Домашна работа - група 8, семинар

Срок за предаване: четвъртък, 12-ти май 2016г., 9:15ч (преди упражнението)
Споменатите в заданието по-долу класове Calendar, Event и Date са имплементираните по време на семинарните упражнения на 8-ма група и могат да бъдат намерени в GitHub: https://github.com/fmi-lab/oop-group8-seminars/tree/master/calendar

Задача 1. Времевите лордове са извънземни същества, могат да пътуват във времето. Това върви със своите отговорности - те трябва да внимават да не срещнат себе си, т.е. да отпътуват до място и момент във времето, където вече са били преди (иначе ще предизвикат времеви парадокс). Времевите лордове обаче ходят на най-различни места и не могат да помнят къде кога са ходили. Единствено пазят дневници на приключенията си, като в тези дневници записват събитията, на които са присъствали. Не за всяко събитие помнят къде се е случило, но винаги записват в кой момент от времето се е случило.
Помогнете им да организират спомените си, за да не предизвикват времеви парадокси. За целта:

1. Създайте **клас TimeLord**, който има член-данни името на времевия лорд и списък от календари (архиви със събитията, които е преживял). За списъка използвайте или динамичен масив (вектор), или свързан списък. Дефинирайте всичко необходимо от голямата четворка за класа, както и следните член-функции:
    a. **void TimeLord::newCalendar()** за добавяне на нов празен календар.
    b. **void TimeLord::newEvent(…)** с подходящи параметри. Функцията да добавя събитие, инициализирано с подадените параметри, в последния календар от списъка.
    c. **Calendar TimeLord::allArchives()**. Функцията да връща календар, обединяващ всички календари на Времевия лорд. За целта може да предефинирате Callendar::operator+ или Calendar::operator+=, или да дефинирате аналогична на тях функция в класа Calendar.
2. Променете класа Event така, че да съдържа информация и къде се е случило събитието (име на мястото - символен низ). Ако няма данни къде се е случило събитието - стойността да бъде празен низ.
3. Дефинирайте член-функции (внимавайте за правилното използване на рефенции и ключовата дума const):
    a. **bool Event::isHappeningAt(<place>, <date>, <hour>)** с подходящи типове на параметрите. Функцията да проверява дали събитието се е случило на посоченото място в посочения ден и час. Ако за събитието не е известно къде се е провело, се приема, че е валидно за всички места (т.е. функцията връща true, ако само датата и часът съвпадат).

   b. **bool Calendar::hasEventAt(<place>, <date>, <hour>)** с подходящи типове на параметрите, която връща истина, ако поне едно събитие от календара се е случило на даденото място в дадения момент.
   c. **bool TimeLord::canVisit(<place>, <date>, <hour>)** с подходящи типове на параметрите. Функцията да преглежда календарите на Времевия лорд и да връща истина, ако е безопасно той да посети даденото място в дадения ден и час.
4. Тествайте създадения клас с main програма, която създава обект от тип TimeLord, след което последователно прави два календара с по три събития във всеки, използвайки член-функциите на класа TimeLord. След това програмата да отпечатва обединението на всичките събития и да проверява дали така създадения Времеви лорд може да посети безопасно ФМИ на 2016-05-01 в 10ч сутринта.

Задача 2. Времевите лордове решават да си направят конкурс в различни категории. Създайте **клас Competition** с член-данни дата и място, на която ще се проведе състезанието, и списък от участниците. Дефинирайте и член-функции:

1. **bool Competition::register(<подходящ тип> timeLord)**, който добавя Времеви лорд в регистрираните участници. Времевият лорд се регистрира успешно само ако може безопасно да посети мястото на състезанието на дадената дата (приемете, че състезанието се провежда цял ден - от 0 до 24ч).
2. **int Competition::numParticipants()**, който връща броя участници.
3. Функция, която отпечатва на екрана данни за участниците в състезанието.
4. **TimeLord Competition::winner()**, която намира Времевия лорд, победител в състезанието. Победителят да се избира на произволен принцип.

Избирането на победител на произволен принцип не е особено честно. Създайте наследници на класа Competition, които отговарят съответно на следните състезателни категории:

1. Най-много преживени събития.
2. Най-ранна посетена дата.
3. Най-продължително време, прекарано в 15-ти век.

Всеки наследник да предефинира по подходящ начин Competition::winner() така че да отговаря на съответния критерий. За целта в класа TimeLord добавете съответните член-функции:

- int TimeLord::numEvents() const
- Date TimeLord::earliestVisitedDate() const
- int TimeLord::timeSpentInCentury(int century) const

Ако са ви необходими нови функции в и някои от другите класове, можете да ги добавяте.

Напишете main програма, в която създавате списък от състезания в различни категории, регистрирате по няколко Времеви лорда за всяка от тях и отпечатвате данни за победителя във всяка от тях.

Екстра кредит: Реализирайте методите в класа TimeLord (numEvents, earliestVisitedDate, timeSpentInCentury), използвайки map/reduce.

Екстра кредит 2: Предайте домашното си чрез GitHub. За целта влезте в акаунта си на github.com, отидете на https://github.com/fmi-lab/oop-group8-seminars и натиснете бутона Fork (горе вдясно). Това ще добави копие на проекта във вашия акаунт. След това разучете как да изтеглите чрез git кода на компютъра си. (тук има списък с приложения, които могат да ви свършат работа).
Добавете в изтегления код решенията на задачите от домашното, след което ги направете публични във вашия fork на проекта (например чрез команди commit / push в приложението, което сте изтеглили). В moodle заданието напишете само вашия GitHub username.