

```

public class PersonApp {
    public static void main(String[] args) {
        Person[] persons = new Person[5];
        Person person1 = new Person("Bob", 25, 'm');
        Person person2 = new Person("Alica", 30, 'w');
        Person person3 = new Person("Cris", 22, 'm');
        Person person4 = new Person("Natalie", 19, 'w');
        Person person5 = new Person("Anna", 34, 'w');

        persons[0] = person1;
        persons[1] = person2;
        persons[2] = person3;
        persons[3] = person4;
        persons[4] = person5;

        System.out.println("Female number = " + femaleNum(persons));
        System.out.println("Aged over 25 number = " + under25Num(persons));
        System.out.println("Average male age = " + averageMaleAge(persons));
        System.out.println("Average people age = " + averageAllAge(persons));

    }

    /**
     * count the number of females in the array
     * @param persons
     * @return the count
     */
    public static int femaleNum(Person[] persons) {
        int count = 0;
        for (Person p : persons) {
            if (p.getGender() == 'w') {
                count++;
            }
        }
        return count;
    }

    /**
     * count the number of people aged over 25
     * @param persons
     * @return the count
     */
    public static int under25Num(Person[] persons) {
        int count = 0;
        for (Person p : persons) {
            if (p.getAge() > 25) {
                count++;
            }
        }
        return count;
    }
}

```

```

    }

    /**
     * find the average age of all males
     * @param persons
     * @return the count
     */
    public static int averageMaleAge(Person[] persons) {
        int count = 0, sum = 0;
        for (Person p : persons) {
            if (p.getGender() == 'm') {
                count++;
                sum += p.getAge();
            }
        }
        return sum / count;
    }

    /**
     * find the average age of all people
     * @param persons
     * @return the count
     */
    public static int averageAllAge(Person[] persons) {
        int sum = 0;
        for (Person p : persons) {
            sum += p.getAge();
        }
        int count = persons.length;
        return sum / count;
    }

}

```

```

/**
 * Person class
 */
public class Person {
    private String name;
    private int age;
    private char gender;

    /**
     * Default constructor
     */
    public Person() {
        name = "";
    }
}

```

```

        age = 0;
        gender ='m';
    }

    /**
     * Constructor
     * @param name
     * @param age
     * @param gender
     */
    public Person(String name, int age, char gender) {
        this.name = name;
        this.age = age;
        this.gender = gender;
    }

    /**
     * Fields getters and setters
     */
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public char getGender() {
        return gender;
    }

    public void setGender(char gender) {
        this.gender = gender;
    }

    @Override
    public String toString() {
        return "Name: " + getName() + " age: " + getAge() + " gender: " + getGender();
    }
}

```