|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №1**

Выполнил студент группы ИНБО-02-20 Абдусалямова С. К.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

Москва 2021г.

## **Задание**

По диаграмме класса UML, описывающей сущность «Автор», написать программу, которая состоит из двух классов: «Author» и «TestAuthor». Класс «Author» должен содержать реализацию методов, представленных на диаграмме.

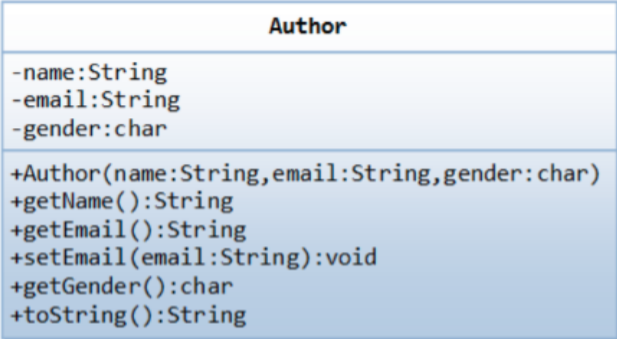


Рисунок 1 – UML-диаграмма

Класс Author cодержит:

* три private переменных-полей данных класса: name (типа String) , email (типа String), и gender (типа char, которая может принимать три значения либо 'M', если автор книги мужчина, 'F' – если автор книги женщина, или 'U', если пол автора неизвестен; вы можете также использовать для реализации логическую переменную под названием male для обозначения пола автора, которая будет принимать значение истина или ложь).
* Один конструктор для инициализации переменых name, email и gender с заданными значениями. (Тут не будет конструктора по умолчанию так как нет значений по умолчанию ни для имени, ни для электронной почты или пола).
* Public методы Геттеры/сеттеры: getName(), getEmail(), setEmail(), and getGender(). (Нужно упомянуть, что там не будет сеттеров для имени и пола, так как эти атрибуты не могут изменяться).
* Метод ToString (), которая должен возвращать следующий текст "автор-имя (пол) на адрес электронной почты», то есть в строке должно быть записано: имя[пробел](пол)[пробел]at[пробел]емайл.

## **Ход Работы**

На основе поставленной задачи был написан код программы:

**Author.java**

package com.company;  
  
public class Author {  
 private String name;  
 private String email;  
 private char gender;  
  
 public Author(String name, String email, char gender) {  
 this.name = name;  
 this.email = email;  
 this.gender = gender;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public char getGender() {  
 return gender;  
 }  
  
 @Override  
 public String toString() {  
 return this.name + " (" + this.gender +") at "+ this.email;  
 }  
}

**Main.java**

package com.company;  
  
public class Main {  
  
 public static void main(String[] args) {  
 Author author = new Author("Peter Parker", "PeterParker@gmail.com", 'm');  
 System.out.println(author.getName());  
 System.out.println(author.getEmail());  
 System.out.println(author.getGender());  
  
 author.setEmail("spiderman@gmail.com");  
 System.out.println(author.toString());  
 }  
}

## **ВЫВОД**

В ходе данной работы я ознакомилась с UML нотацией, а также закрепила навыки создания классов, полей, описания методов.