```java
package components.queue;

import components.queue.QueueSecondary;
import components.sequence.Sequence;
import components.sequence.Sequence1L;

/**
 * {@code Queue} represented as a {@code Sequence} of entries, with
 * implementations of primary methods.
 *
 * @param <T>
 *            type of {@code Queue} entries
 * @correspondence this = $this.entries
 */
public class Queue3<T> extends QueueSecondary<T> {

    /*
     * Private members --------------------------------------------------------
     */

    /**
     * Entries included in {@code this}.
     */
    private Sequence<T> entries;

    /**
     * Creator of initial representation.
     */
    private void createNewRep() {
        this.entries = new Sequence1L<T>();
    }

    /*
     * Constructors -----------------------------------------------------------
     */

    /**
     * No-argument constructor.
     */
    public Queue3() {
        this.createNewRep();
    }

    /*
     * Standard methods removed to reduce clutter...
     */

    /*
     * Kernel methods ---------------------------------------------------------
     */

    @Override
    public final void enqueue(T x) {
        assert x != null : "Violation of: x is not null";

        this.entries.add(this.entries.length(), x);

    }

    @Override
    public final T dequeue() {
        assert this.length() > 0 : "Violation of: this /= <>";

        T result = this.entries.remove(0);

        return result;
    }

    @Override
    public final int length() {

        int result = this.entries.length();

        return result;
    }

    /*
     * Iterator removed to reduce clutter...
     */

    /**
     * Reports the front of {@code this}.
     *
     * @return the front entry of {@code this}
     * @aliases reference returned by {@code front}
     * @requires this /= <>
     * @ensures <front> is prefix of this
     */
    @Override
    public T front() {
        assert this.length() > 0 : "Violation of: this /= <>";

        T result = this.entries.entry(0);

        return result;
    }

}
```