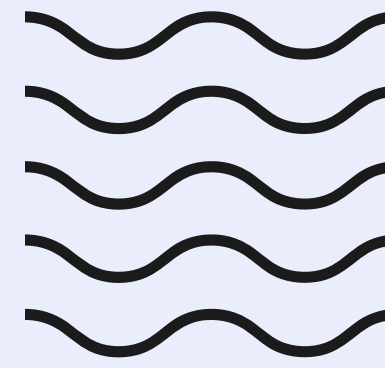


Queue

Deque



ПОДГОТОВИЛ: АЛИШЕР ХАМИДОВ



Основные моменты:

План лекции

1. FIFO (first in - first out)
2. LIFO (last in - first out)
3. Интерфейс Queue<E>
4. Интерфейс Deque<E>
5. Класс ArrayDeque



LIFO

Последний вошел - первый вышел

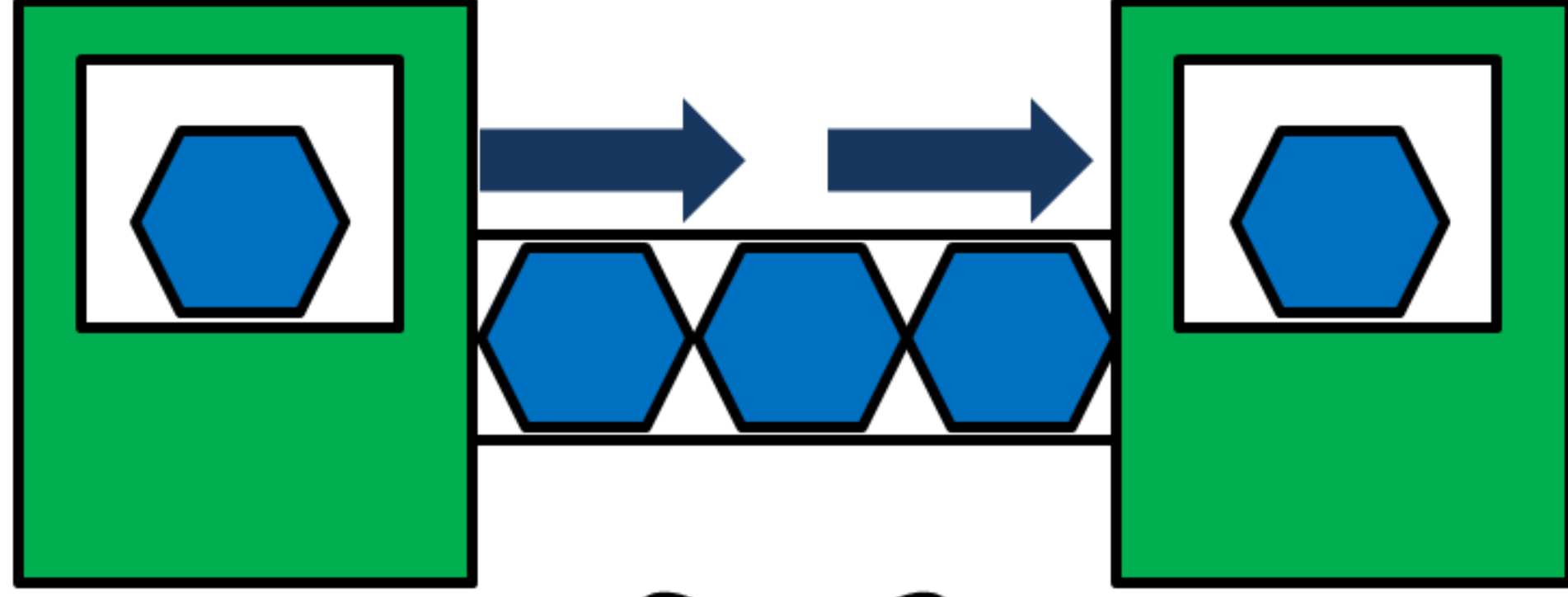


FIFO

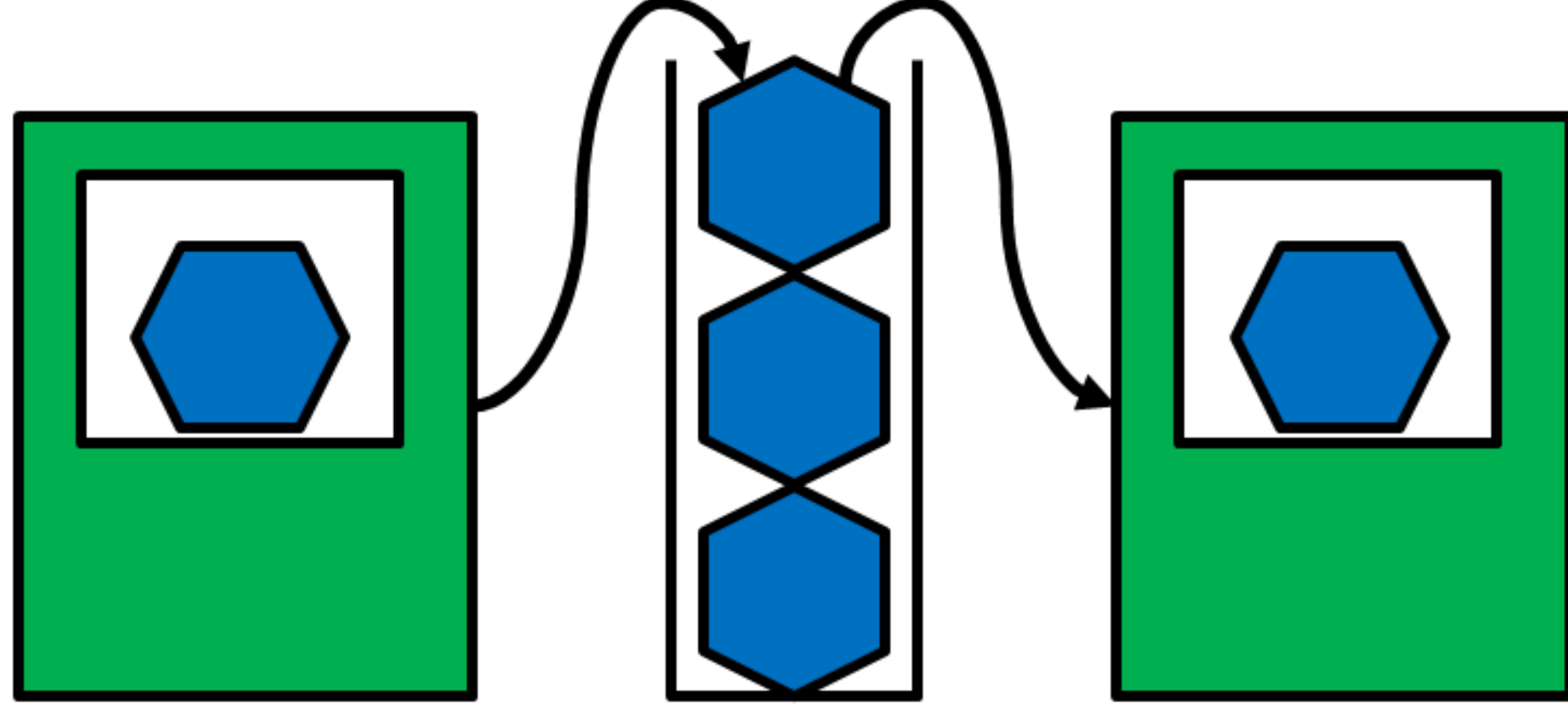
Первый вошел - первый вышел



FIFO



LIFO

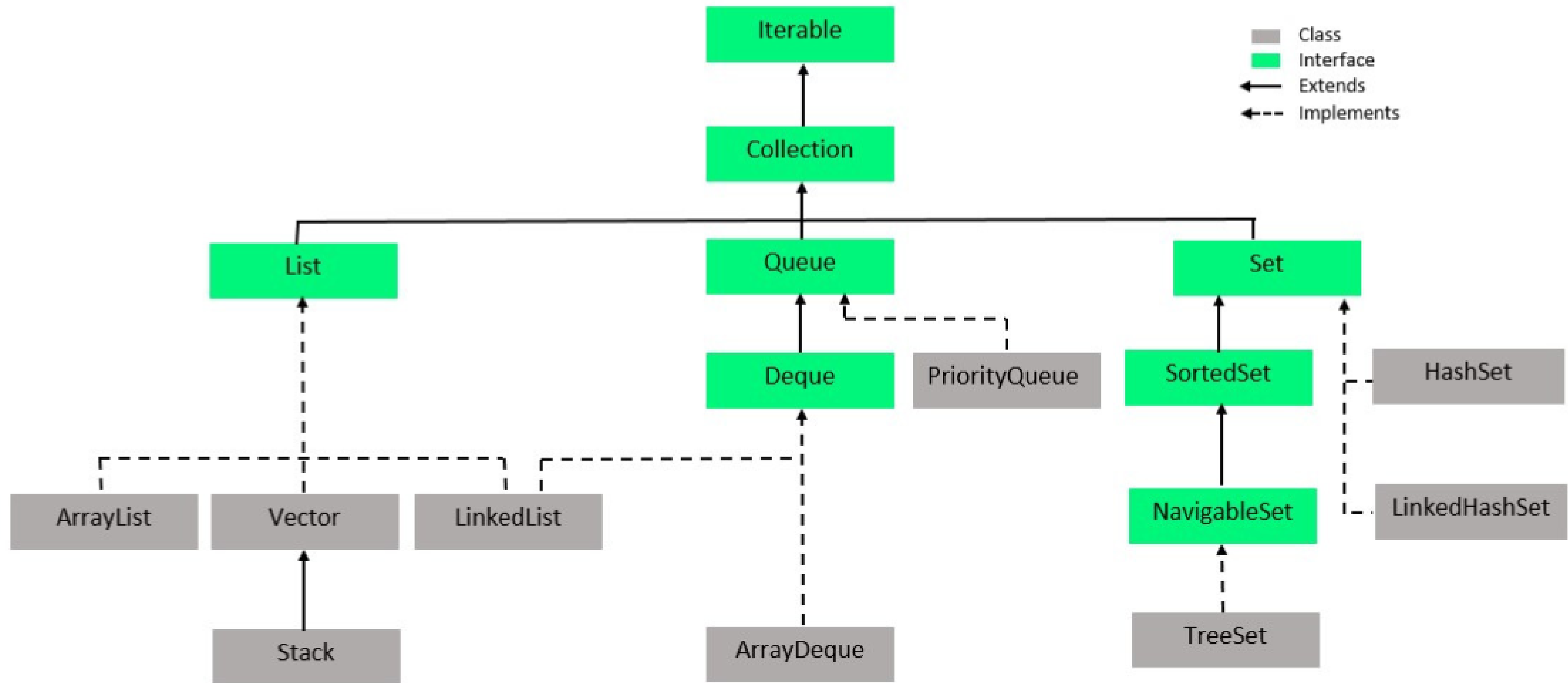


Два интерфейса для очередей



Queue

Deque



Обобщенный интерфейс Queue<E> (однонаправленная очередь)

- **E element()**: возвращает, но не удаляет, элемент из начала очереди. Если очередь пуста, генерирует исключение NoSuchElementException
- **boolean offer(E obj)**: добавляет элемент obj в конец очереди. Если элемент удачно добавлен, возвращает true, иначе - false
- **E peek()**: возвращает без удаления элемент из начала очереди. Если очередь пуста, возвращает значение null
- **E poll()**: возвращает с удалением элемент из начала очереди. Если очередь пуста, возвращает значение null
- **E remove()**: возвращает с удалением элемент из начала очереди. Если очередь пуста, генерирует исключение NoSuchElementException

FIFO

Обобщенный интерфейс Deque<E> (двунаправленная очередь)

- void **addFirst(E obj)**: добавляет элемент в начало очереди
- void **addLast(E obj)**: добавляет элемент obj в конец очереди
- E **getFirst()**: возвращает без удаления элемент из головы очереди. Если очередь пуста, генерирует исключение NoSuchElementException
- E **getLast()**: возвращает без удаления последний элемент очереди. Если очередь пуста, генерирует исключение NoSuchElementException
- boolean **offerFirst(E obj)**: добавляет элемент obj в самое начало очереди. Если элемент удачно добавлен, возвращает true, иначе - false
- boolean **offerLast(E obj)**: добавляет элемент obj в конец очереди. Если элемент удачно добавлен, возвращает true, иначе - false

I

FIFO & LIFO

Обобщенный интерфейс Deque<E> (двунаправленная очередь)

- E **peekFirst()**: возвращает без удаления элемент из начала очереди. Если очередь пуста, возвращает значение null
- E **peekLast()**: возвращает без удаления последний элемент очереди. Если очередь пуста, возвращает значение null
- E **pollFirst()**: возвращает с удалением элемент из начала очереди. Если очередь пуста, возвращает значение null
- E **pollLast()**: возвращает с удалением последний элемент очереди. Если очередь пуста, возвращает значение null
- E **pop()**: возвращает с удалением элемент из начала очереди. Если очередь пуста, генерирует исключение NoSuchElementException
- void **push(E element)**: добавляет элемент в самое начало очереди

II

FIFO & LIFO

Обобщенный интерфейс Deque<E> (двунаправленная очередь)

- E **removeFirst()**: возвращает с удалением элемент из начала очереди. Если очередь пуста, генерирует исключение NoSuchElementException
- E **removeLast()**: возвращает с удалением элемент из конца очереди. Если очередь пуста, генерирует исключение NoSuchElementException
- boolean **removeFirstOccurrence(Object obj)**: удаляет первый встреченный элемент obj из очереди. Если удаление произошло, то возвращает true, иначе возвращает false.
- boolean **removeLastOccurrence(Object obj)**: удаляет последний встреченный элемент obj из очереди. Если удаление произошло, то возвращает true, иначе возвращает false.

III

FIFO & LIFO

Класс `ArrayDeque<E>` реализует интерфейс `Deque`

В классе `ArrayDeque` определены следующие конструкторы:

- **`ArrayDeque()`**: создает пустую очередь
- **`ArrayDeque(Collection<? extends E> col)`**: создает очередь, наполненную элементами из коллекции `col`
- **`ArrayDeque(int capacity)`**: создает очередь с начальной емкостью `capacity`. Если мы явно не указываем начальную емкость, то емкость по умолчанию будет равна 16

Класс `ArrayDeque<E>` реализует интерфейс `Deque`

В классе `ArrayDeque` определены следующие конструкторы:

- **`ArrayDeque()`**: создает пустую очередь
- **`ArrayDeque(Collection<? extends E> col)`**: создает очередь, наполненную элементами из коллекции `col`
- **`ArrayDeque(int capacity)`**: создает очередь с начальной емкостью `capacity`. Если мы явно не указываем начальную емкость, то емкость по умолчанию будет равна 16

