

Проект 1

Вариант 1 за всички с четни факултетни номера

1. Да се реализира клас Person (човек) с член данни:

- Име (name) – низ с произволна дължина
- ЕГН (EGN) – низ с дължина 10 символа
- Адрес (address) – низ с произволна дължина
- Години (age) – естествено число
- Пол (sex) – да се реализира подходящ изброим тип
- Професия (profession) – да се реализира подходящ изброим тип от професии (Professions)
- Годишен доход (income) – реално неотрицателно число

1.1. Да се реализират за класа Person:

- Конструктор с параметри за име, ЕГН, адрес, професия и годишен доход. Параметрите за адрес професия и годишен доход да имат стойности по подразбиране съответно празен низ, безработен и Олв. годишен доход. Член-данната години да се изчислява спрямо подаденото ЕГН и годината 2013г. Член-данната пол да се определя спрямо подаденото ЕГН. /Полът се определя по предпоследната цифра от ЕГН-то, ако тя е четно число полът е мъжки, ако е нечетно число женски/.
- Копиращ конструктор
- Оператор =

Копиращият конструктор и оператор равно не са много логични, не е етично да се клонират хора, но това да се пренебрегне)

- Деструктор
- Селектори
- Мутатори

Да не се реализират мутатори за член-данните ЕГН, години и пол.

1.2. Да се реализира метод Information на класа Person, който извежда на стандартния изход информация за човека.

1.3. Да се предефинира оператора за изход <<, който да извежда в поток информация за човека.

2. Да се реализира клас Community (общност), която представлява общност от хора. Нега общността да има следните член данни:

- Име на общността (communityName) – низ с произволна дължина
- Дата на основаване на общността (foundationDate) – низ във формат дд.мм.гггг
- Основател и глава на общността (founder) – указател към човек

- Максимален брой членове (maxMembersCount) – естествено число
- Брой членове (membersCount) – естествено число
- Членове на общността (members) – динамичен масив от хора
- Брой общности – статична целочислена член данна, която съдържа броя съществуващи общности.

2.1. Да се реализират за класа Community:

- Конструктор с параметри за член-данните име на общност, дата на основаване на общността, основател на общността и максимален брой членове със стойност по подразбиране 2. Масиват от членове на общността по подразбиране на бъде празен.
- Копиращ конструктор
- Оператор =
- Деструктор
- Селектори
- Мутатори

Да не се реализират мутатори за член-данните дата на основаване на общността и брой членове на общността.

2.2. Да се дефинират следните методи за класа Community:

- Булев метод isFull, който проверява дали има свободни места в общността.
- Метод addMember, който по подаден човек да го добавя в общността. В общността даден човек не може да го има повече от веднъж. Ако в общността няма места, динамичният масив от членове на общността трябва да се resize-не двойно. Методът да връща булева стойност спрямо това дали е успял да добави новия член на общността или не.
- Метод removeMember, който по подадено ЕГН премахва член на общността с това ЕГН, ако съществува такъв. Методът да връща булева стойност спрямо това дали е успял да премахне въпросния член на общността. Ако броя на членовете на общността падне под 25% от максималния брой членове на общността, динамичния масив от членовете на общността да се resize-не наполовина.
- Булев метод isMember, който по подаден човек или ЕГН, проверява дали има такъв член на общността. Да се реализират два отделни метода за двата вида подавани параметри.
- Метод information, който извежда на стандартния изход информация за всичките членове на общността.

2.3. Да се предефинират следните оператори за класа Community:

- Оператор +=, който добавя член към обществото.
- Оператор +, който намира обединението на членовете на две общества. В резултатът да няма повтарящи се хора.
- Оператор -=, който премахва член на обществото по подаден човек или ЕГН, ако той съществува.
- Оператор -, който намира разликата на членовете на две общества.

- Оператор $/$, който намира сечението на членовете на две общества.
- Оператор $*$, който намира допълнението на членовете на една общност до членовете на друга подадена общност.

Операторите $+$, $-$, $/$, $*$, които намират обединение, разлика, сечение, допълнение на общности да връщат масив от хора.

- Оператор $==$, който проверява дали две общности се състоят от едни и същи членове.
- Оператор $<$, който проверява дали членовете на едната общност са членове и на другата (подмножество).
- Оператор за изход $<<$, който извежда в поток информация за всичките членове на общността.

Един човек може да бъде едновременно в повече от една общности.

3. Да се напише функция `writelnFile`, която получава като параметри общество и име на файл. Функцията да запише информацията за обществото заедно със всичките й членове в текстов файл с подаденото име.

Бонус за поддържане на членовете на всяка общност в сортиран ред по тяхното ЕГН и реализирани ефективни алгоритми за добавяне, премахване и търсене на член на общността.

Приятна работа