Code Inspection-Assigned class: TechDataServices

- C27 L.224 L.378

Both the switch blocks are a copy paste in our service which paves way for duplicates. But this is quite preferable here rather than using different piece of code for the same logical implementation (i.e. for a better understanding).

- C27 L.218 L.271 L.301 L.344 L.372 L.425 L.455 L.500

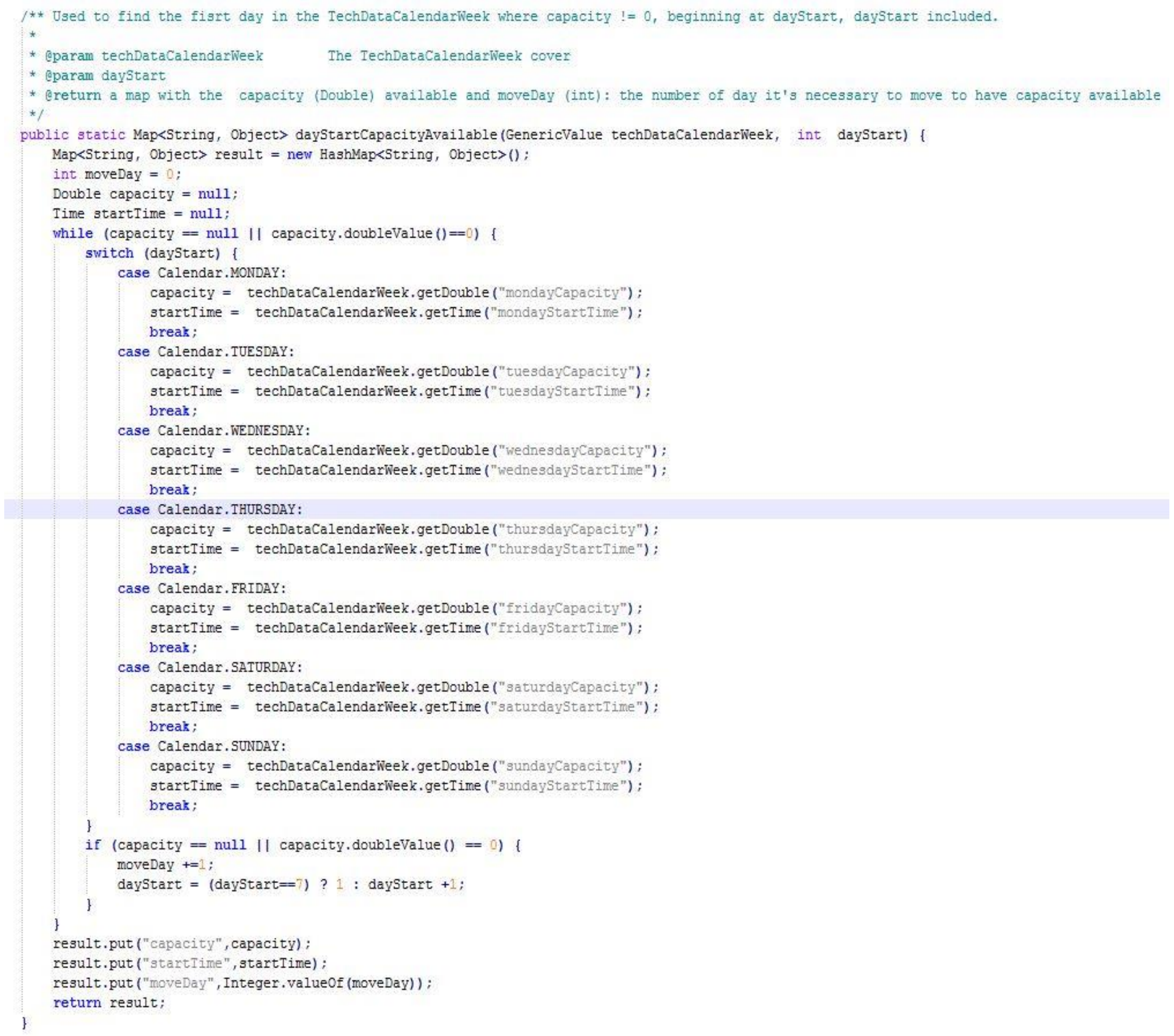
All the methods are long more than 10 lines and some methods are even more than 20 lines. This should be optimized by using fewer conditional checks covering all the requirements.

- C54

All switch statements are addressed by a break.

- C55 L.224 L.378

There is no default branch for any of the switch statements. The solutions we suggest for this problem is to leave the code undisturbed, because all the days are present in the switch cases such that it executes any one of the switch case and so no default is needed. Without knowing the dayStart or dayEnd it would be wrong to define a default branch here.



```java
/** Used to find the fisrt day in the TechDataCalendarWeek where capacity != 0, beginning at dayStart, dayStart included.
 *
 * @param techDataCalendarWeek        The TechDataCalendarWeek cover
 * @param dayStart
 * @return a map with the  capacity (Double) available and moveDay (int): the number of day it's necessary to move to have capacity available
 */
public static Map<String, Object> dayStartCapacityAvailable(GenericValue techDataCalendarWeek,  int  dayStart) {
    Map<String, Object> result = new HashMap<String, Object>();
    int moveDay = 0;
    Double capacity = null;
    Time startTime = null;
    while (capacity == null || capacity.doubleValue()==0) {
        switch (dayStart) {
            case Calendar.MONDAY:
                capacity =  techDataCalendarWeek.getDouble("mondayCapacity");
                startTime =  techDataCalendarWeek.getTime("mondayStartTime");
                break;
            case Calendar.TUESDAY:
                capacity =  techDataCalendarWeek.getDouble("tuesdayCapacity");
                startTime =  techDataCalendarWeek.getTime("tuesdayStartTime");
                break;
            case Calendar.WEDNESDAY:
                capacity =  techDataCalendarWeek.getDouble("wednesdayCapacity");
                startTime =  techDataCalendarWeek.getTime("wednesdayStartTime");
                break;
            case Calendar.THURSDAY:
                capacity =  techDataCalendarWeek.getDouble("thursdayCapacity");
                startTime =  techDataCalendarWeek.getTime("thursdayStartTime");
                break;
            case Calendar.FRIDAY:
                capacity =  techDataCalendarWeek.getDouble("fridayCapacity");
                startTime =  techDataCalendarWeek.getTime("fridayStartTime");
                break;
            case Calendar.SATURDAY:
                capacity =  techDataCalendarWeek.getDouble("saturdayCapacity");
                startTime =  techDataCalendarWeek.getTime("saturdayStartTime");
                break;
            case Calendar.SUNDAY:
                capacity =  techDataCalendarWeek.getDouble("sundayCapacity");
                startTime =  techDataCalendarWeek.getTime("sundayStartTime");
                break;
        }
        if (capacity == null || capacity.doubleValue() == 0) {
            moveDay +=1;
            dayStart = (dayStart==7) ? 1 : dayStart +1;
        }
    }
    result.put("capacity",capacity);
    result.put("startTime",startTime);
    result.put("moveDay",Integer.valueOf(moveDay));
    return result;
}
```