|  |  |  |
| --- | --- | --- |
| Федеральное государственное автономное образовательное учреждение  высшего образования «Национальный исследовательский  Нижегородский государственный университет им. Н.И. Лобачевского» НИИМ Нижегородского университета | | |
| УДК  № госрегистрации  Инв. № | | **УТВЕРЖДАЮ**  Профессор кафедры  ИАНИ ННГУ, д.т.н.  \_\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Старостин  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г. |
| Научно-технический отчет  **разработка программного средства**  **«Разработка и реализация программного обеспечения планирования графика спортивных мероприятий» (ПО «График»)»** | | |
|  |  | |
|  |  | |
|  |  | |

2019

Оглавление

[ВВЕДЕНИЕ 3](#_Toc9893056)

[Разработка алгоритма автоматизации задачи составления графика спортивных мероприятий. 4](#_Toc9893057)

[1 Входные данные 4](#_Toc9893058)

[1.1 Файл исходных данных задачи 4](#_Toc9893059)

[1.2 Файл конфигурации системы 7](#_Toc9893060)

[2 Выходные данные 7](#_Toc9893061)

[2.1 Решение 7](#_Toc9893062)

[2.2 Метрики 8](#_Toc9893063)

[2.3 Лог-файл 8](#_Toc9893064)

[1 Содержательная постановка задачи 9](#_Toc9893065)

[2 Математическая постановка задачи 9](#_Toc9893066)

[2.1 Исходные параметры 9](#_Toc9893067)

[2.2 Варьируемые параметры 10](#_Toc9893068)

[2.3 Ограничения 10](#_Toc9893069)

[2.4 Критерии оптимальности 11](#_Toc9893070)

[1 Алгоритм «Жадный» 12](#_Toc9893071)

[1.1 Описание алгоритма 12](#_Toc9893072)

[1.2 Идея алгоритма 12](#_Toc9893073)

[2 Алгоритм «Локального поиска» 13](#_Toc9893074)

[2.1 Описание алгоритма 13](#_Toc9893075)

[2.2 Идея алгоритма 13](#_Toc9893076)

[1 Общее описание подсистемы тестирования 14](#_Toc9893077)

[2 Детальное описание модулей 14](#_Toc9893078)

[2.1 Генератор тестовых задач 14](#_Toc9893079)

[2.2 Модуль анализа 14](#_Toc9893080)

[3 Пример работы программы 15](#_Toc9893081)

[1 Описание эксперимента 16](#_Toc9893082)

[2 Результаты эксперимента 17](#_Toc9893083)

[3 Выводы 20](#_Toc9893084)

[ЗАКЛЮЧЕНИЕ 22](#_Toc9893085)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc9893086)

# ВВЕДЕНИЕ

Рассматривается проблема разработки программного средства составления расписаний.

В рамках данного проекта предполагались следующие работы:

Разработка алгоритма автоматизации процесса составления графика спортивных мероприятий обеспечивающего решение задач составления расписания, с учётом требований команд логистики, мест проведения спортивных мероприятий, составляющего расписание в автоматическом режиме.

## Разработка алгоритма автоматизации задачи составления графика спортивных мероприятий.

# 1 Входные данные

Исходные данные поступают в систему в формате .xml и представляют собой файл исходных данных задачи и файл конфигурации.

## 1.1 Файл исходных данных задачи

Файл исходных данных задачи должен содержать следующую информацию:

* информация о чемпионате:
  + дата начала;
  + дата окончания;
  + список игровых дней;
  + список нежелательных дней для игр;
  + количество игр между парой команд;
* список команд. Для каждой команды должны быть указаны:
  + название;
  + бинарный признак лидирующей команды;
  + список нежелательных дней для игр;
* информация о месте проведения чемпионата:
  + название;
  + минимальное возможное число игр в день;
  + максимальное возможное число игр в день;
  + множество временных слотов возможного проведения игр. Под временным слотом понимается число от 0 до n. Каждое число соответствует фиксированному суточному интервалу. Число n определяется максимальным возможным числом игр в день.

Структура XML-документа исходных данных задачи представлена в Листинг 1. Пример исходных данных приведён на Рисунке 1.

Листинг 1. XSD-схема документа исходных данных

<?xml version="1.0" encoding="UTF-8" ?>

<xs:schema id="schedule" targetNamespace="http://schedule.xsd" elementFormDefault="qualified" xmlns="http://schedule.xsd" xmlns:mstns="http://schedule.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!-- simple types begin -->

<xs:simpleType name="itemType">

<xs:restriction base="xs:string">

<xs:pattern value="[0-2][0-9]:[0-5][0-9]" />

</xs:restriction>

</xs:simpleType>

<!-- simple types end -->

<!-- championship begin -->

<xs:complexType name="daysType">

<xs:sequence>

<xs:element name="weekDay" type="xs:string" maxOccurs="7"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="reservType">

<xs:sequence>

<xs:element name="date" type="xs:date" minOccurs="0" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="decreesType">

<xs:sequence>

<xs:element name="decree" type="xs:date" minOccurs="0" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="toursType">

<xs:sequence>

<xs:element name="tour" type="xs:decimal" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="championshipType" >

<xs:sequence>

<xs:element name="days" type="daysType" />

<xs:element name="reserv" type="reservType" />

<xs:element name="decrees" type="decreesType" />

<xs:element name="tours" type="toursType" />

</xs:sequence>

<xs:attribute name="start" type="xs:date" use="required" />

<xs:attribute name="end" type="xs:date" use="required" />

<xs:attribute name="coupleGames" type="xs:positiveInteger" use="required" />

</xs:complexType>

<!-- championship end -->

<!-- stadium begin -->

<xs:complexType name="timeType">

<xs:sequence>

<xs:element name="item" type="itemType" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="stadiumType">

<xs:sequence>

<xs:element name="time" type="timeType" />

</xs:sequence>

<xs:attribute name="name" type="xs:string" use="required" />

<xs:attribute name="minGame" type="xs:nonNegativeInteger" use="required" />

<xs:attribute name="maxGame" type="xs:positiveInteger" use="required" />

</xs:complexType>

<!-- stadium end -->

<!-- teams begin -->

<xs:complexType name="slotType">

<xs:attribute name="day" type="xs:string" use="required" />

<xs:attribute name="time" type="itemType" use="required" />

</xs:complexType>

<xs:complexType name="slotsType">

<xs:sequence>

<xs:element name="slot" type="slotType" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="periodType">

<xs:attribute name="day" type="xs:string" use="required" />

<xs:attribute name="number" type="xs:positiveInteger" use="required" />

</xs:complexType>

<xs:complexType name="periodsType">

<xs:sequence>

<xs:element name="period" type="periodType" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="commandType">

<xs:sequence>

<xs:element name="slots" type="slotsType" minOccurs="0" />

<xs:element name="periods" type="periodsType" minOccurs="0" />

</xs:sequence>

<xs:attribute name="name" type="xs:string" use="required" />

<xs:attribute name="leader" type="xs:boolean" use="required" />

</xs:complexType>

<xs:complexType name="teamsType">

<xs:sequence>

<xs:element name="command" type="commandType" minOccurs="2" maxOccurs="100" />

</xs:sequence>

</xs:complexType>

<!-- teams end -->

<!-- schedule begin -->

<xs:element name="sсhedule">

<xs:complexType>

<xs:sequence>

<xs:element name="championship" type="championshipType" />

<xs:element name ="stadium" type="stadiumType" />

<xs:element name="teams" type="teamsType" />

</xs:sequence>

</xs:complexType>

</xs:element>

<!-- schedule end -->

</xs:schema>



Рисунок 1. Пример исходных данных

## 1.2 Файл конфигурации системы

Конфигурационный файл должен содержать настройки преобразования исходных данных задачи в программную модель, настройки проверки программной модели, настройки блока выбора алгоритма решения текущей задачи, выбор метрики для расчета. Структура XML-документа файла конфигурации представлена на Рисунке 2.



Рисунок 2. Структура XML-документа конфигурационного файла

# 2 Выходные данные

Выходными данными являются HTML-документ с решением поставленной задачи, HTML-документ со значениями метрик и txt-файл с протоколом работы системы.

## 2.1 Решение

HTML-документ с решением задачи должен содержать следующую информацию: расписание матчей в виде таблицы, в которой строки/столбцы соответствуют командам, а в каждой ячейке (кроме диагональных) отображается дата и временной слот игры команды строки с командой столбца. Пример решения задачи представлен на Рисунке 3.

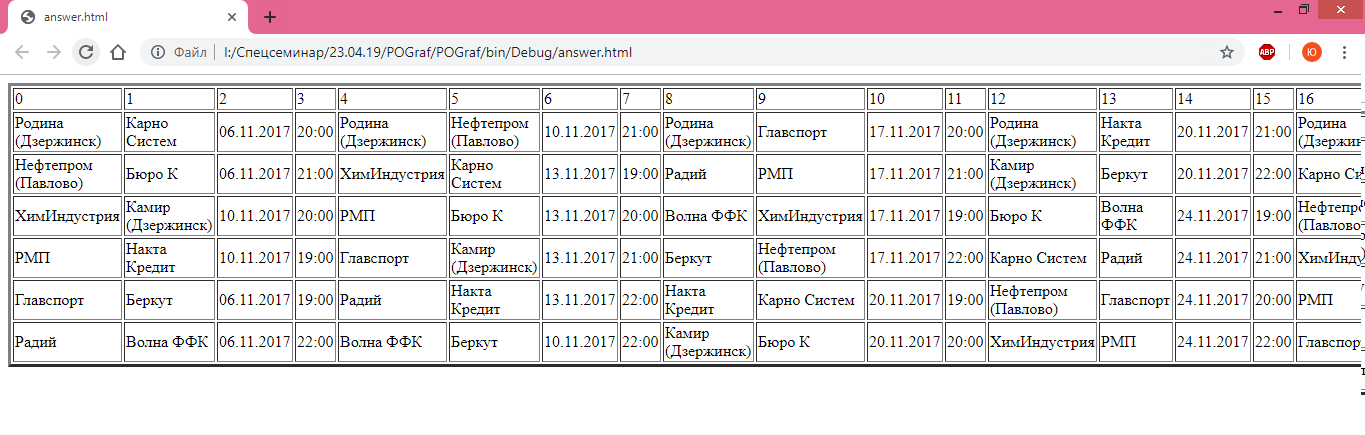


Рисунок 3. Пример HTML-страницы с расписанием

## 2.2 Метрики

HTML-документ со значениями метрик задачи должен содержать следующую информацию: таблицу со значениями критериев. Пример решения задачи представлен на Рисунок 4.

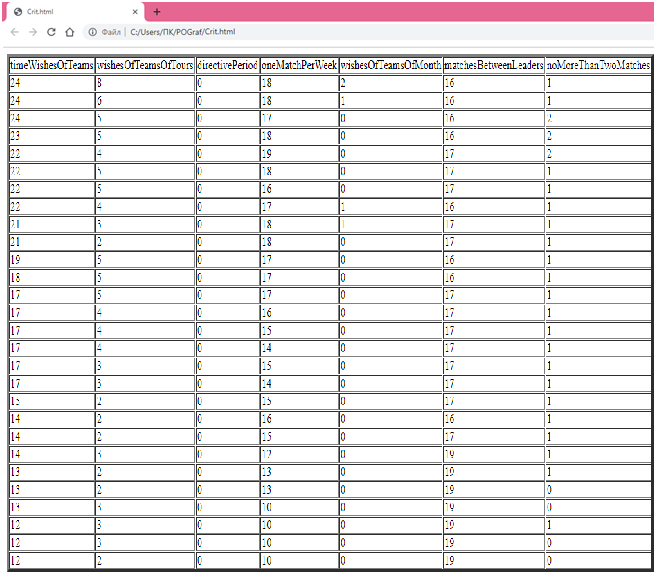


Рисунок 4. Пример HTML-страницы со значениями критериев

## 2.3 Лог-файл

Вместе с решением в качестве выходных данных система генерирует лог-файл (документ формата .txt), содержащий протокол работы системы, в котором должны отражаться:

* результат проверки исходных данных;
* данные о стратегии (алгоритме), использованной при построении расписания;
* результат работы алгоритма;
* в случае получения результата – численные значения метрик, включая нарушения ограничений/требований команд.

# 1 Содержательная постановка задачи

Рассматривается задача составления календаря спортивных матчей. Имеются команды и один стадион. Матчи разыгрываются по круговой системе, при которой каждый участник турнира играет с каждым в ходе тура. Чемпионат проходит в два круга.

Для одного матча требуется один час времени. Время работы стадиона ограничено определенными днями и часами. Команды выдвигают свои пожелания по расписанию. Для составления более зрелищного турнира необходимо учитывать лидеров и ставить их определенным образом.

Требуется составить календарь спортивных матчей, который как можно лучше бы учитывал пожелания команд, требования по директивному сроку проведения чемпионата и был более зрелищным для зрителей.

# 2 Математическая постановка задачи

## 2.1 Исходные параметры

– число команд

– число месяцев

– число недель

– число дней

– число дней в директивном сроке

– число часовых слотов для матчей в день

– минимальное число игр за один игровой день

– максимальное число игр за один день

– номера -ых дней недели,

– номера дней -ой недели,

– номера дней -ого месяца,

– номера команд лидеров

– номера зрелищных туров

– номера приоритетных часовых слотов -ой команды,

– номера приоритетных дней недели в равных долях в течение месяца -ой команды,

– номера приоритетных дней недели для одного матча раз в четыре тура -ой команды,

## 2.2 Варьируемые параметры

– номер -ой команды в -ом матче -ого тура, , ,

– номер дня для -ого матча -ого тура, ,

– номер часа для -ого матча -ого тура, ,

## 2.3 Ограничения

, , ,

, ,

, ,

, ,

, , , , – команда играет один раз в туре

, , , – соперники не повторяются в первом круге

, , , – соперники не повторяются во втором круге

, , , – в одно время не больше одного матча

, , – туры проходят последовательно

– число матчей в день ограничено

## 2.4 Критерии оптимальности

– минимизация используемых резервных дней за пределами директивного срока

– каждая команда играет один матч в неделю

– матчи между лидерами проходят во время зрелищных туров

– за один день не более 2 матчей между лидерами

– пожелания команд по времени

– пожелания команд по дням недели

– пожелания команд по равным долям дней недели в течение месяца

– пожелания команд по дням недели для одного матча раз в четыре тура

# 1 Алгоритм «Жадный»

## 1.1 Описание алгоритма

Алгоритм заключается в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным.

Принцип принятия решения: детерминированный.

Тип используемых пространств: последовательный.

## 1.2 Идея алгоритма

1. Распределение матчей по турам
   1. Построение допустимого распределения матчей по турам по круговой системе
   2. Сортировка туров по числу матчей между лидерами по убыванию
   3. Последовательное размещение туров сначала в позиции зрелищных туров, потом в остальные
2. Распределение матчей по дням
   1. Последовательное размещение туров по дням
      1. Выделение временного интервала для текущего тура из оставшегося с учетом равного распределения для всех не размещенных туров
      2. Последовательное размещение матчей текущего тура при наличии подходящих предпочтений по дням
      3. Последовательное размещение матчей текущего тура в игровые дни с недостаточным числом игр
      4. Последовательное размещение оставшихся матчей текущего тура
      5. Перемещение матчей из оставшихся игровых дней с недостающим числом игры, кроме последнего игрового дня, в предыдущие дни
3. Распределение матчей по часам
   1. Последовательное размещение туров по часам
      1. Определение свободных часов в последний игровой день
      2. Последовательное размещение матчей текущего тура при наличии подходящих предпочтений по часам
      3. Последовательное размещение оставшихся матчей текущего тура

# 2 Алгоритм «Локального поиска»

## 2.1 Описание алгоритма

Алгоритм берёт потенциальное решение задачи и проверяет его непосредственных соседей (то есть решения, которые похожи, за исключением нескольких очень малых деталей) в надежде нахождения улучшенного решения.

Принцип принятия решения: стохастический.

Тип используемых пространств: итерационный.

Тип формируемой траектории: траекторно-непрерывный.

## 2.2 Идея алгоритма

1. Перемещение команд в расписании
   1. Сохранение отдельной копии текущего решения
   2. Подсчет числа конфликтных ситуаций, влияющих на критерий, для каждой команды
   3. Выбор двух команд случайным образом, отдающим большую вероятность выбора при большем числе конфликтных ситуаций
   4. Замена одной выбранной команды другой в турнирном расписании матчей
   5. Построение нового расписания матчей по дням и часам методами «Жадного» алгоритма
   6. Сравнение значений критерия полученного решения и сохраненной копии прошлого
   7. Сохранение одного решения с наилучшим критерием
2. Перемещение туров в расписании
   1. Сохранение отдельной копии текущего решения
   2. Подсчет числа конфликтных ситуаций, влияющих на критерий, для каждого тура
   3. Выбор двух туров случайным образом, отдающим большую вероятность выбора при большем числе конфликтных ситуаций
   4. Замена одного выбранного тура другим в турнирном расписании
   5. Построение нового расписания матчей по дням и часам методами «Жадного» алгоритма
   6. Сравнение значений критерия полученного решения и сохраненной копии прошлого
   7. Сохранение одного решения с наилучшим критерием

# 1 Общее описание подсистемы тестирования

Программа Test Infrastructure предназначена для тестирования ПО «График» и сбора статистики по работе этой программы на сгенерированом множестве задач для дальнейшей демонстрации конечному пользователю.

Инфраструктура тестирования содержит два основных компонента:

* генератор множества тестовых задач;
* модуль анализа.

Модуль анализа служит для сбора статистики (под статистикой понимаются значения критериев оптимизации, описанных в математической модели) и демонстрации конечному пользователю.

# 2 Детальное описание модулей

## 2.1 Генератор тестовых задач

Данный модуль генерирует множество задач, на котором выполняется проверка на соответствие формату, на полноту и непротиворечивость исходной информации.

Множество задач состоит из трех основных групп:

* корректные задачи;
* задачи с несовместными данными;
* задачи, не соответствующие установленному формату оформления входных данных.

Работа модуля заключается в следующем: в заданной директории создаются 3 папки, соответствующие определенному типу задач, в каждую из которых помещаются сгенерированные в формате XML задачи.

## 2.2 Модуль анализа

Данный модуль представляет собой систему запуска задач и дальнейшего анализа результатов запуска для последующего предоставления статистики по всем запускам конечному пользователю.

Формат сводного отчета по всем задачам – таблица, содержащая значения метрик (критериев математической модели).

Результат выполнения программы представлен на Рисунок *1*.

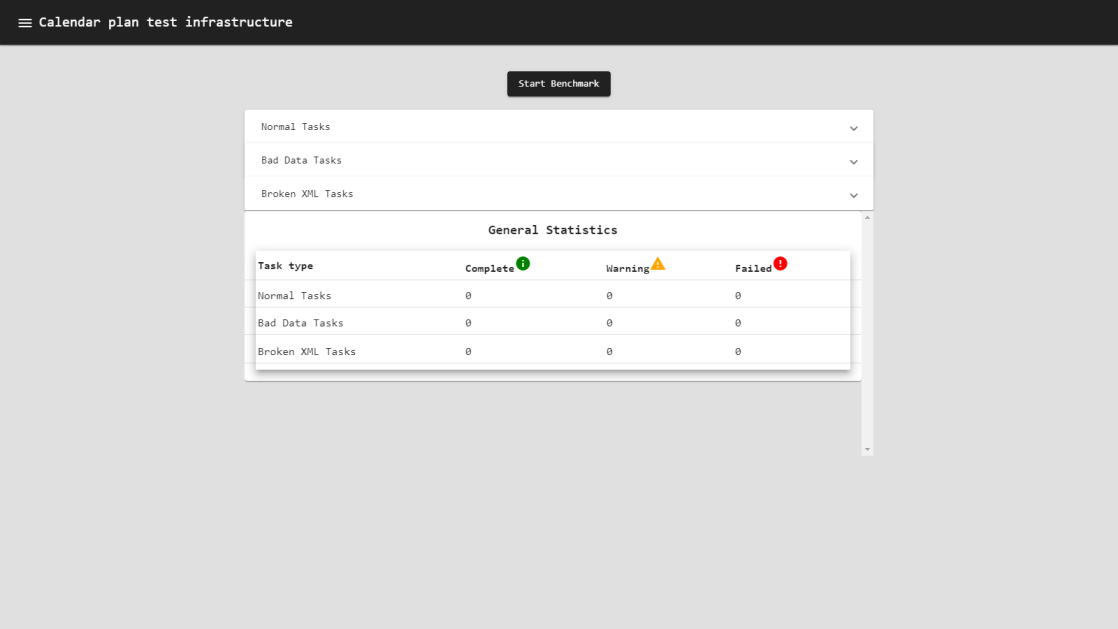


Рисунок 1. Результат работы системы тестирования

Таблица с названием General Statistic содержит данные о результатах тестов для каждой категории задач. В столбце «Complete» отображается число пройденных тестов, в столбце «Warning» – число задач, на которых при прохождении теста был получен не тот результат, который ожидался, в столбце «Failed» – число непройденных тестов.

Значки рядом с именами колонок используются при маркировании списка тестируемых задач (Рисунок 2)

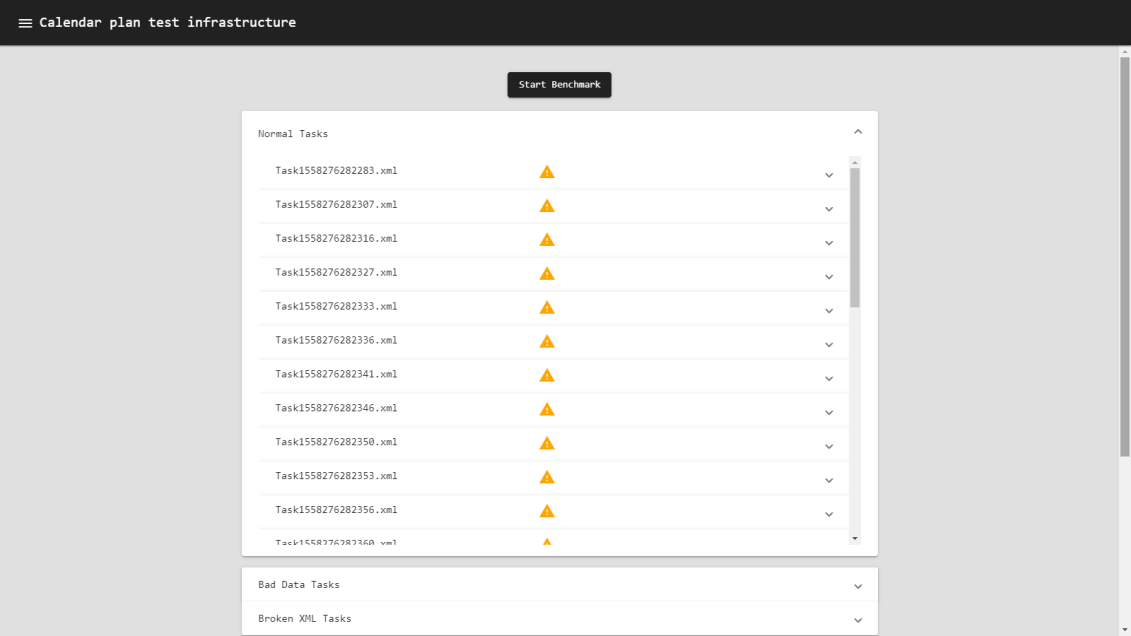


Рисунок 2. Маркированный список задач

# 3 Пример работы программы

Для запуска программы необходимо дважды кликнуть (ЛКМ) по файлу «Start for Windows.bat». После чего запустится программа: откроется окно браузера с кратким описанием инфраструктуры тестирования.

Затем необходимо кликнуть по значку C:\Users\Mr user\AppData\Local\Temp\Tmp_view\2x\baseline_menu_black_18dp.png: откроется боковое меню (Рисунок *3*), в котором нужно выбрать пункт Analyser.

# 1 Описание эксперимента

Для проведения эксперимента тестовой инфраструктурой были случайно сгенерированы 3 набора исходных данных:

* корректные задачи;
* задачи с несовместными данными (например, время игры «35 часов и 79 минут»);
* задачи, не соответствующие установленному формату оформления входных данных.

Далее подсистемой тестовой инфраструктуры производился последовательный запуск задач каждого из наборов на ПО «График».

При этом модулем анализа тестовой инфраструктуры осуществлялся сбор статистики (под статистикой понимаются значения критериев оптимизации, описанных в математической модели) для дальнейшей демонстрации пользователю.

# 2 Результаты эксперимента

Для каждой задачи из сгенерированных наборов исходных данных отображается список со значениями критериев при решении жадным алгоритмом и алгоритмом локального поиска (Рисунок 1). Приоритет критериев следующий:

1. Пожелания команд по времени (timeWishesOfTeams);
2. Пожелания команд по дням недели для одного матча раз в 4 тура (wishesOfTeams);
3. Расписание укладывается в директивный срок (directivePeriod);
4. Каждая команда играет один матч в неделю (oneMatchPerWeek);
5. Матчи между лидерами проходят во время зрелищных туров (matchesBetweenLeaders);
6. За один день не более 2 матчей между лидерами (noMoreThanTwoMatches).

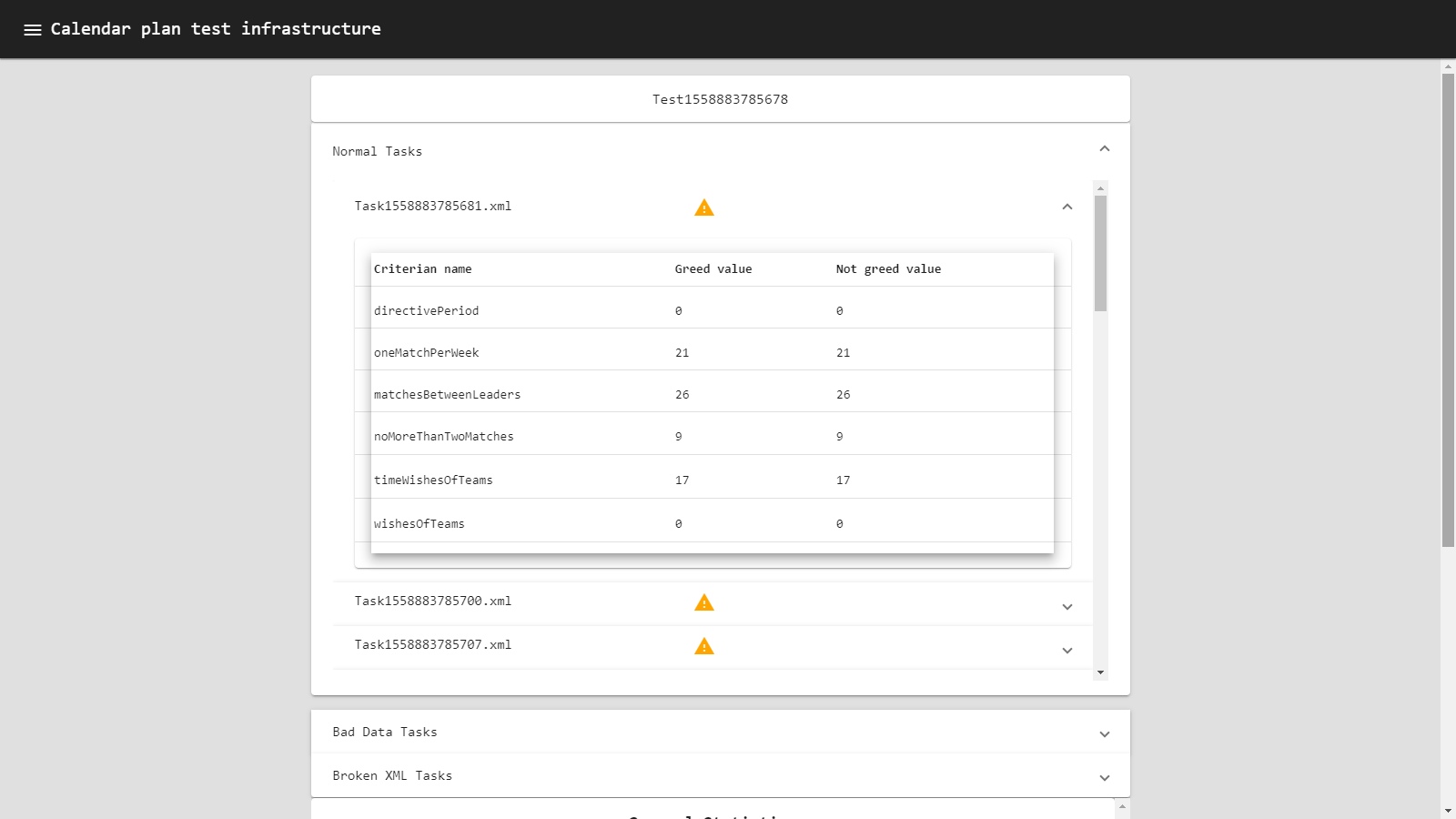


Рисунок 1. Значения критериев конкретной задачи

Результаты работы тестовой инфраструктуры по всем сгенерированным задачам отображаются в сводной таблице (Рисунок 2).

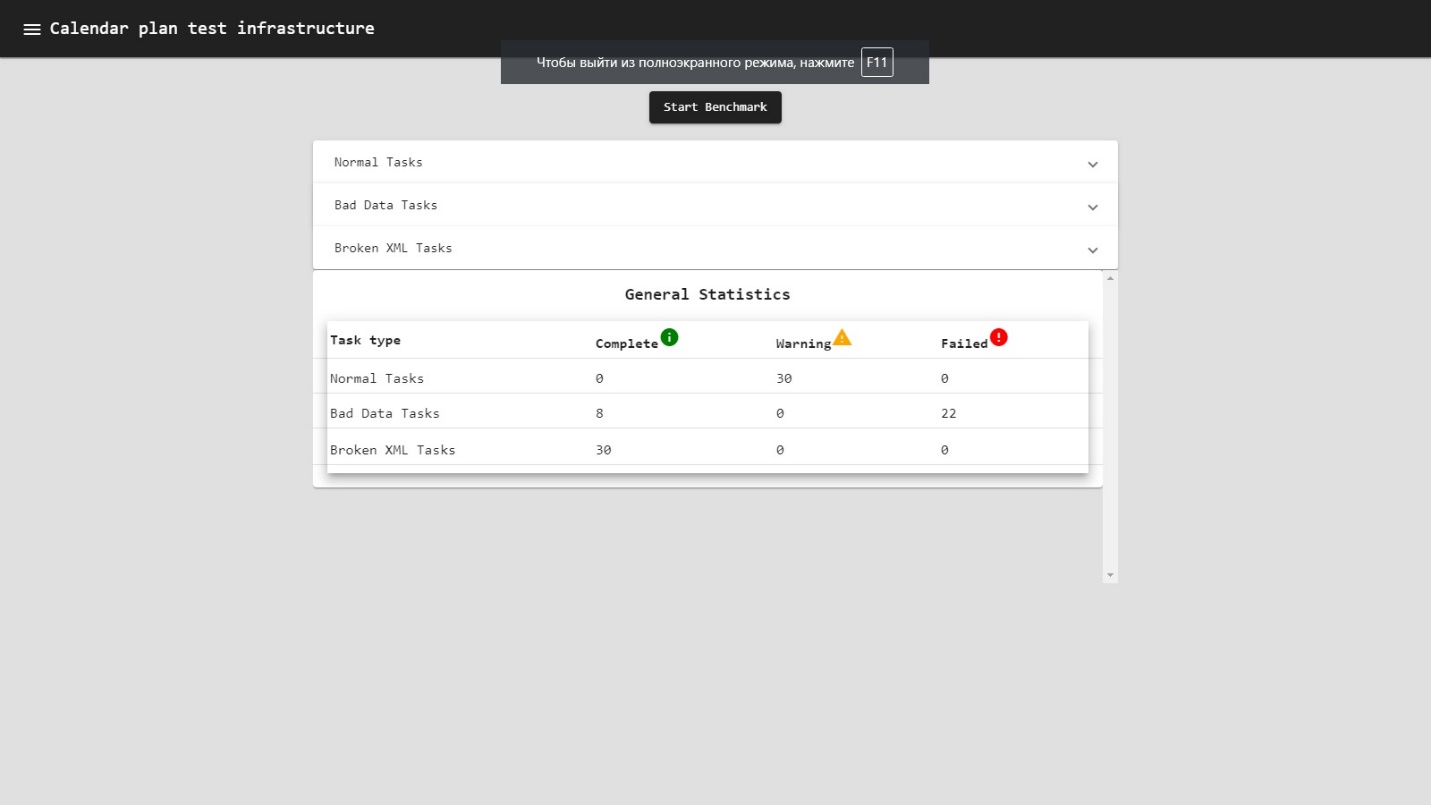


Рисунок 2. Сводная таблица результатов

Из таблицы видно, что после запуска тестов для категории нормальных задач ни в одном запуске линейный алгоритм не превзошел жадный. При просмотре значений критериев при решении жадным алгоритмом и алгоритмом локального поиска выясняется, что полученные значения равны. Это можно объяснить недостаточной сложностью сгенерированных тестовой инфраструктурой исходных данных, вследствие чего решение, полученное жадным алгоритмом, является оптимальным и не может быть улучшено при решении задачи алгоритмом локального поиска. Это, в свою очередь, не говорит о некорректности «продвинутого» алгоритма (алгоритма локального поиска), что подтверждается проведением эксперимента на более сложных задачах (Таблица 1).

Таблица 1 – Сравнение работы алгоритмов на более сложных задачах

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Число команд | Алгоритм | Время решения | Нарушения пожеланий по времени | Нарушения пожеланий по дням недели с числом повторений в 4 тура | Нарушения директивного срока | Нарушения одной игры для команды в неделю | Нарушения пожеланий по дням недели с долями в месяц | Нарушения игр между лидерами в зрелищные туры | Нарушения игр между лидерами в разных днях |
| 10 | Жадный | 1 | 25 | 30 | 0 | 21 | 76 | 52 | 21 |
| 10 | Локальный | 5818 | 14 | 28 | 0 | 9 | 71 | 52 | 19 |
| 20 | Жадный | 3 | 256 | 135 | 5 | 67 | 321 | 6 | 7 |
| 20 | Локальный | 29111 | 189 | 122 | 5 | 36 | 273 | 21 | 2 |
| 30 | Жадный | 3 | 417 | 131 | 36 | 59 | 615 | 0 | 7 |
| 30 | Локальный | 77089 | 299 | 100 | 39 | 19 | 503 | 0 | 2 |
| 40 | Жадный | 4 | 946 | 325 | 68 | 55 | 1175 | 220 | 104 |
| 40 | Локальный | 142000 | 666 | 260 | 67 | 15 | 1025 | 236 | 75 |
| 50 | Жадный | 6 | 1343 | 642 | 0 | 119 | 1801 | 352 | 1386 |
| 50 | Локальный | 247660 | 991 | 539 | 0 | 64 | 1557 | 352 | 1388 |

В соответствии со сводной таблицей результатов для категории задач, не соответствующих установленному формату оформления входных данных, все тесты пройдены. То есть, во всех случаях ПО «График» распознал исходные данные как некорректные.

Для оставшейся категории, которая содержит задачи с несовместными данными, результаты разнятся: 8 задач распознаны как некорректные, в остальных случаях (22 задачи) ПО «График» попытался решить задачу, следовательно, тесты не пройдены.

# 3 Выводы

По результатам эксперимента можно сделать следующие выводы:

1. Продвинутый алгоритм (алгоритм локального поиска) показал себя как алгоритм, находящий более точное решение по сравнению с жадным методом;
2. Модуль тестовой инфраструктуры «Генератор множества тестовых задач» требует доработки с целью повышения сложности генерируемых исходных данных в категории «Корректные задачи»;
3. ПО «График» корректно распознает задачи, не соответствующие установленному формату оформления входных данных;
4. ПО «График» требует доработки в части взаимодействия с несовместными данными.

По результатам эксперимента можно судить о правильной и корректной работе отдельных компонентов программы.

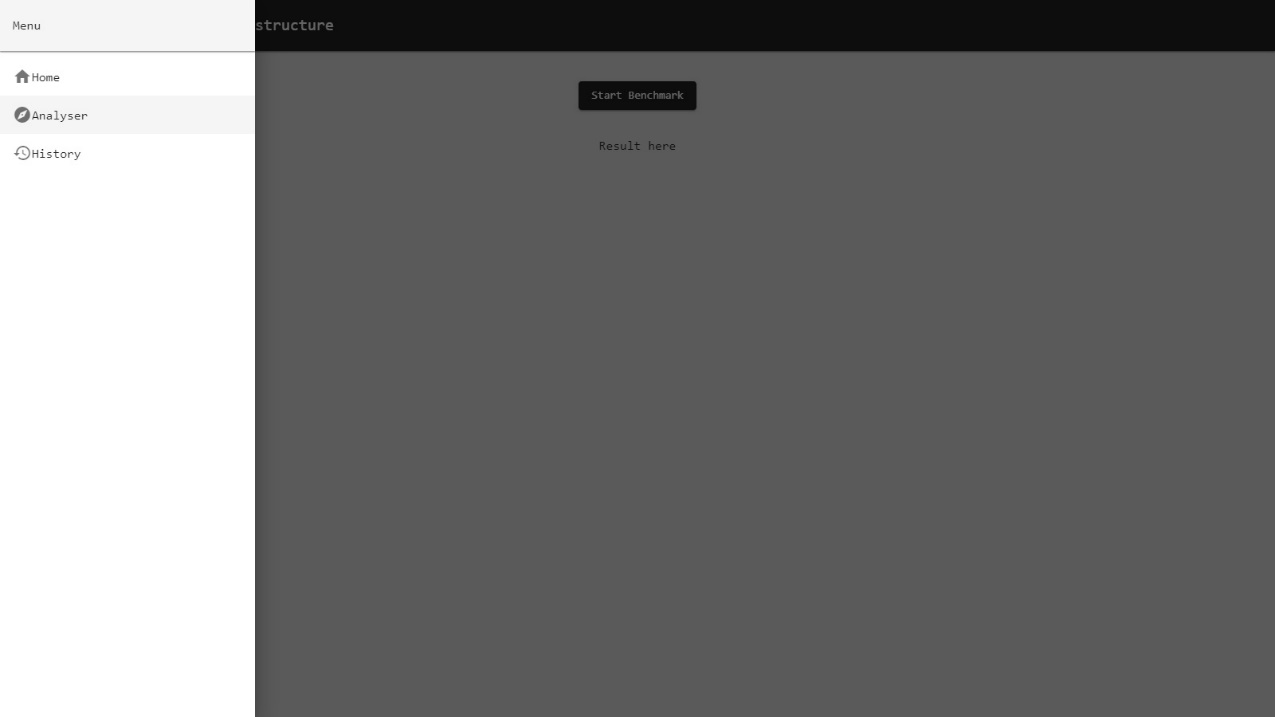


Рисунок 3. Меню

Далее в новом окне необходимо нажать на кнопку «Start Benchmark». Данное действие запустит тестовую инфраструктуру. После завершения её работы, на месте надписи «Result here» будет показан результат тестирования с разделенными по категориям задачами (Рисунок *4*).

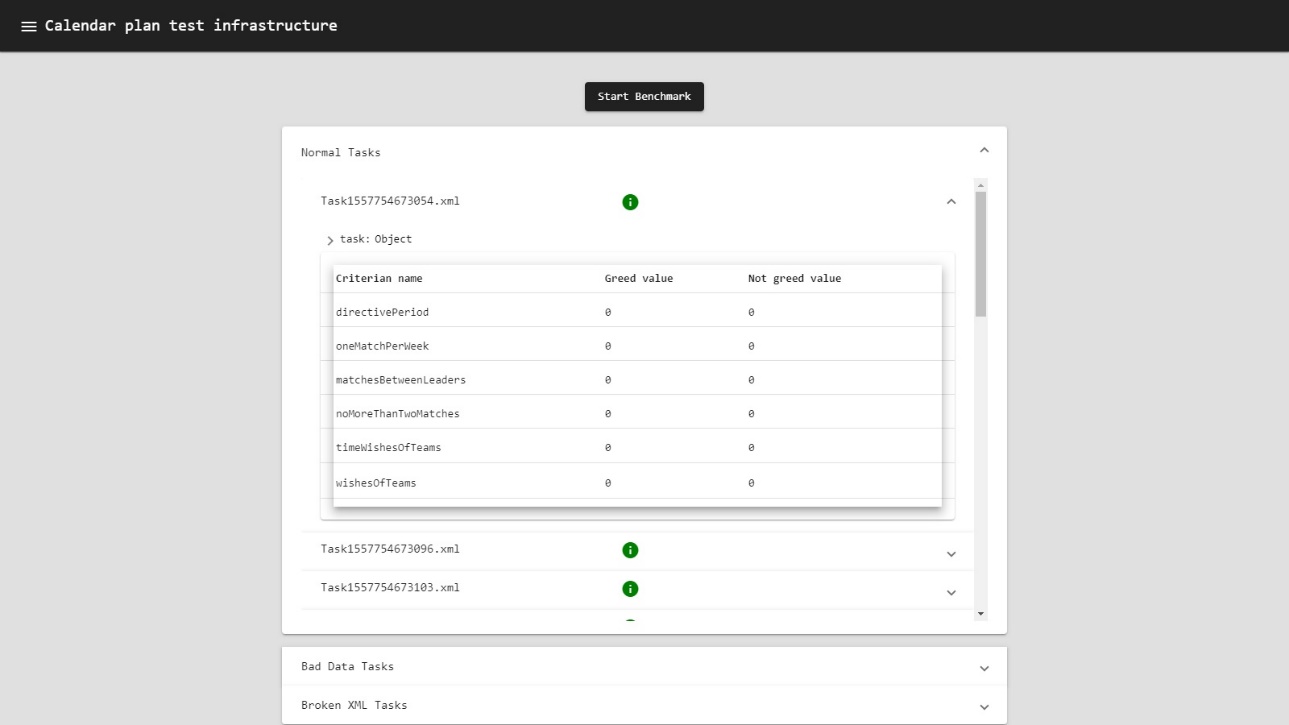


Рисунок 4. Результат работы программы

Кроме этого программа содержит раздел History – историю запусков тестов (Рисунок *5*). Перейдя в раздел History, можно заново запустить тесты без генерации новых задач. Для этого необходимо нажать на кнопку «Run Test» рядом с тестом (результат работы аналогичен «Start Benchmark»).

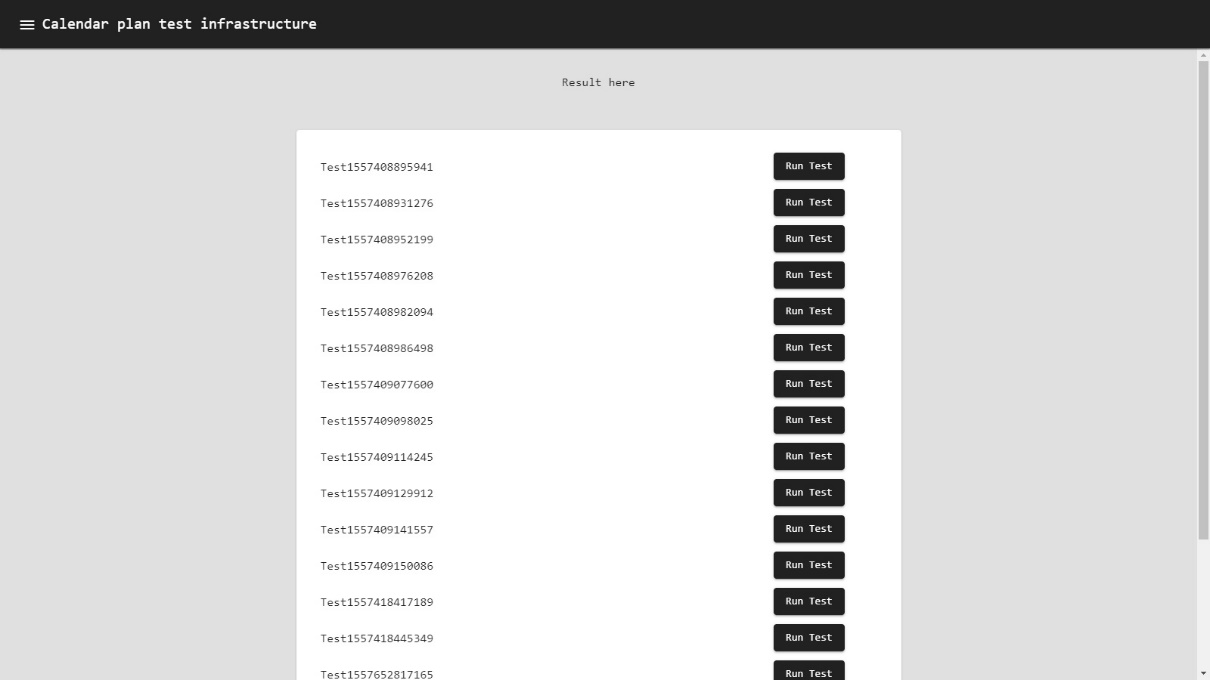


Рисунок 5. История запусков тестов

# ЗАКЛЮЧЕНИЕ

По результатам проведённых работ можно констатировать следующее:

Разработано два алгоритма составления графика спортивных мероприятий: Жадный и Локальный. Предложен программный интерфес для передачи информации для входных данных ввиде .xml документа и интерфейс для выходных данных в виде .html таблицы и лог файла log.txt. Создана инфраструктура автоматического тестирования алгоритмов составления расписаний. Результат тестирования показывает корректную работу алгоритмов, что отражено в статической информации выводимой после тестирования.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Техническое задание на выполнение работы «Разработка и реализация программного обеспечения планирования графика спортивных мероприятий», Шифр ПО«График»