Для всех задач необходимо дополнительно создавать класс Application, который будет содержать статический метод main.

**Задача 1**

Необходимо создать два класса – Calculator и Application. В классе калькулятора реализовать 4 метода – сложение, вычитание, умножение, деление. В методе main создать экземпляр данного класса и вызвать все методы. Результаты выполнения методов вывести на экран.

|  |  |
| --- | --- |
| Входные данные | Ожидаемый результат |
| calculator.summ(4, 2) | 6 |
| calculator.substract(5, 3) | 2 |
| calculator.multiply(2, 6) | 12 |
| calculator.divide(5, 3) | 1 |

**Задача 2**

Создать класс Cat с тремя полями: int age, int weight, String name. В классе должен быть конструктор, принимающий все переменные в качестве аргументов и устанавливающий значения полей объекта переданными данными. В методе main создать массив из 5 элементов, каждый из которых является экземпляром класса Cat. Отсортировать созданный массив по полю name. Вывести отсортированный массив в формате

Имя: <name>, Возраст: <age>, Вес: <weight>

каждый элемент – с новой строки

|  |  |
| --- | --- |
| Входные данные | Ожидаемый результат |
| [ new Cat(2, 5, “Барсик”),  new Cat(3, 11, “Мурзик”),  new Cat(7, 2, “Матильда”),  new Cat(1, 3, “Леопольд”),  new Cat(11, 4, “Базилио”) ] | Имя: Базилио, Возраст: 11, Вес: 4  Имя: Барсик, Возраст: 2, Вес: 5  Имя: Леопольд, Возраст: 1, Вес: 3  Имя: Матильда, Возраст: 7, Вес: 2  Имя: Мурзик, Возраст: 3, Вес: 11 |

**Задача 3**

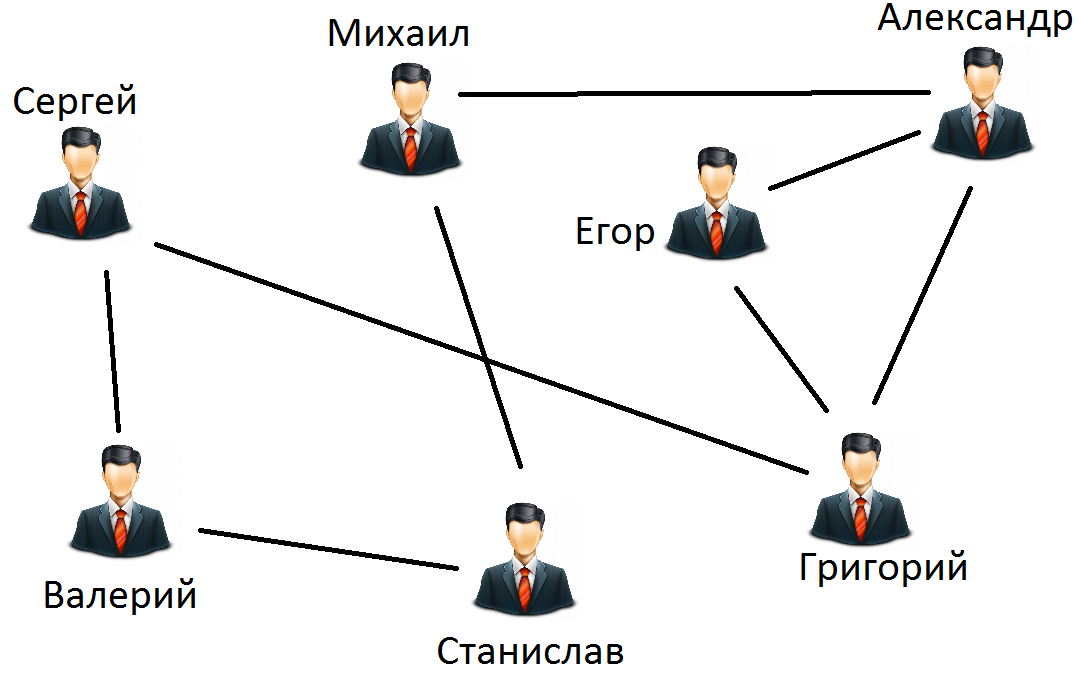
Необходимо создать класс Human, описывающий человека. У каждого человека есть имя, возраст, родители. В классе Human реализовать конструктор, в который передается имя, возраст, отец (null, если отсутствует), мать (null, если отсутствует)

В методе main создать семью, состоящую из 8 человек: супруги Сергей и Марина. У них есть общие дети – сын Виталий и дочь Елена. У Сергея родителей зовут Валерий и Александра. У Марины есть только отец - Иван. Возраст выбрать произвольно. После создания семьи необходимо создать массив, содержащий всех бабушек и дедушек Елены и вывести каждый из элементов массива на экран в произвольном формате.

**Задача 4**

Необходимо создать класс Human, описывающий человека. У каждого человека есть имя и друзья. В конструкторе класса Human инициализируется только имя, массив друзей должен быть пустым. Реализовать в классе Human метод, позволяющий добавить в друзья другого человека. Данный метод автоматически должен добавлять другому человеку в друзья исходного (должна отсутствовать возможность «односторонней» дружбы)

В классе Application необходимо реализовать статический метод, возвращающий логическую переменную в зависимости от того, являются друзьями два человека или нет. В методе main добавить дружеские связи согласно рисунку и проверить, что созданный статический метод работает правильно.



|  |  |
| --- | --- |
| Входные данные | Ожидаемый результат |
| Сергей, Григорий | true |
| Сергей, Александр | false |
| Станислав, Михаил | true |
| Михаил, Валерий | false |

**Задача 5**

Дано: Есть случайное число людей. У каждого из людей может быть несколько животных – кошек или собак. Количество кошек и собак у каждого человека также случайно (от 0 до 3).

Задание:

1. Реализовать класс Names, содержащий несколько статических переменных – массивов строк: Имена людей, имена собак, имена кошек. Имена выбрать произвольные. Реализовать статические методы, каждый из которых возвращает случайное имя человека, собаки или кошки соответственно
2. Реализовать класс Cat и класс Dog, описывающий кошек и собак – кличку и возраст. В каждом из этих классов описать конструктор без параметров, который создает животное со случайным именем из массива в классе Names и случайным возрастом.
3. Реализовать класс Human, который содержит имя человека, а также информацию о его животных. Добавить конструктор без параметров, который добавляет случайное количество животных согласно «Дано», а также присваивает случайное имя из класса Names.
4. В методе main создать массив типа Human из 20 элементов, с помощью цикла сгенерировать случайных людей. Вывести на экран тех людей, у которых совпадает одновременно количество кошек и собак.