

1. Создать структуру пользователя **User** со свойствами:

- **name**
- **surname**
- **age**
- **salary**

Добавить методы:

- **promotion**. Аргументом необходимо передавать новый размер зарплаты и проверять, что если сумма больше предыдущей, то изменять зарплату, иначе выводить ошибку в консоль и не менять размер зарплаты.
- **birthday**. Метод не должен ничего принимать, а должен увеличивать возраст пользователя на 1.
- **printInfo**. Выводить в консоль информацию о пользователе в читаемом виде.

Создать экземпляр пользователя, повысить его пару раз, провести пару дней рождения и вывести информацию в консоль.

2. Создать структуру **Factory** со свойствами:

- **workers**. Массив работников типа **User**, описанного в первом задании.
- **name**
- **director**. Тоже типа **User**.

Реализовать метод:

- **printInfo**. Метод, подобный методу из заданий по перечислениям. Данный метод должен принимать только один аргумент: режим вывода (один из вариантов перечисления **full**, **middle**, **short**) и выводить в консоль информацию о всех работниках и директору завода согласно указанному режиму.
- **newDirector**. Метод не должен принимать никаких параметров. Нужно выбирать из работников того, который получает больше всех и назначать его директором завода (перезаписывать свойство директор), соответственно удаляя из списка рабочих этого человека. Если работников на заводе не осталось, нужно вывести ошибку, что директора назначить невозможно.
- **hire**. Метод должен принимать нового работника типа **User** и добавлять в массив работников. *\*Дополнительно: после устройства на завод работник должен получать ровно в два раза больше, чем получал до этого.*

Создать работников, директора и на основании их создать завод. Нанять на завод пару работников и сменить директора. После чего вывести информацию о всех работниках и директору в любом режиме вывода.