенный символ	Следующее состояние
Старт	0	Состояние 1
Состояние 1	1	Состояние 2
Состояние 2	0	Состояние 1



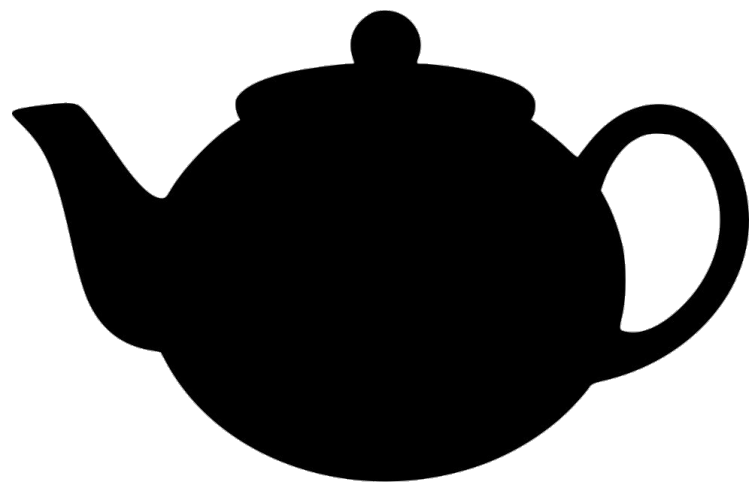
# Пример №1

Рассмотрим бытовой пример с чайником.

**Стартовое состояние:** «в режиме ожидания».

**Событие:** пользователь нажимает кнопку «включения/выключения».

**Событие:** вода нагрелась до заданной температуры.



Текущее состояние	Событие	Следующее состояние
В режиме ожидания	Нажата кнопка «включения/выключения»	В процессе нагрева воды
В процессе нагрева воды	Вода нагрелась до заданной температуры	В режиме ожидания



# Пример №2

Система контроля доступа в здание.

Стартовым состоянием в этом случае будет «Доступ запрещен», а заключительным «Доступ разрешен».

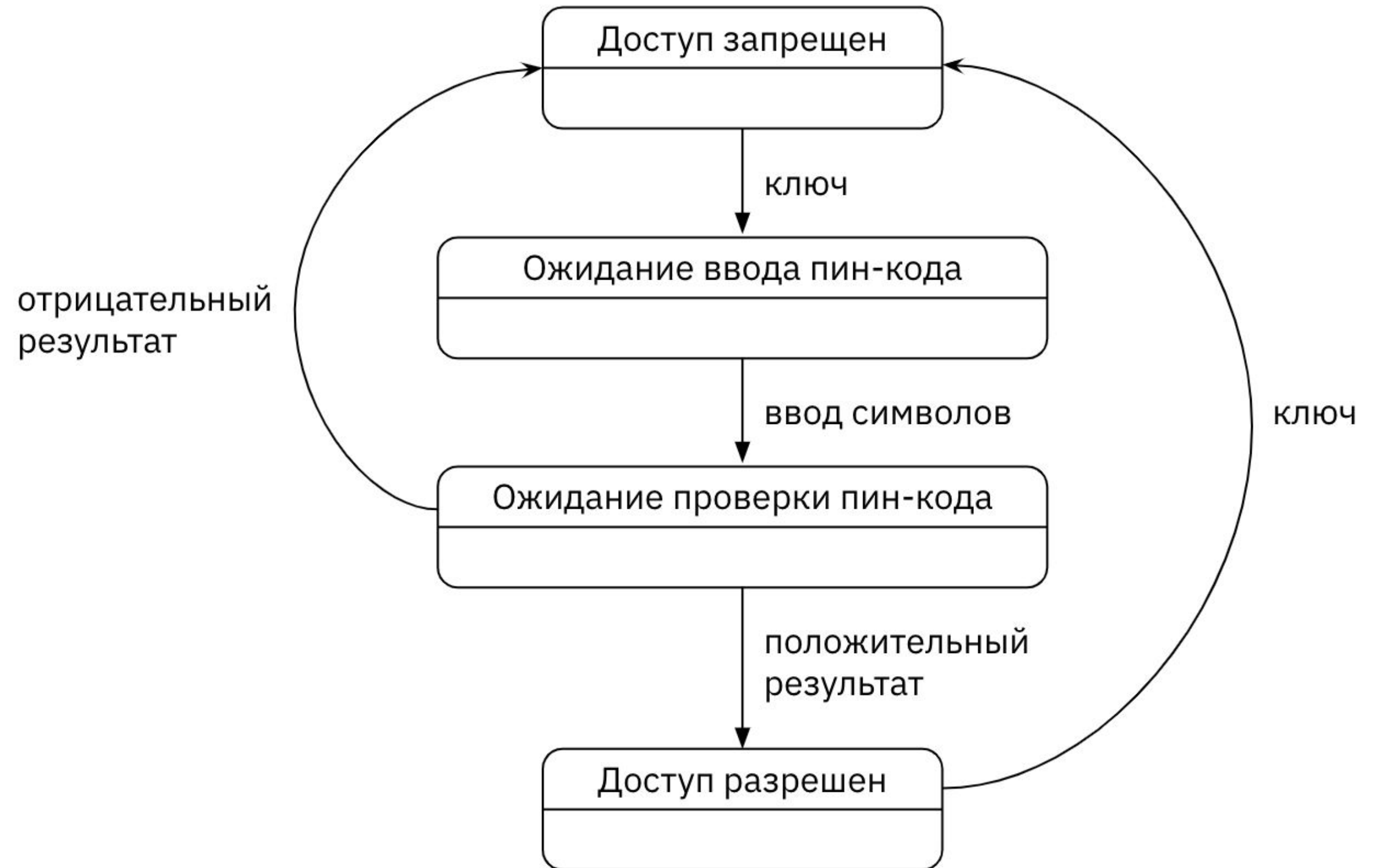
Текущее состояние	Символ	Следующее состояние
Доступ запрещен	Ключ	Ожидание ввода пин-кода
Ожидание ввода пин-кода	Ввод символов	Ожидание проверки пин-кода
Ожидание проверки пин-кода	Проверка пин-кода	Доступ разрешен
Ожидание проверки пин-кода	Отрицательный результат	Доступ запрещен
Доступ разрешен	Ключ	Доступ запрещен



## Пример №2

Система контроля доступа в здание.

Стартовым состоянием в этом случае будет «**Доступ запрещен**», а заключительным «**Доступ разрешен**».





# Примеры

Проверка синтаксиса арифметического выражения

Состояниями автомата будут являться:

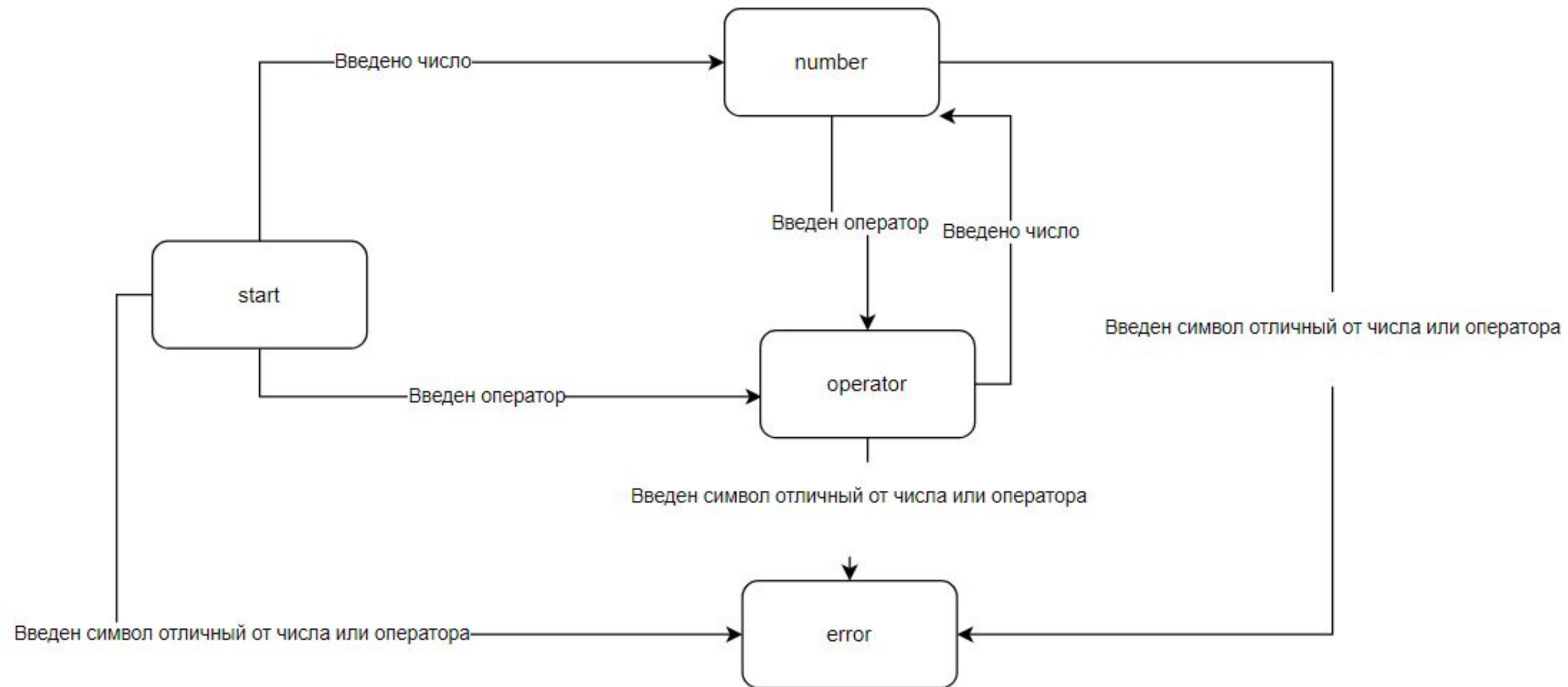
- ✓ «start» — начальное состояние;
- ✓ «number» — если в выражении встречена цифра;
- ✓ «operator» — если в выражении встречен оператор;
- ✓ «error» — в случае ошибки синтаксиса.

Текущее состояние	Символ	Следующее состояние
start	цифра	number
start	оператор	operator
number	цифра	number
number	оператор	operator
operator	цифра	number
operator	оператор	error





# Графическое отображение





# Применение конечных автоматов



## Пример

Программа для проверки домашнего задания  
от студентов Гикбрейнса



## Пример

Программа для проверки домашнего задания от студентов Гикбрейнса

Она может иметь следующие состояния:

- «Не начато» — студент еще не начал выполнять домашнее задание.
- «В процессе» — студент в процессе выполнения домашнего задания.
- «Отправлено» — студент отправил свое домашнее задание на проверку.
- «Оценено» — домашнее задание было проверено и получило оценку.



Текущее состояние	Входные данные	Следующее состояние
Не начато	Начало работы	В процессе
В процессе	Отправка	Отправлено
Отправлено	Оценка	Оценено

## Пример

Программа для проверки домашнего задания от студентов Гикбрейнса

Она может иметь следующие состояния:

- «Не начато» — студент еще не начал выполнять домашнее задание.
- «В процессе» — студент в процессе выполнения домашнего задания.
- «Отправлено» — студент отправил свое домашнее задание на проверку.
- «Оценено» — домашнее задание было проверено и получило оценку.



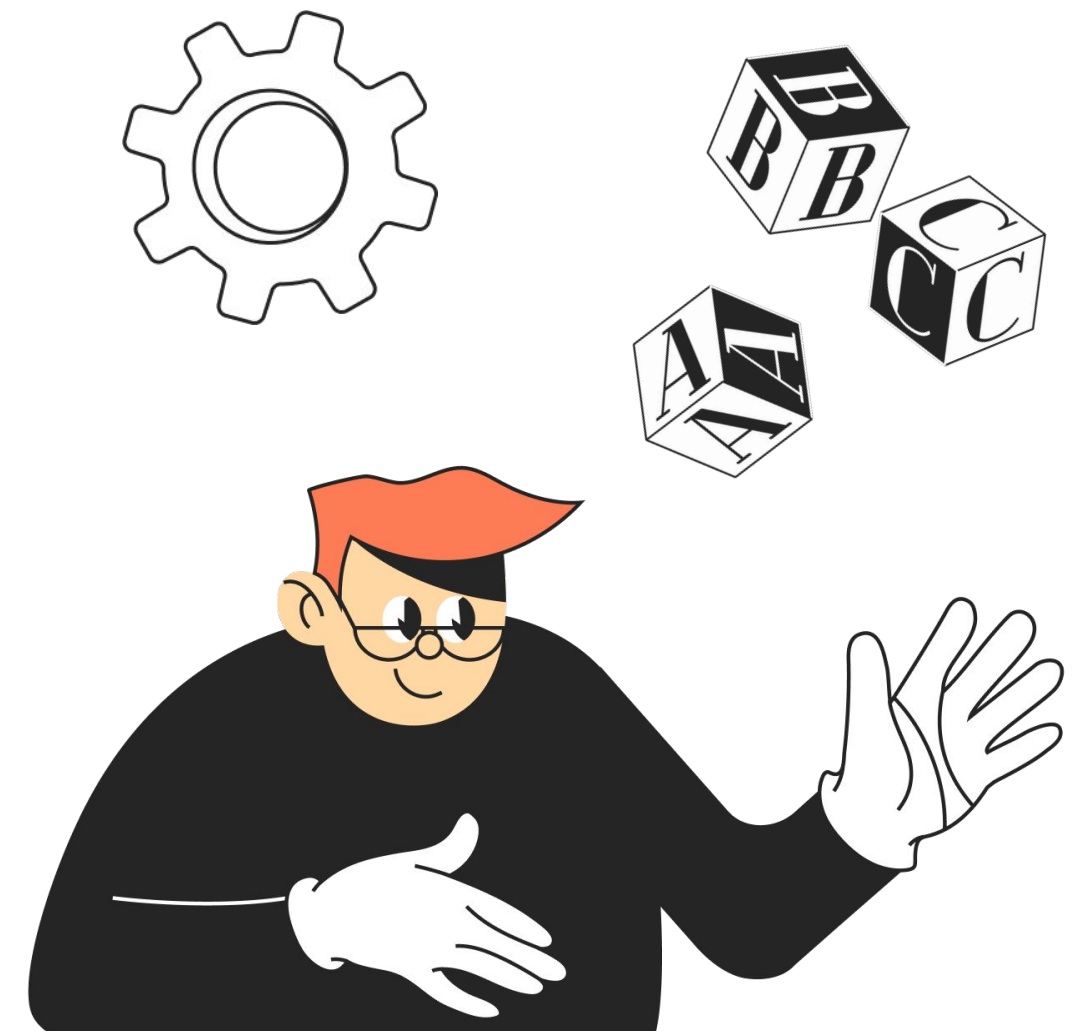
## Немного практики

Рассмотрим еще один пример, который подчеркивает важность этих инструментов, — **воронку продаж.**



## Продукт и проджект менеджеры

- **Управление жизненным циклом продукта:** конечные автоматы могут использоваться для отображения различных состояний продукта и переходов между ними, в том числе в релизах, версиях и исправлениях.
- **Управление проектом:** конечные автоматы могут использоваться для отображения и отслеживания этапов проекта, в том числе планирования, разработки, тестирования и выпуска.
- **Оценка рисков:** конечные автоматы могут использоваться для моделирования рисков и выявления возможных сценариев для их минимизации.

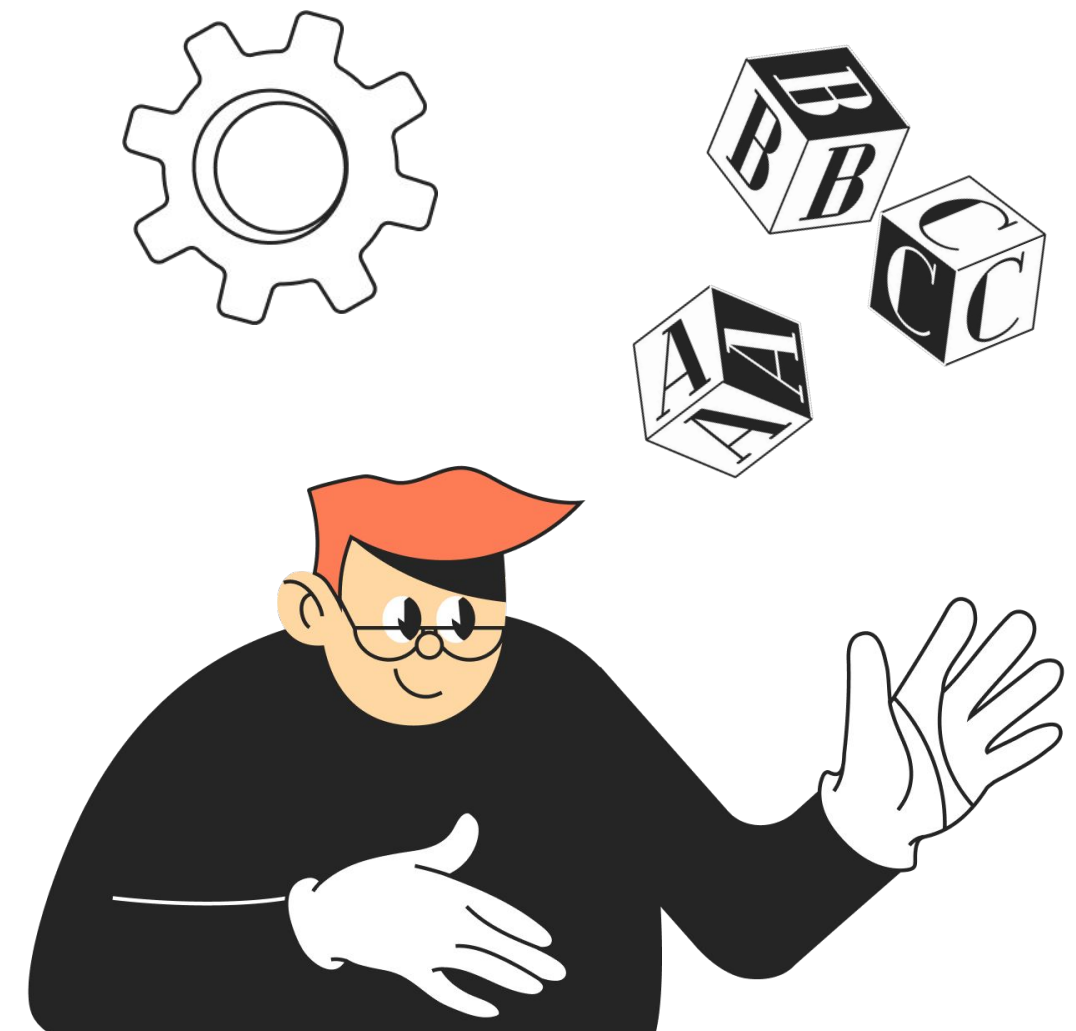






## Продукт и проджект менеджеры

- **Отслеживание сроков исполнения:** конечные автоматы могут использоваться для отслеживания сроков выполнения проекта, чтобы определять, в какой стадии проекта находится, и для предотвращения задержек.
- **Автоматизация рутинных задач:** Конечные автоматы могут использоваться для автоматизации рутинных задач, таких как отслеживание задач или выполнение действий в ответ на события.



## Аналитики

- **Обработка текстовых данных:** анализ текстов, определение тональности, классификация текстов или элементов в них.
- **Обнаружение мошенничества:** анализ транзакций, определение аномалий в поведении пользователей.
- **Маркетинг:** анализ данных о поведении покупателей, идентификация паттернов поведения и сегментация аудитории.
- **Биоинформатика:** анализ генетических данных, поиск зависимостей между генами и заболеваниями.
- **Управление производством:** мониторинг и контроль производственных процессов, определение аномалий.
- **Телекоммуникации:** мониторинг сетевых устройств, определение аномалий или ошибок.





## Разработчики

- **Реализация состояний:** программисты могут использовать конечные автоматы для реализации систем, основанных на состояниях, таких как игры, медиаплееры или устройства умного дома.
- **Улучшение кода:** использование конечных автоматов помогает программистам улучшить читаемость и управляемость кода, устраняя ненужную сложность и дублирование кода.
- **Реализация протоколов:** программисты могут использовать конечные автоматы для реализации протоколов связи, таких как протоколы передачи данных по сети.
- **Автоматизация процессов:** программисты могут использовать конечные автоматы для автоматизации рутинных задач, таких как обработка данных.





## Тестировщики

- Проектирование тестовых случаев для проверки правильной работы состояний в программных приложениях.
- Моделирование ожидаемого поведения программных систем и использование этой информации для написания более эффективных тестов.
- Тестирование производительности и масштабируемости программных систем, использующих состояния.
- Отладка проблем, связанных с переходами состояний в программных системах.





# Типы конечных автоматов



# Типы конечных автоматов

1

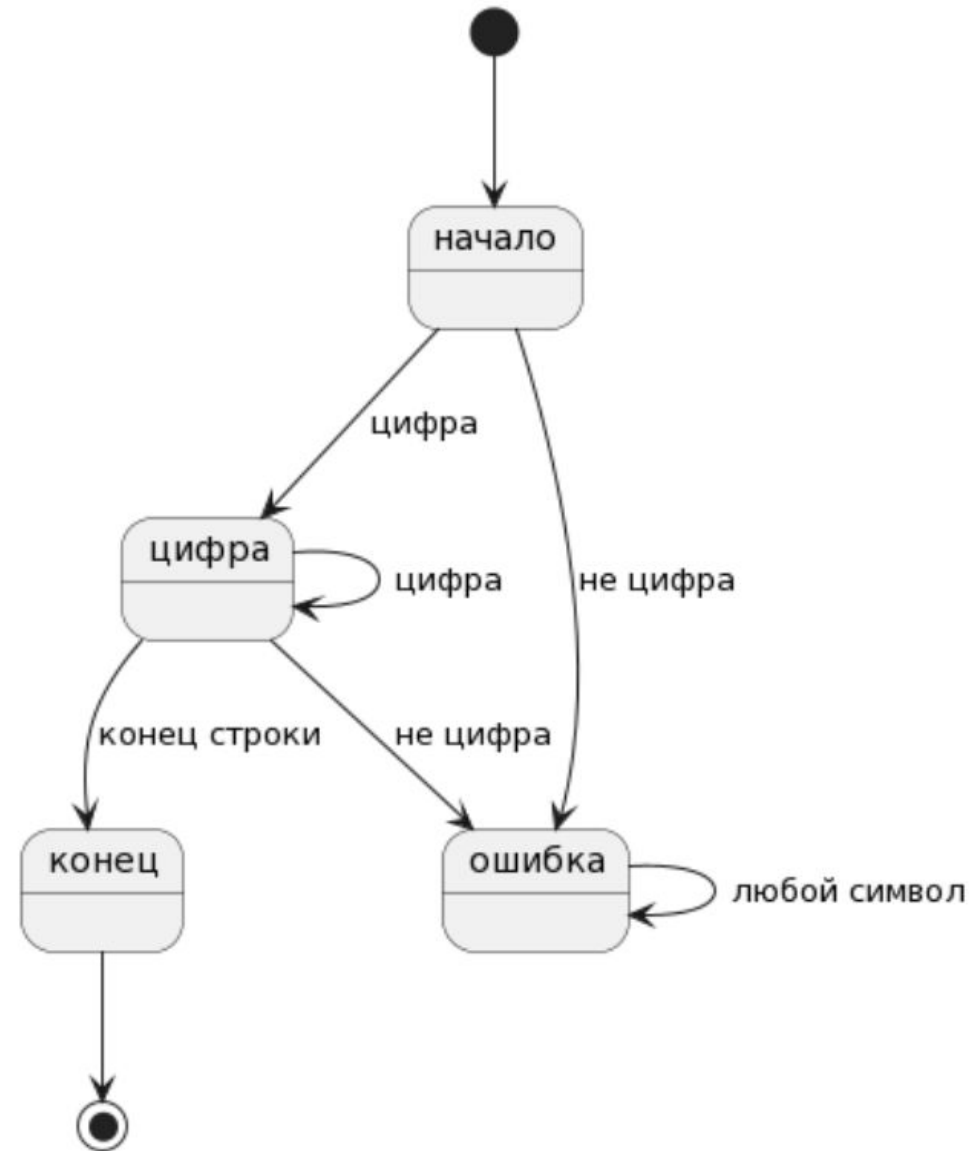
## **Детерминированные конечные автоматы и недетерминированные конечные автоматы**

Конечный автомат, который имеет единственный переход из каждого состояния при данном входном символе. Модель конечного автомата, в которой возможен переход из одного состояния в несколько других состояний одновременно при получении одного и того же символа входной строки.





Детерминированный конечный автомат для определения целого числа



## Пример

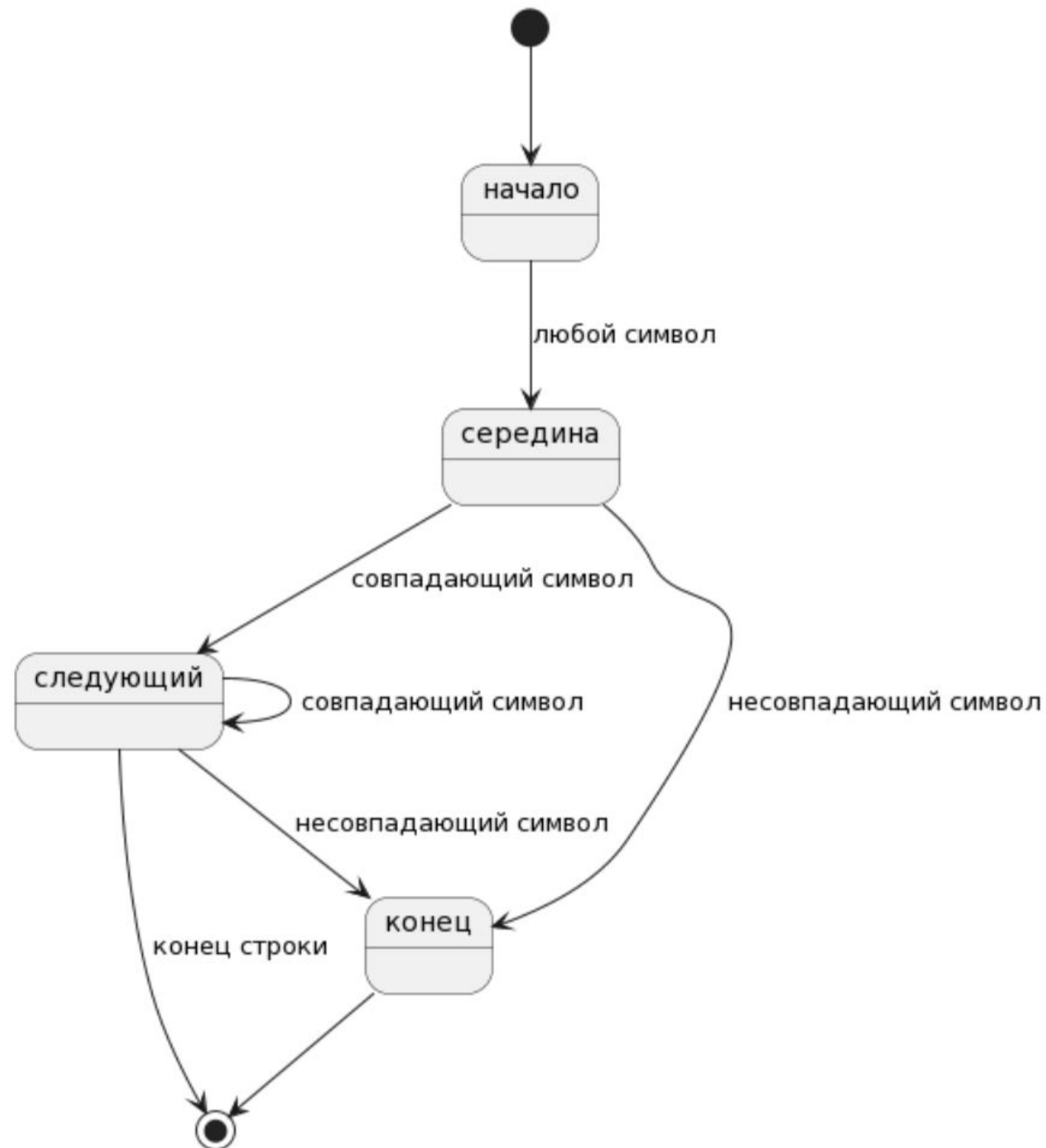
Автомат, который определяет, является ли входная строка целым числом.

- ✓ Состояниями автомата могут быть «начало», «цифра» и «ошибка».
- ✓ Если символ является цифрой, то автомат переходит в состояние «цифра».
- ✓ Если символ не является цифрой, то автомат переходит в состояние «ошибка», далее мы завершаем выполнение нашего автомата.
- ✓ Если входная строка заканчивается в состоянии «цифра», то выражение является целым числом, в противном случае — нет.





Недетерминированный конечный автомат для палиндрома



## Пример

Автомат, проверяющий, является ли входная строка палиндромом.

- ✓ В качестве состояний могут быть «начало строки», «середина строки» и «конец строки».
- ✓ На каждом шаге автомат сравнивает символ с начала и конца строки.
- ✓ Если символы совпадают, то автомат переходит в следующее состояние, если нет, то автомат заканчивает работу и выдает «нет».
- ✓ Можно представить таблицу для начала из середины строки, которая показывает, какие переходы возможны на каждом шаге



# Типы конечных автоматов

1

## **Детерминированные конечные автоматы и недетерминированные конечные автоматы**

Конечный автомат, который имеет единственный переход из каждого состояния при данном входном символе. Модель конечного автомата, в которой возможен переход из одного состояния в несколько других состояний одновременно при получении одного и того же символа входной строки.

2

## **Конечные автоматы с памятью**

Расширенный вариант конечного автомата, который имеет дополнительный стек, который может быть использован для хранения информации. Это позволяет выполнять более сложные операции парсинга, такие как разбор выражений или вложенных скобок.



Конечные автоматы с памятью для проверки суммы чека в магазине



Пример

Пример: конечный автомат для проверки суммы чека в магазине.

- ✓ Стартовое состояние: Проверка пуста.
- ✓ Текущее состояние: Добавление товара в чек.
- ✓ Заключительное состояние: Проверка завершена, выдача результата (соответствует сумма или нет).



# Типы конечных автоматов

1

## Детерминированные конечные автоматы и недетерминированные конечные автоматы

Конечный автомат, который имеет единственный переход из каждого состояния при данном входном символе. Модель конечного автомата, в которой возможен переход из одного состояния в несколько других состояний одновременно при получении одного и того же символа входной строки.

2

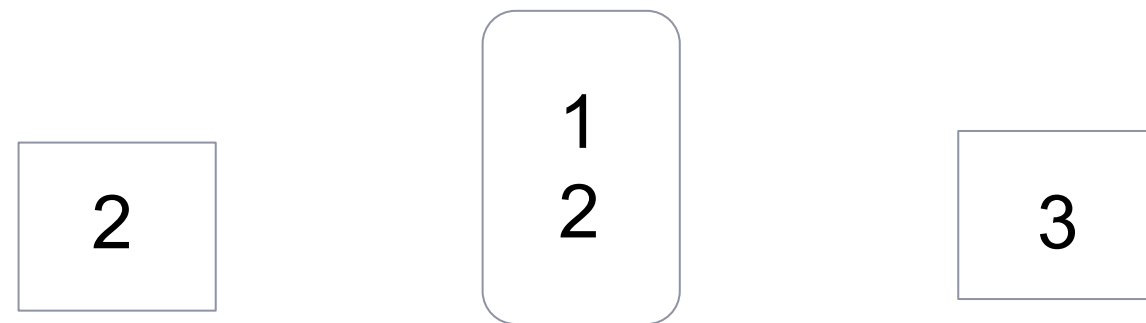
## Конечные автоматы с памятью

Расширенный вариант конечного автомата, который имеет дополнительный стек, который может быть использован для хранения информации. Это позволяет выполнять более сложные операции парсинга, такие как разбор выражений или вложенных скобок.

3

## Машина Тьюринга

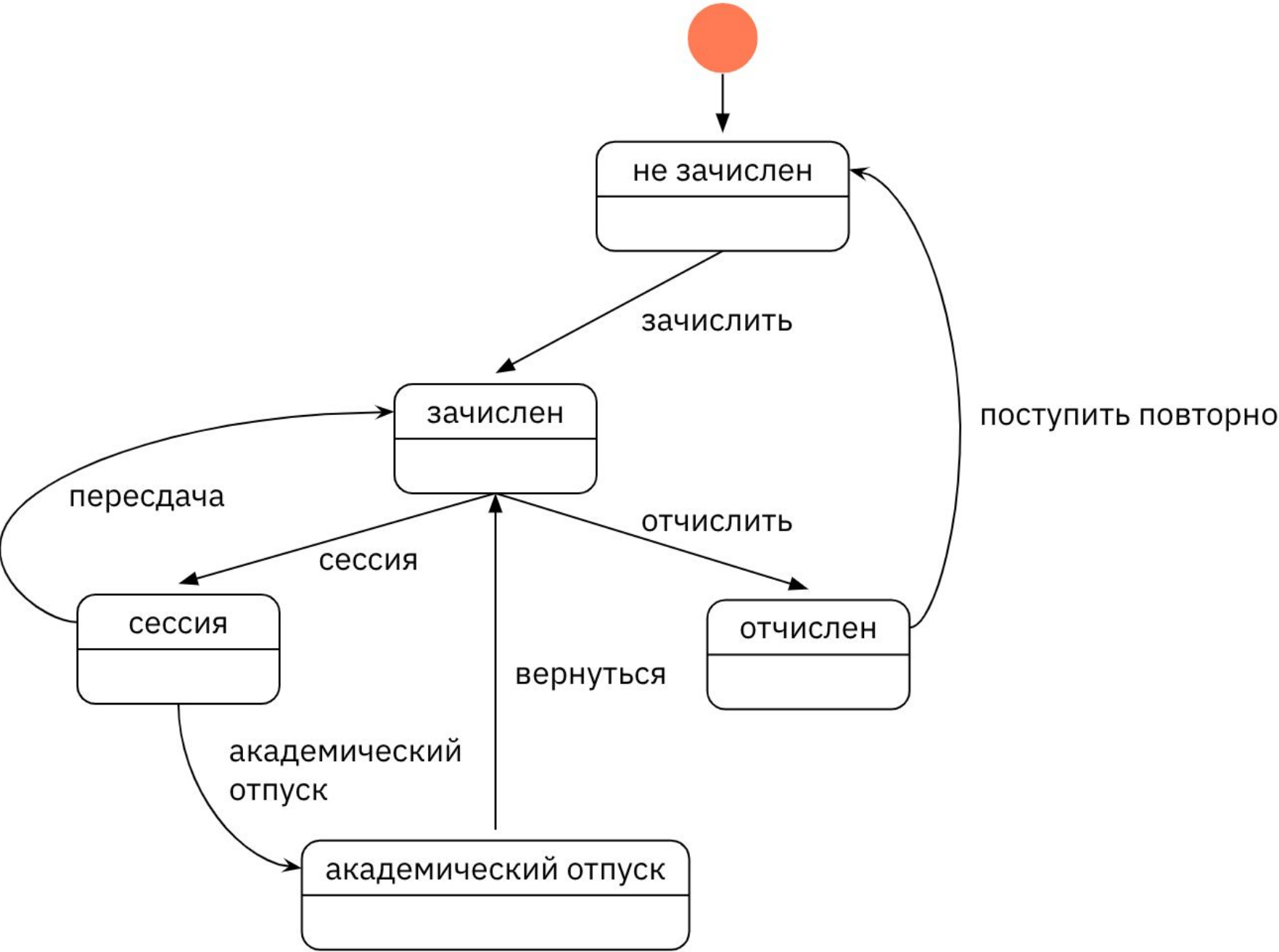
Абстрактный компьютер, который может выполнять вычисления, используя два понятия: последовательность команд и память с перезаписываемым содержимым (обычно реализуемым в виде стека).



## Машина Тьюринга

Конечный автомат

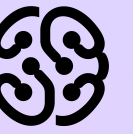
1. Прочитать первое число.
2. Поместить его в стек.
3. Прочитать второе число.
4. Поместить его в стек.
5. Извлечь верхний элемент из стека и сложить его со следующим верхним элементом.
6. Результат сложения поместить в стек.
7. Выдать результат из стека.



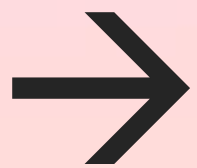
# Конечный автомат для описания статуса обучения



Текущее состояние	Событие	Следующее состояние
Не зачислен	Зачислить	Зачислен
Зачислен	Сессия	Сессия
Сессия	Пересдача	Зачислен
Сессия	Академический отпуск	Академический отпуск
Зачислен	Отчислить	Отчислен
Отчислен	Поступить повторно	Не зачислен
Академический отпуск	Вернуться	Зачислен



Промежуточный  
ИТОГ







# Моделирование жизненного цикла объекта



# Жизненный цикл объекта

## 1. Состояния



Первый шаг в моделировании жизненного цикла объекта — определить его состояния.



Например, для объекта «автомобиль» состояния могут быть «в движении», «стоит на месте», «в ремонте» и т.д. Состояния могут быть взаимоисключающими, или же объект может находиться в нескольких состояниях одновременно.



# Жизненный цикл объекта

1. Состояния объекта

2. **События**



События представляют собой действия, которые могут произойти с объектом.



Например, для объекта «автомобиль» события могут быть «завести двигатель», «остановиться на светофоре», «выйти из строя двигатель» и т.д. События могут приводить к изменению состояния объекта.



# Жизненный цикл объекта

1. Состояния объекта
2. События
- 3. Переходы**



Переходы связывают состояния объекта и события.



Например, для объекта «автомобиль» переход может быть «из состояния 'в движении' в состояние 'стоит на месте' при событии 'остановиться на светофоре'». Переходы также могут быть условными, когда переход происходит только при определенных условиях.



# Жизненный цикл объекта

1. Состояния объекта
2. События
3. Переходы
4. **Действия**



Действия представляют собой действия, которые выполняются при переходе из одного состояния в другое.



Например, для объекта «автомобиль» действие может быть «включить сигнал поворота при переходе из состояния 'в движении' в состояние 'поворот налево'». Действия могут быть как внутренними для объекта, так и вызывать внешние события или функции.



## Жизненный цикл объекта

1. Состояния объекта
2. События
3. Переходы
4. Действия
5. **Конечное состояние**



Когда объект перестает существовать, он достигает конечного состояния.



Например, для объекта «автомобиль» конечное состояние может быть «утилизация». В конечном состоянии объект может быть удален из памяти или переведен в режим ожидания.



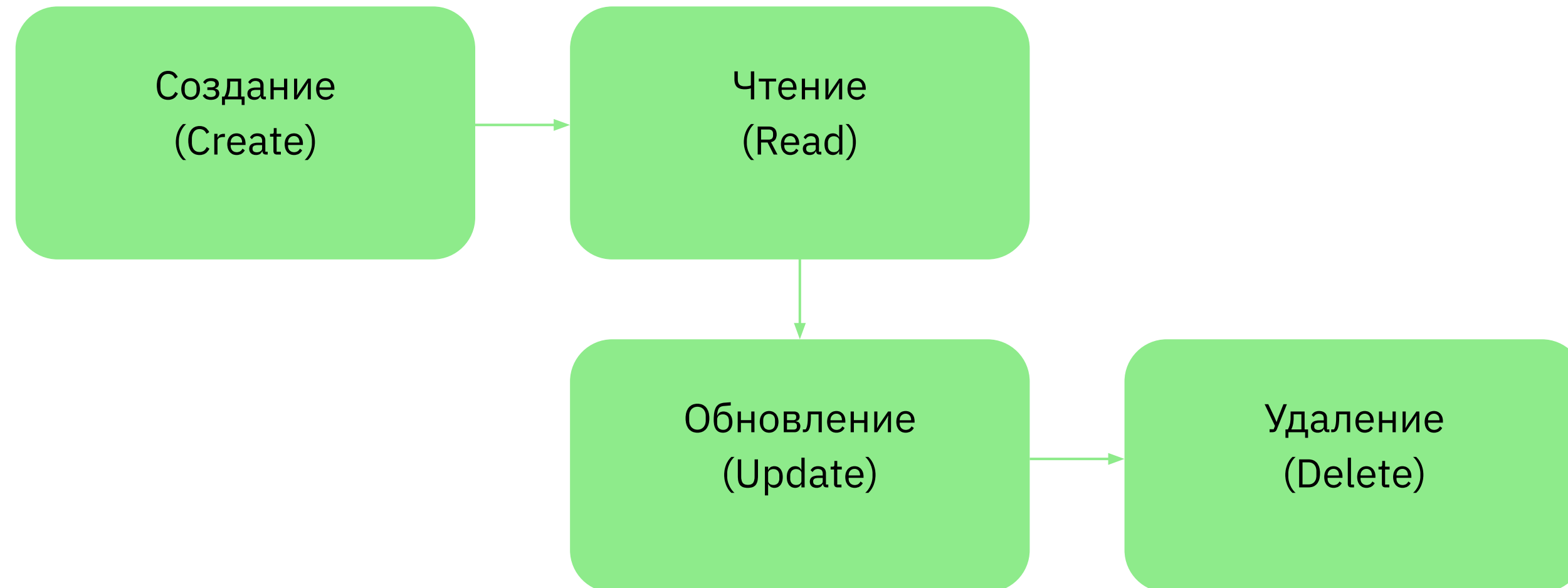
# Моделирование жизненного цикла объекта



- 1. Состояния** **объекта**  
Первый шаг в моделировании жизненного цикла объекта - определить его состояния. Например, для объекта "автомобиль" состояния могут быть "в движении", "стоит на месте", "в ремонте" и т.д. Состояния могут быть взаимоисключающими, или же объект может находиться в нескольких состояниях одновременно.
- 2. События**  
События представляют собой действия, которые могут произойти с объектом. Например, для объекта "автомобиль" события могут быть "завести двигатель", "остановиться на светофоре", "выйти из строя двигатель" и т.д. События могут приводить к изменению состояния объекта.
- 3. Переходы**  
Переходы связывают состояния объекта и события. Например, для объекта "автомобиль" переход может быть "из состояния 'в движении' в состояние 'стоит на месте' при событии 'остановиться на светофоре'". Переходы также могут быть условными, когда переход происходит только при определенных условиях.
- 4. Действия**  
Действия представляют собой действия, которые выполняются при переходе из одного состояния в другое. Например, для объекта "автомобиль" действие может быть "включить сигнал поворота при переходе из состояния 'в движении' в состояние 'поворот налево'". Действия могут быть как внутренними для объекта, так и вызывать внешние события или функции.
- 5. Конечное** **состояние**  
Когда объект перестает существовать, он достигает конечного состояния. Например, для объекта "автомобиль" конечное состояние может быть "утилизация". В конечном состоянии объект может быть удален из памяти или переведен в режим ожидания.



# Модель CRUD







## Усложнение конечных автоматов





## Домашнее задание

- ✓ Постройте конечный автомат по работе с состояниями врача в программе для онлайн-записи из бизнес-кейса.

### Кейс 1: программа для записи к врачу

Программное обеспечение для онлайн-записи к врачу позволяет пациентам записываться на прием через интернет без необходимости обращаться в медицинское учреждение лично.

Функционал, который может включать программа:

- просмотр доступных врачей и их расписания,
- выбор времени и даты приема,
- подтверждение записи через электронную почту или SMS,
- отмена или изменение записи,
- напоминание о записи,
- предоставление доступа с информацией о медицинской карте пациента и истории записей.

Вы можете дополнять этот кейс недостающей информацией.



## Итого

В ходе данной лекции:

- ✓ Разобрались с конечными автоматами
- ✓ Научились строить конечные автоматы
- ✓ Разобрались с усложнением конечных автоматов





Спасибо за внимание