

Преди да започнем

Привет, колеги радвам се че сте намерили време да порешавате задачи. Задачите ни дават възможност да потренираме наученият материал и да пробваме нещо ново. Следвайте инструкциите и ще се справите с тях.

Първата ни задача е да си направим [GitHub](#) профил. GitHub е платформа за споделяне на код, освен че дава възможност да демонстрирате уменията си по програмиране имате възможност и да разгледате стотици дори хиляди проекти на ваши колеги, които да ви помогнат да станете по-добър в професията.

- Влезте в GitHub и се регистрирайте с потребителско име / парола и E-mail адрес
- Изпратете Вашият профил заедно с трите си имена и факултетният номер, на E-mail адреса на курса : unijs@mihail-petrov.me за да знаем кой сте при проверката на вашите домашни
- Направете нов проект в GitHub и реализирайте първият си **commit**. Задължително проекта си **uni-js-hello-world** и качете първият си файл.

GitHub е платформа която използва системата за управление на версиите на кода наречена **Git** една от най-популярните система в индустрията. Силно препоръчваме да се запознаете с основите на системата, никак не е сложно да започнете.

Вече получихте първите си **50 точки**, сега нека започнем да разработваме малко практически проекти.

Максимален резултат от раздела

50 точки

Органайзер на събития

Въведение в проекта:

Местна Пловдивска компания специализираща в забавления и организация на луди партита иска да разшири дейността си и за тази цел се нуждае от система за управление на събития.

Фирмата се свързва с вас, с молба да разработите набор от скриптове с които да помогнете на ентусиастите да посрещнат предстоящите празници.

Всяко едно събитие на фирмата се характеризира със следните основни данни.

уникален идентификатор	Генерира се от системата при съхранение на ново събитие
име на събитието	<i>не е необходимо да е уникално</i>
флаг за достъп	Дефинира дали събитието е подходящо за лица под 18 годишна възраст

Всяко събитие се посещава от **клиенти**, характеризиращи се със следните данни:

име и фамилия	Низов тип
пол	булев / низов тип
възраст	целочислен тип

Основни задачи

Да се създаде органаизър за събития който да изпълнява следните задачи

1. Съхранява в колекция списък с всички събития, които са организирани. *Помислете за подходящ начин да съхранявате събитията както индивидуално така и групово.*
2. Извежда всички вече съхранени събития, като визуализира цялата необходима информация за тях.
Тази информация трябва да съдържа свойствата които са описани във заданието, като име на събитието, уникален идентификатор и др.

Примерна визуализация на единично събития
Grand opening of new Club : 18+

3. Изтрива събитие по **уникален идентификатор**, и извежда съобщение за успешно извършена операция.

Не забравяйте, събитията трябва да имат уникален идентификатор който да ги различава едно от друго. Имената на събитията може да съвпадат затова трябва да помислите за начин по който да ги идентифицирате, числово или буквено
4. Създава ново събитие. **Задължителни атрибути на събитието са неговото име.** Ако потребителя не подаде флаг, указващ дали събитието е подходящо за непълнолетни то по подразбиране е.

Ако потребителя не подаде име на събитието, е необходимо да се обработи невалидната операция по подходящ начин.
5. Актуализира събитие по уникален идентификатор и изведете съобщение за правилно извършена операция. *Проверявайте дали актуализираните данни отговарят на изискванията за създаване на събитие.*
6. Добавете клиент към вече създадено събитие. Ако възрастта на клиента не му позволява да присъства на събитието, известете с помощта на необходимото съобщение.
7. Визуализирайте списък с всички клиенти които присъстват на определено събитие. Предоставете възможност да бъдат филтрирани по пол, тоест да се визуализират само мъжете или само жените.
8. Премахнете присъстващ потребител от събитието.

Допълнителни задачи част 1

1. Създайте функционалност който да спира добавянето на събития или добавянето на клиенти на централно ниво. Когато бъде активирана при опит да се добави събитие или клиент потребителя получава съобщение че операцията не може да бъде извършена, защото системата е затворена.

Припомнете си, когато реализирате функционалност който трябва да работи за определен тип операции е необходимо да имате променлива или константа която да съхранява състоянието на вашата конфигурация

2. Добавете възможност за запазване на текуща дата на събитието.

Изпълнението на тази задача предполага, модифициране на структурата за съхранение на данни за събитието. Помислете по какъв начин ще съхраните данните за датата месеца и годината.

3. Създайте функционалност за извеждане на събитието с най-много добавени клиенти. Ако такова не съществува (всички са с равен брой) или не съществуват събития изведете необходимите съобщения, по ваш избор.
4. Изведете всички събития които са подходящи за малолетни посетители.
5. Изведете всички събития като ги групирате, събитията които са предназначени за пълнолетни посетители трябва да имат звездичка пред името си "*" а тези подходящи за непълнолетни диез "#"
6. Създайте механизъм за филтриране на събития по определен критерии. Функцията трябва да има възможност да получава име / или флаг за достъп и да визуализира само тези събития които отговарят на критериите.

Припомнете си можете да подадете **функция като аргумент на друга **функция** за да разширите възможните приложения на нейната употреба**

Допълнителни задачи част 2

Работата по проекта върви много приятно, фирмата е доволна от вас а партитата, които се организират са безпаметни и епични. Получавате обаждане от изпълнителният директор на вашият клиент, с допълнителни искания и желания.

1. Оказва се че фирмата се е сетила в следствие че партитата, които организира не са благотворителни, и следователно трябва да събира някаква входна такса от всички клиенти които я посещават.
2. Добавете, свойство цена към всяко събитие което организирате. Цената не е задължително свойство, всяко събитие което е регистрирано без цена, става автоматично безплатно.
3. Всички събития, които са платени трябва да визуализират заглавията си със знака \$ пред имената си. Безплатни събития трябва да визуализират имената си със знак “!”
4. Всеки регистриран клиент, в системата трябва да разполага с портфейл. Портфейла съдържа пари които се намаляват при регистрация за ново събитие. Ако потребителя няма пари в портфейла си, системата не го регистрира.
5. Фирмата добавя понятие **VIP клиент**. VIP е всеки който е добавен като клиент на поне 5 събития. VIP клиентите не заплащат такса при посещението си на следващото събитие. При регистрация за шестото си събитие статуса им се нулира и те отново се превръщат в обикновенни клиенти.

Максимален резултат от раздела

20 точки

Допълнителни задачи част 3

Толкова много събития бяха организирани вече че бройката им малко по малко започна да се губи, Телефона ви отново звъни настоятелно и нова вълна от желания ви залива. Клиентите ви искам да имат по голям контрол над случващото се в организацията на събитията.

1. Създайте функционалност за архивиране на събития. Архивираните събития не могат да приемат гости. Архивираните събития, могат да бъдат само и единствено преглеждани като такива
2. Названието на всяко архивирано събитие трябва да започва със символа ~
3. Създайте функционалност за листинг на архивирани събития. Разширете функционалността на листинга така че да могат да се листват
 - Всички събития
 - Само тези събития които предстоят да се посещават от клиенти
 - Само архивираните събития
4. Създайте функционалност за визуализация на приходите от клиенти, които едно архивирано събитие е генерирало
Припомнете си, клиентите ви вече имат портфейли всяко събитие изисква от тях определена сума
5. Създайте функционалност за оценка на събитията **рейтинг**. Всяко едно събития в началото на своето създаване има рейтинг 0.
 - Максималният рейтинг който може да се получи е 6 а минималният е 0.
 - Клиентите оценяват събитието, с точкова система от 1 до 10.
 - Въведете защита която да елиминира възможността за въвеждане на точки различни от упоменатите
 - Едно събитие може да се оцени само от клиент който е бил регистриран за него
 - Могат да се оценяват само архивирани събития
6. Разширете функционалността на листинг на събития, която да показва рейтинга на всяко едно събития. Ако събитието все още не е оценено, да се визуализира предстой актуализация

Работа по заданието и предаване на крайният резултат

В рамките на първите **три седмици** от курса ще работите по първото задание с идеята да свикнете със синтаксиса и възможностите на езика.

Всички функционалности които са описани трябва да бъдат реализирани в рамките на конзолно приложение и да се изпълняват на ръка от оператора. За да демонстрирате че вашата функционалност работи е необходимо да предоставите тестови файлове, с примерни данни които да могат да бъдат изпълнени от проверяващият.

Използвайте коментари. В рамките на тестовите файлове запишете, за кое условие или задача се отнасят вашите тестове, бъдете подробни всяка една помощ от ваша страна спомага за по лесното проверяване на задачите в последствие.

Заданието се разработва и качва **задължително и само** в рамките на GitHub проверяват се всички commit- и които са направени преди крайната дата на заданието **01.02.2019 г.**

Именувайте проекта си : **projectjs-event-organizer**

Няма да се приемат решения пратени като прикачени файлове по E-mail. В E-mail а си очаквам само и единствено вашият GitHub username

Проверката на заданието както и крайните резултати ще получите до една седмица от неговият край.

Пожелавам ви приятна работа