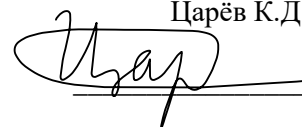


Правительство Российской Федерации  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет «Высшая школа экономики»  
Факультет компьютерных наук  
Образовательная программа бакалавриата 09.03.04 «Программная инженерия»

**ОТЧЕТ**  
**по учебной (технологической) практике**  
**в ООО "1С"**

Выполнил студент  
группы БПИ193  
Царёв К.Д.



**Проверили:**

**Руководитель практики от предприятия**

Старичков Никита Юрьевич

Заместитель директора по работе с научно-исследовательскими университетами, ООО  
"1С"

Дата 27.08.2021

\_\_\_\_\_9\_\_\_\_\_  
(оценка)



**Руководитель практики от факультета компьютерных наук**  
к.э.н., доцент Департамента программной инженерии

С.А. Лебедев

Дата \_\_\_\_\_

\_\_\_\_\_  
(оценка)

\_\_\_\_\_  
(подпись)

## **СОДЕРЖАНИЕ**

|   |          |
|---|----------|
| <b>АННОТАЦИЯ .....</b>  | <b>2</b> |
| <b>1. ЦЕЛЬ И ЗАДАЧИ ПРАКТИКИ.....</b>   | <b>3</b> |
| <b>2. ОПИСАНИЕ МЕСТА ПРОХОЖДЕНИЯ ПРАКТИКИ.....</b>  | <b>3</b> |
| <b>3. ОБЗОР ИЗУЧЕННЫХ ТЕХНОЛОГИЙ.....</b>   | <b>3</b> |
| <b>4. ОПИСАНИЕ АЛГОРИТМОВ, ТЕХНОЛОГИЙ, СРЕДСТВ РАЗРАБОТКИ, ИСПОЛЬЗОВАННЫХ ДЛЯ РЕШЕНИЯ ПОСТАВЛЕННЫХ ЗАДАЧ.....</b> | <b>4</b> |
| <b>5. ЗАКЛЮЧЕНИЕ .....</b>  | <b>6</b> |
| <b>6. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....</b>   | <b>7</b> |
| <b>7. РАБОЧИЙ ПЛАН-ГРАФИК ПРОХОЖДЕНИЯ ПРАКТИКИ.....</b>   | <b>8</b> |

## **АННОТАЦИЯ**

Данный отчет по учебной (технологической) практике в ООО "1С" содержит следующие разделы: «Цель и задачи практики», «Описание места прохождения практики», «Обзор изученных технологий», «Описание алгоритмов, технологий, средств разработки, использованных для решения поставленных задач», «Заключение», «Список использованных источников», «Рабочий план-график прохождения практики».

В разделе «Цель и задачи практики» указаны поставленная на период практики цель и задачи, которые необходимо было выполнить в ходе прохождения практики.

В разделе «Описание места прохождения практики» указаны виды деятельности и основные задачи ООО "1С".

В разделе «Обзор изученных технологий» описаны изученные алгоритмы, технологии, сервисы.

Раздел «Описание алгоритмов, технологий, средств разработки, использованных для решения поставленных задач» содержит подробное описание проделанной за период прохождения практики работы.

В разделе «Заключение» указаны полученные во время практики результаты и выводы.

Раздел «Список использованных источников» содержит оформленный по ГОСТу список источников.

В разделе «Рабочий план-график прохождения практики» содержится рабочий план-график прохождения практики с отметками о выполнении.

## **1. ЦЕЛЬ И ЗАДАЧИ ПРАКТИКИ**

### **Цель прохождения практики:**

Реализовать серверную часть веб-приложения по планированию отпусков для сотрудников.

### **Задачи практики:**

1. Изучить задачи и алгоритмы в теории расписаний;
2. Выбрать для реализации алгоритм планирования отпусков;
3. Реализовать серверную часть с базой данных сотрудников и методами доступа по REST Api;
4. Опубликовать серверную часть в облаке Microsoft Azure;
5. Предоставить доступ к Api сервера клиентской части приложения.

## **2. ОПИСАНИЕ МЕСТА ПРОХОЖДЕНИЯ ПРАКТИКИ**

Фирма "1С" специализируется на разработке, дистрибьюции, издании и поддержке компьютерных программ. Из разработок "1С" наиболее известна система программ "1С:Предприятие" – решения ERP-класса для повышения эффективности предприятий и учреждений, построенные на инновационной технологической платформе, которая обеспечивает высокую гибкость, масштабируемость и производительность корпоративных решений, работу в режиме "облачного" сервиса (SaaS) и на мобильных устройствах. "1С:Предприятие" успешно применяется для автоматизации управления и учета более чем в 1 500 000 организаций, включая крупнейшие корпорации и госструктуры. Система лидирует на рынке ERP России по количеству автоматизированных рабочих мест.

## **3. ОБЗОР ИЗУЧЕННЫХ ТЕХНОЛОГИЙ**

В ходе прохождения практики первым делом было начато изучение теории расписаний, а именно типовых задач и алгоритмов. В интернете было найдено множество научных статей, связанных с теорией расписаний, но все алгоритмы, описываемые в них, не подходили под конкретную цель практики – “планирование отпусков”, ведь в этих статьях описывались алгоритмы, связанные с выполнением работ на машинах. Это алгоритмы, решающие задачи из четырех основных классов:

1. Open shop, открытая линия — для каждого требования задано своё подмножество машин, на каждой из которых оно должно обслуживаться некоторое время. Порядок обслуживания на этих машинах произвольный.
2. Job shop, рабочий цех — для каждого требования задано своё упорядоченное подмножество машин (маршрут), на которых оно должно обслуживаться в заданном порядке.
3. Flow shop, потоковая линия — все машины упорядочены и каждое требование проходит все машины в этом порядке.
4. Задача с директивными сроками — для каждого требования задан момент поступления, время обслуживания и срок окончания обслуживания. Порядок обслуживания на приборах произвольный.

В итоге было обнаружено такое решение как “программирование в ограничениях” – это парадигма программирования, в которой отношения между переменными указаны в форме ограничений. Было найдено множество подключаемых библиотек, поддерживающих данную парадигму:

1. Google Optimization Tools (OR-Tools) (C++, Python, .Net, Java)
2. OptaPlanner (Java)
3. JaCoP (Java)
4. Microsoft Solver Foundation (.Net)

Все эти библиотеки являются программным обеспечением для комбинаторной оптимизации, которое находит наилучшее решение проблемы из очень большого набора возможных решений. В списке задач, решаемых данными библиотеками, присутствует такая как “Планирование: Поиск оптимального расписания для сложного набора задач.”. Это именно то, что нужно.

Далее было начато изучение платформы ASP.Net с использованием Entity Framework. ASP.Net – это платформа разработки веб-приложений. Она входит в состав платформы .Net Framework, поэтому разработку можно вести на языке программирования C#. Entity Framework – это объектно-ориентированная технология доступа к данным. Данная платформа была выбрана, потому что на первом курсе обучения на дисциплине “Программирование” изучался язык C#, и следовательно, уже имеется некоторый опыт работы с ним.

#### **4. ОПИСАНИЕ АЛГОРИТМОВ, ТЕХНОЛОГИЙ, СРЕДСТВ РАЗРАБОТКИ, ИСПОЛЬЗОВАННЫХ ДЛЯ РЕШЕНИЯ ПОСТАВЛЕННЫХ ЗАДАЧ**

После изучения материалов, описанных в предыдущем разделе, необходимо было выбрать библиотеку, поддерживающую парадигму программирования в ограничениях. Так как всё web-API необходимо было написать на ASP.Net, которое входит в состав платформы .Net, то и библиотеку нужно было подбирать так, чтобы она поддерживалась данной платформой. Из обнаруженных библиотек подходили две (“Google OR-Tools” и “Microsoft Solver Foundation”). Выбор пал на библиотеку от Google, так как документация к ней более понятна, а также в документации присутствует множество примеров кода, что делает изучение новой технологии проще.

Далее началось осваивание “программирования в ограничениях”, и написание основного алгоритма планирования отпусков. Основная проблема, которую необходимо было решить придуманным алгоритмом – это то, что в каждый день в отпуске может находиться не более половины сотрудников одной должности. Из данной проблемы сразу выливается первое ограничение входных данных – должно быть как минимум два взаимозаменяемых сотрудника. Иначе алгоритм просто не сможет подобрать даты отпусков.

##### **Алгоритм планирования отпусков**

Суть программирования в ограничениях заключается в том, что мы создаем какую-то модель и накладываем на нее некоторые ограничения, а уже в соответствии с этими ограничениями, нам автоматически выдается решение.

В случае планирования отпусков моделью является двумерная матрица, где по горизонтали указаны дни, а по вертикали – взаимозаменяемые сотрудники. Изначально матрица заполнена нулями, а те дни, в которые определенный сотрудник будет находиться в отпуске, будут отмечаться единицами. Таким образом, накладываемые ограничения становятся очевидными: 1) в каждом столбце единиц может быть не больше, чем количество сотрудников, деленное на два; 2) в каждой строке единиц может быть не меньше 14 и не больше 28. Ниже приведен условный пример заполнения матрицы (0 заменены на “-”, 1 заменены на “+” для лучшего восприятия).

Библиотека Google OR-Tools позволяет оптимизировать решение и находить лучшее, но так как в случае планирования отпусков это не требуется, то для сокращения времени работы программы, выдается первое найденное.

Так как в матрице числа по горизонтали находятся в диапазоне от 0 до 364, то затем они преобразовываются в формат даты и записываются каждому сотруднику.

После написания основного алгоритма настало время написания web-API с моделью сотрудников и методами доступа по Rest Api.

### Модель сотрудников

Так как данное приложение является лишь прототипом, то в модели сотрудников присутствует минимальный набор полей, необходимый для тестирования: ID сотрудника, фамилия, имя, отчество, дата рождения, пол, должность, даты отпуска. ID сотрудника необходимо для создания базы данных, оно представляет из себя целое число, автоматически присваиваемое сотруднику, при добавлении его в БД. Фамилия, имя и отчество представляют из себя строки, ограниченные по длине (от 2 до 40 символов), и по наличию в них только букв. Дата рождения представлена в формате даты, и ограничена возрастом сотрудника (от 16 до 99 лет). Возраст вычисляется исходя из даты рождения и текущей даты. Даты отпусков представлены в строковом формате. Пол представлен целым числом (0 – мужской, 1 – женский). Должность также представлена числом для простоты обращения с ней. Ниже представлены коды должностей и их названия (взаимозаменяемые должности отмечены одинаковым цветом).

1. Директор
2. Заместитель директора
3. Начальник IT-отдела
4. Заместитель начальника IT-отдела
5. Главный бухгалтер
6. Бухгалтер
7. C# разработчик
8. Java разработчик
9. C++ разработчик
10. Дизайнер

11. Сотрудник HR
12. Системный администратор

Все вводимые данные валидируются атрибутами, так что некорректные данные ввести не получится. Для валидации даты рождения был написан собственный атрибут, вычисляющий возраст, так как готового решения в ASP.Net нет.

### Методы доступа по REST Api

Ниже представлена таблица, в которой указаны все методы доступа.

| Методы            |          |                        |   |
|-------------------|----------|------------------------|---|
| Имя               | HTTP тип | Пример URL-адреса      | Назначение                                |
| GenerateVacations | GET      | api/Employees/generate | Генерация отпусков.                       |
| GetEmployees      | GET      | api/Employees          | Возвращает всех сотрудников.              |
| GetEmployee       | GET      | api/Employees/1        | Возвращает конкретного сотрудника по ID.  |
| PutEmployee       | PUT      | api/Employees/1        | Обновление данных конкретного сотрудника. |
| PostEmployee      | POST     | api/Employees          | Добавление нового сотрудника.             |
| DeleteEmployee    | DELETE   | api/Employees/1        | Удаление сотрудника.                      |
| DeleteEmployees   | DELETE   | api/Employees          | Удаление всех сотрудников.                |

В методе GenerateVacations все сотрудники разделяются на группы по взаимозаменяемости, и затем, описанным выше алгоритмом, планируются отпуска по отдельности для каждой группы.

Всё было готово для подключения серверной части к клиентской, но к моменту завершения работ фронтенд не был готов для интеграции. Поэтому 8-я задача из плана графика (Подключение серверной части к клиентской и устранение возникших проблем, связанных с подключением) снялась. И, соответственно, 7-я (Публикация серверной части приложения в облаке Microsoft Azure) тоже. Так как не было смысла публиковать бэкенд в облаке и оплачивать эту услугу без готовой клиентской части.

## 5. ЗАКЛЮЧЕНИЕ

По окончании практики была реализована рабочая серверная часть веб-приложения по планированию отпусков, были получены теоретические знания в области теории расписаний и практические навыки разработки бэкенда. В ходе практики также были изучены библиотека Google OR-Tools и принципы работы в ASP.Net с использованием Entity Framework.

## 6. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Задачи теории расписаний. Методы решения: комбинаторный подход, эвристический и комбинаторный методы [Электронный ресурс] // Студопедия. [2021—2022]. Дата обновления: 24.02.2014. URL: [https://studopedia.ru/3\\_200683\\_zadachi-teorii-raspisaniy-metodi-resheniya-kombinatorniy-podhod-evristicheskiy-i-kombinatorniy-metodi.html](https://studopedia.ru/3_200683_zadachi-teorii-raspisaniy-metodi-resheniya-kombinatorniy-podhod-evristicheskiy-i-kombinatorniy-metodi.html) (дата обращения: 01.07.2021)
- 2) Задачи планирования и программирование в ограничениях [Электронный ресурс] // Хабр. [2021—2022]. Дата обновления: 22.11.2017. URL: <https://habr.com/ru/company/billing/blog/342550/> (дата обращения: 02.07.2021)
- 3) About OR-Tools [Электронный ресурс] // Google OR-Tools. [2021—2022]. Дата обновления: 17.06.2021. URL: <https://developers.google.com/optimization/introduction/overview> (дата обращения: 03.07.2021)
- 4) Microsoft Solver Foundation 3.1 [Электронный ресурс] // Microsoft. [2021—2022]. Дата обновления: 12.06.2011. URL: [https://docs.microsoft.com/en-us/previous-versions/visualstudio/ff524509\(v=vs.93\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/visualstudio/ff524509(v=vs.93)?redirectedfrom=MSDN) (дата обращения: 03.07.2021)
- 5) Клиент-сервер [Электронный ресурс] // Википедия. [2021—2022]. Дата обновления: 21.10.2017. URL: [https://ru.wikipedia.org/wiki/Клиент\\_—\\_сервер](https://ru.wikipedia.org/wiki/Клиент_—_сервер) (дата обращения: 05.07.2021)
- 6) Руководство по ASP.NET Core 5 [Электронный ресурс] // metanit. [2021—2022]. Дата обновления: 15.12.2020. URL: <https://metanit.com/sharp/aspnet5> (дата обращения: 06.07.2021).

## 7. РАБОЧИЙ ПЛАН-ГРАФИК ПРОХОЖДЕНИЯ ПРАКТИКИ

| № п/п | Сроки проведения        | Планируемые работы  | Отметка о выполнении |
|-------|-------------------------|---|----------------------|
| 1     | 01.07.2021              | Инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка | ВЫПОЛНЕНО            |
| 2     | 01.07.2021 – 03.07.2021 | Изучение типовых задач и алгоритмов в теории расписаний   | ВЫПОЛНЕНО            |
| 3     | 05.07.2021 – 09.07.2021 | Изучение основ создания серверной части веб-приложений на платформе ASP.Net Core с использованием Entity Framework                                      | ВЫПОЛНЕНО            |
| 4     | 10.07.2021              | Консультация с руководителем практики по выбору оптимального алгоритма планирования отпусков  | ВЫПОЛНЕНО            |
| 5     | 12.07.2021 – 15.07.2021 | Создание серверной части веб-приложения   | ВЫПОЛНЕНО            |
| 6     | 16.07.2021              | Консультация с руководителем практики по промежуточным результатам работы   | ВЫПОЛНЕНО            |
| 7     | 16.07.2021              | Публикация серверной части приложения в облаке Microsoft Azure  | НЕ<br>ВЫПОЛНЕНО      |
| 8     | 17.07.2021 – 21.07.2021 | Подключение серверной части к клиентской и устранение возникших проблем, связанных с подключением   | НЕ<br>ВЫПОЛНЕНО      |