



**Министерство образования и науки Российской Федерации**  
**Федеральное государственное бюджетное образовательное**  
**учреждение**  
**высшего образования**  
**«Московский государственный технический университет**  
**имени Н.Э. Баумана**  
**(национальный исследовательский университет)»**  
**(МГТУ им. Н.Э. Баумана)**

---

**Факультет**

«Информатика, искусственный интеллект и системы управления»

**Кафедра**

«Программное обеспечение ЭВМ и информационные технологии» (ИУ7)

**Отчет по лабораторной работе**

«Генерация отчета»

Выполнил: Великанов К.Ю., гр. ИУ7-13М

Москва 2022

## Цель работы

Целью работы является разработка программы, которая создаёт html файл, содержащий характеристику на ЕЯ на момент вводимой пользователем даты на каждого сотрудника фонда из приложенного xls файла в формате

*ФИО|пол|должность|дата рождения|дата приёма на работу|дата увольнения*

При этом об одном сотруднике может быть несколько записей.

Предложения должны быть согласованы в роде, числе и падеже.

Формат даты должны использоваться согласно выданному варианту.

Пример:

*Характеристика.*

*Иванов Иван Иванович родится 25 апреля 2050 года.*

*Первого 04 в 2035 году устроится на работу в Фонд на должность старшего помощника младшего ассенизатора.*

*Уволится в 2068 в связи с переходом на должность директора.*

*Окончательно уволится 31 декабря 2077.*

*Дата создания документа 19.09.2022.*

*Вариант: Двенадцатое марта 2022.*

## Ход работы

В ходе разработки программы будет использоваться язык Dart.

Перед началом работы программа ожидает на вход дату, которая будет использоваться при выборе характеристик.

Для считывания таблицы будет использоваться пакет excel [1]. При обработке excel книги функции из пакета проходят по всем заполненным строкам и по всем заполненным столбцам.

Каждая запись записывается в специальный объект класса Person. Его реализация представлена на Листинге 1.

Листинг 1

```
class Person {
  final String name;
  final String gender;
  final String position;
  final DateTime birthDate;
  final DateTime hireDate;
  final DateTime resignationDate;
  final List<PositionChange> positionChanges;

  const Person(this.name, this.gender, this.position, this.birthDate,
    this.hireDate, this.resignationDate, this.positionChanges);

  factory Person.fromXlsRow(Map<String, dynamic> row) {
    try {
      final name = row["ФИО"];
      final gender = row["пол"];
      final position = row["должность"];
      final birthDate = parseDate(row["дата рождения"]);
      final hireDate = parseDate(row["дата приёма на работу"]);
      final resignationDate = parseDate(row["дата увольнения"]);
      final positionChanges = <PositionChange>[];

      // Создаем список изменений должностей

      return Person(name, gender, position, birthDate, hireDate,
        resignationDate, positionChanges);
    } catch (e, s) {
      print('failed to parse person from xls row: $row');
```

```

        print(e);
        print(s);
        exit(70);
    }
}

@override
int get hashCode => name.hashCode + gender.hashCode +
birthDate.hashCode;

@override
operator ==(Object other) =>
    other is Person &&
    hashCode == other.hashCode &&
    name == other.name &&
    gender == other.gender &&
    birthDate == other.birthDate;

@override
String toString() {
    return "[$name, $gender, $birthDate]";
}
}

```

Каждая запись из строки записывается в этот класс. Если в таблице сотрудник встречается больше одного раза, то его переходы записываются в массив объектов класса `PositionChange`. Одним и тем же сотрудником считается, если его ФИО, пол и дата рождения полностью совпадают. Реализация класса `PositionChange` представлена на Листинге 2.

Листинг 2

```

class PositionChange {
    final DateTime startDate;
    final DateTime? endDate;
    final String position;

    const PositionChange(this.startDate, this.endDate, this.position);
    factory PositionChange.fromXlsRow(Map<String, dynamic> row) {
        final startDate =

```

```

        DateFormat("dd.MM.yyyy").parse(row["дата приёма на работу"]);
final endDate = row["дата увольнения"] != null
    ? DateFormat("dd.MM.yyyy").parse(row["дата увольнения"])
    : null;
final position = row["должность"];

return PositionChange(startDate, endDate, position);
}
@override
String toString() {
    return "$position [$startDate - $endDate]";
}
}

```

Программа заносит память информацию о каждой записи в таблицу. При формировании характеристик, учитываются только те события, что произошли до даты, указанной при начале работы с программой.

Так, например, если в программу ввести дату “Двенадцатое марта 2022”, а дата рождения сотрудника “30.12.2040”, то характеристика на такого сотрудника не будет составлена, так как он еще не родился.

Также в характеристику не будут добавлены сведения о принятиях и увольнениях, если они еще не произошли в данный момент времени.

Для формирования файла используется встроенная библиотека dart:io, которая сохранит файл в файловой системе.

Для форматирования дат и остальной локализации используется пакет intl [2].

## **ЗАКЛЮЧЕНИЕ**

В ходе лабораторной работы на языке Dart была разработанная программа генерации характеристик из приложенного excel файла.

## **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. excel | Dart Package [Электронный ресурс] (URL: <https://pub.dev/packages/excel>) Дата обращения: 25.12.2022
2. intl | Dart Package [Электронный ресурс] (URL: <https://pub.dev/packages/intl>) Дата обращения: 25.12.2022