

CSC1110 Exercise 9 – Kaiden Pollesch

Generating Documentation using Chat-GPT

1. My Javadoc Comments

```
/**
 * Is the main interface for a Persons Calendar
 */
public interface Schedulable {

    /**
     * To input a day and hour and see if the Calendar is filled during that time,
     * or if it is free during that time in that persons calendar.
     * @param day the day that they want to schedule with person
     * @param hour the hour that they want to schedule with person
     * @return true if they can be scheduled at that time or false if they cannot.
     */
    boolean isBusy(int day, int hour);

    /**
     * This will try and add an item to the calendar
     * @param item will be the item added to the calendar
     * @return true if it was added return false if it wasn't added.
     */
    boolean add(Schedulable item);

    /**
     * This will try and remove an item to the calendar
     * @param item will be the item removed from the calendar
     * @return true if it was removed return false if it wasn't removed
     */
    boolean remove(Schedulable item);

    /**
     * This will get the name of the item being added to the calendar
     * @return the name of the item being added to the calendar
     */
    String getName();
}
```

2. Chat-GPT Javadoc Comments

```
/**
 * The {@code Schedulable} interface represents entities that can be scheduled and
 * managed within a time-based system.
 * Implementing classes are expected to provide functionality for checking
 * availability, adding and removing scheduled items,
 * and retrieving the name of the schedulable entity.
 */
public interface Schedulable {

    /**
     * Checks if the entity is busy at the specified day and hour.
     *
     * @param day    The day of the week, where 1 represents Monday and 7 represents
     * Sunday.
     * @param hour   The hour of the day in 24-hour format (0 to 23).
     * @return       {@code true} if the entity is busy at the specified day and hour,
     {@code false} otherwise.
     */
    boolean isBusy(int day, int hour);

    /**
     * Adds a schedulable item to the entity's schedule.
     *
     * @param item   The {@code Schedulable} item to be added to the schedule.
     * @return       {@code true} if the item is successfully added, {@code false} if
     the addition fails (e.g., due to conflicts).
     */
    boolean add(Schedulable item);

    /**
     * Removes a schedulable item from the entity's schedule.
     *
     * @param item   The {@code Schedulable} item to be removed from the schedule.
     * @return       {@code true} if the item is successfully removed, {@code false}
     if the removal fails (e.g., item not found).
     */
    boolean remove(Schedulable item);

    /**
     * Retrieves the name of the schedulable entity.
     *
     * @return       A {@code String} representing the name of the schedulable entity.
     */
    String getName();
}
```

3. Compare

Both Javadocs are very similar, and they are both saying the same thing. I think the only difference is that the AI-generated ones use more coding language and maybe a little more in-depth to try and communicate their ideas better.

I think the AI-generated one is better because it goes into a little more detail, uses the words that come with coding languages and uses better terminology.

4. UML Diagram for Schedulable

```
+-----+
|   Schedulable   |
+-----+
| - isBusy(day: int, hour: int): boolean
| - add(item: Schedulable): boolean
| - remove(item: Schedulable): boolean
| - getName(): String
```

Clearly doesn't know exactly what the UML diagram should look like it doesn't have `<<interface>>` at the top, also those methods should not be private.

5. Generate 2 UML diagrams

```
+-----+
|      Calendar      |
+-----+
| - DEFAULT_START_OF_DAY: int = 8
| - DEFAULT_END_OF_DAY: int = 17
| - items: List<Schedulable>
| - name: String
| - startOfDay: int
| - endOfDay: int
|
| + Calendar(name: String, startOfDay: int,
|           endOfDay: int, items: Schedulable...)
| + Calendar(name: String, items: Schedulable...)
| + isBusy(day: int, hour: int): boolean
| + add(item: Schedulable): boolean
| + remove(item: Schedulable): boolean
| + getName(): String
| + toString(): String
| - getScheduleMatrix(): int[][]
+-----+
```

6. Problems

Well first it left the the whole interface Schedulable blank as if that file were just empty. Also where we have a public static final ints, it shows those as private so, if someone were to build off of this UML their variables would not be able to be accessed by other classes. It also forgot to add the final part {readOnly} as well as the top two being Static (Underlined). The class type at the top defining an interface <<interface>>. These are all of the things that I would change.

7. My UML diagrams

