**Сессия 2**

**Разработка библиотеки**

Разработайте библиотеку (отдельным проектом), которая позволит вернуть список свободных временных интервалов (заданного размера) в графике сотрудника для формирования оптимального графика работы сотрудников.

Необходимо обязательно следовать правилам именования библиотек, классов и методов в них. В случае ошибок в рамках именования ваша работа не может быть проверена и ваш результат не будет зачтен. Классы и методы должны содержать модификатор public (если это реализуемо в рамках платформы), чтобы внешние приложения могли получить к ним доступ.

Вход:

- список занятых промежутков времени (в двух массивах: startTimes - начало, durations - длительность),
- минимальное необходимое время для работы менеджера (consultationTime),
- рабочий день сотрудника (начало - beginWorkingTime и завершение - endWorkingTime)

Выход:

- список подходящих свободных временных промежутков (в массив строк формата HH:mm-HH:mm)

Требования к именованиям и форматам:

| | C# | Java | Python |
|---|---|---|---|
| Библиотека классов | SF2022User{NN}Lib.dll | SF2022User{NN}Lib.jar | SF2022User{NN}Lib |
| Название класса | Calculations | Calculations | Calculations |
| Название метода | AvailablePeriods() | availablePeriods() | available periods() |
| Входящие обязательные параметры | TimeSpan[] startTimes,<br>int[] durations,<br>TimeSpan beginWorkingTime,<br>TimeSpan endWorkingTime,<br>int consultationTime | LocalTime[] startTimes,<br>int[] durations,<br>LocalTime beginWorkingTime,<br>LocalTime endWorkingTime,<br>int consultationTime | time[] start times,<br>int[] durations,<br>time begin working time,<br>time end working time,<br>int consultation time |
| Возвращаемые параметры | string[] | string[] | string[] |

Пример:

| Вход | Выход |
|---|---|
| startTime \| duration<br>10:00 60<br>11:00 30<br>15:00 10<br>15:30 10<br>16:50 40<br>Working Times<br>08:00-18:00<br>Consultation Time<br>30 | 08:00-08:30<br>08:30-09:00<br>09:00-09:30<br>09:30-10:00<br>11:30-12:00<br>12:00-12:30<br>12:30-13:00<br>13:00-13:30<br>13:30-14:00<br>14:00-14:30<br>14:30-15:00<br>15:40-16:10<br>16:10-16:40<br>17:30-18:00 |

## Модуль 7: Тестирование программных решений

### Тестирование

Модульные тесты

Реализуйте 10 unit-тестов на основе технологии TDD для библиотеки, функционал которой описан ранее. Важно, чтобы тестовые данные предусматривали различные ситуации. Например, недостаточное время в промежутках между ранее созданными консультациями, либо в начале рабочего дня, либо в конце рабочего дня; различная длительность консультация и т.д.

### Тестовая документация

Для выполнения процедуры тестирования добавления товара администратором Вам нужно описать пять сценариев. Добавление может быть выполнено успешно, а может быть отклонено согласно требованиям предметной области. Необходимо, чтобы варианты тестирования демонстрировали различные исходы работы алгоритма. Для описания тестовых сценариев в ресурсах предоставлен шаблон testing-template.docx.