

Вариант 1.

Реализирайте клас **Потребител**, който има следните характеристики:

- email - символен низ с произволна дължина, който съдържа точно един символ @ и играе роля на потребителско име;
- парола - непразен символен низ с произволна дължина.

Реализирайте методи за коректен жизнен цикъл, както и достъп до данните. Не се очаква промяна на потребителското име.

Предефинирайте оператори за вход и изход в поток и сравнение за равенство. Два потребителя са равни, ако имат еднакви имейли.

Реализирайте клас **Администратор**, който е потребител с допълнителни свойства като години стаж (цяло неотрицателно число) и ниво на компетентност (цяло число между 1 и 5).

Реализирайте необходимите методи за коректен жизнен цикъл, както и оператори за сравнение (наредба). По-голям е администраторът с по-високо ниво на компетентност. В случай на равна компетентност, по-голям е този с повече години трудов стаж.

Реализирайте методи за работа с тези характеристики, както и оператори ++ и -- , които увеличават / намаляват компетентността на администратор с единица.

Предефинирайте оператори за вход и изход в поток.

Реализирайте клас **СуперПотребител**, който е администратор, но може да създава потребители, чрез предефиниран функционален оператор с подходящи параметри. Трябва да съхранява потребителските имена на всички създадени от него потребители. Добавете оператор за индексване, който получава като аргумент потребителско име, и проверява дали потребителят с това име е бил създаден от съответния **СуперПотребител** или не. Предефинирайте оператори за вход и изход в поток.

Реализирайте клас **Система**, която се характеризира с определено име (произволен символен низ). Системата притежава произволен брой регистрирани в нея потребители и произволен брой администратори. Тя може да има или да няма **СуперПотребител**. Ако такъв има, той е единствен.

Могат да се създават нови системи, копия на съществуващи, но информацията от вече съществуваща система не може да бъде копирана във друга вече съществуваща система.

Реализирайте методи за коректен жизнен цикъл на системата, както и за достъп до данните. Реализирайте също следните оператори:

- Оператор за събиране, който да добавя потребител в системата. Това е възможно, ако в системата няма потребител със същото име.
- Функция, с която се добавя администратор. Той не трябва да е бил нито потребител, нито администратор в тази система.
- Оператор за изваждане, който премахва потребител или администратор от системата. Премахването е възможно, ако поисканият потребител е в системата, но не е последният администратор.
- Оператор \wedge , който по подадено число – минимално ниво на компетентност, връща указател към най-малкия от администраторите, който е с поне такова ниво на компетентност, ако има такъв, или nullptr, ако няма.

- Оператори за вход и изход от поток.

Демонстрирайте използването на системата в main функция. Трябва да построите система и да демонстрирате записването ѝ в текстов файл. Проверете, че можете да прочетете от този файл нова система, и покажете, че двете са еднакви.

Забележка! Изисква се само слаба гаранция при възникване на изключение.

Вариант 2.

Реализирайте клас **Събитие**, който има следните характеристики:

- име - символен низ с произволна дължина;
- описание – символен низ с произволна дължина;
- продължителност в часове – неотрицателно цяло число;
- дали събитието се повтаря всеки месец – булева стойност.

Реализирайте методи за коректен жизнен цикъл, както и достъп до данните.

Предефинирайте оператори за вход и изход в поток и сравнение за равенство. Две събития са равни, ако имат еднакви стойности във всички полета.

Реализирайте клас **Задача**, която е събитие с допълнителната характеристика краен срок – дата. За датата не е нужно да реализирате прецизна валидация.

Реализирайте необходимите методи за коректен жизнен цикъл, както и оператори за сравнение (наредба по спешност). По-голяма е задачата с по-ранен краен срок. В случай на задачи с един и същи краен срок, по-голяма е задачата с по-голяма продължителност.

Реализирайте методи за задаване на тези характеристики. Предефинирайте оператори за вход и изход в поток.

Реализирайте клас **Среща**, която е събитие, но допълнително може да създава задача, чрез предефиниран функционален оператор с подходящи параметри. В срещата се съхраняват и списък от имената на гости, поканени за срещата. Добавете метод за добавяне гост към срещата. Добавете оператори за вход и изход в поток.

Реализирайте клас **Месечен Календар**, който се характеризира с месец, за който е създаден. Притежава произволен брой записани в него събития, задачи и срещи. За всяко събитие и среща са ясни денят и часът, в който започват.

Могат да се създават нови календари, копия на съществуващи, но информацията от вече съществуващ календар не може да бъде копирана във вече съществуващ такъв.

Реализирайте методи за коректен жизнен цикъл на календара, както и за достъп до данните.

Реализирайте също следните оператори:

- Оператори за събиране, които да добавят събитие, среща и задача. Задача може да се добави, само ако крайният ѝ срок е в рамките на месеца или след него. Ново събитие или среща се добавят в първия свободен час за месеца, в който не започва друго събитие.
- Оператор за изваждане, който премахва всички събития с дадено име.

- Оператор за индексирание, който дава достъп до задача по нейното име, ако такава е включена в календара. Ако повече от една задача отговаря на условията, да се върне най-малката.
- Оператор ++, който създава календар за следващия месец и записва в него всички задачи, които приключват в следващия месец или по-късно, както и всички повтарящи се събития в календара.

Демонстрирайте използването на календара в main функция. Трябва да построите календар за месец май и да прочетете от стандартния вход няколко срещи, събития и задачи, които да добавите в него. Добавете поне една задача, която да приключва в месеца и една, която да приключва през юни или по-късно. Демонстрирайте използването на оператора ++ и проверете, че правилните задачи са в получения календар.

Забележка! Изисква се само слаба гаранция при възникване на изключение.