



Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика

ДОМАШНА РАБОТА №2

курс Обектно-ориентирано програмиране
специалност Софтуерно инженерство
летен семестър 2018/2019 г.

СРОК ЗА ПРЕДАВАНЕ: посочен в Moodle

Изисквания за предаване:

- Предаване на домашното в указания срок от всеки студент като .zip архив със следното име: (номер_на_домашно)_SI_(курс)_(група)_(факултетен_номер), където:
 - (номер_на_домашно) е цяло число, отговарящо на номера на домашното за което е отнася решението (например 1);
 - (курс) е цяло число, отговарящо на курс (например 1);
 - (група) е цяло число, отговарящо на групата Ви (например 1);
 - (факултетен_номер) е цяло число, отговарящо на факултетния Ви номер (например 63666);
- Архивът да съдържа само изходен код (.cpp и .h файлове) с решение отговарящо на условията на задачите, като файловете изходен код за всяка задача трябва да са разположени в папка с име (номер_на_задача), където (номер_на_задача) е номера на задачата към която се отнася решението;
- **Не е разрешено** да ползвате класове от библиотеката STL като std::string, std::vector, std::stack и др.
- Качване на архива на посоченото място в Moodle;

Пример за .zip архив за домашно: 2_SI_1_1_63666.zip

Задача 1. MagicBox

3 точки

Магьосникът Маг-О-Кодерски е магьосник в местното за ФМИ заведение BeerOverflow. Там той изпълнява всяка вечер своя коронен номер - всеки посетител на BeerOverflow пуска нещо в "магическата кутия" на Маг-О-Кодерски, а в края на вечерта кутия изхвърля един предмет във въздуха, на случаен принцип.

Маг-О-Кодерски също е и програмист. Той иска да напише програма, в която да въвежда променливи от даден тип (един път може да са `int`, друг път `char`, а трети път - `Rational`), и при извикване на даден метод, "кутията" да му връща случайна променлива от вече въведените.

Да се напише клас **MagicBox** със следните член-данни:

- Динамичен масив (типа на променливите не е ясен предварително, но знаем че всички променливи са от един тип)

Методи:

- *insert* – добавя елемент към кутията;
- *pop* – премахва случаен елемент от кутията. Ако кутията е празна, да показва подходящо съобщение;
- *list* – извежда на екрана списък с елементите в кутията.

Може да дефинирате допълнителни методи и член-данни.

Задача 2. MMORPG игра

3 точки

Реализирайте следните класове, които ще служат като част от MMORPG игра

Клас **GameCharacter**:

GameCharacter е базовият герой - той не може да бъде избран от играч, само служи като основа на другите класове

Член-данни:

- Име на героя
- Точки живот (HP)
- Точки за магия (MP)
- Точки атака (AP)
- Точки защита (DP)

Методи:

- *getattacked()* - *героят губи няколко от своите точки живот (точки атака на противника - точки защита на героя. Ако разликата е < 0, нищо не се променя в точките живот) (getattacked() приема като аргумент int - точките атака на противника).*

Всички точки на героя са зададени в конструктора, и не могат да бъдат променяни

Клас **Warrior**:

Има следните характеристики:

- Име на героя - "Warrior"
- HP = 20
- MP = 5
- AP = 4
- DP = 1

Клас **Guardian**:

Има следните характеристики:

- Име на героя - "Guardian"
- HP = 40
- MP = 25
- AP = 1
- DP = 3

Клас **Wizzard**:

Има следните характеристики:

- Име на героя - "Wizzard"
- HP = 30
- MP = 30
- AP = 2
- DP = 2

Задача 3. Логин система

4 точки

(Ако нещо в задачата не е пояснено, се реализира по избран от Вас начин)

За реализирането на задачата трябва да се използва наследяване.

Реализирайте главна програма с тестове.

Реализирайте част от логин система за форум, която има следните типове потребители:

- **Guest**, който има следната информация
 - **IP Address** - IP адрес на потребителя
- **User**
 - User има всичката информация/функционалност от **Guest** и
 - **Username**
 - **Password**
 - **Title**
 - **Password** да се съхранява в криптиран формат
 - **User** не може да променя своя **Title**

- **User** не може да променя своя **Username**
- **User** може да променя паролата си, като трябва да даде старата си парола при смяната
- **Power User**
 - **Power User** има всичката информация/функционалност от **User** и
 - **Reputation** - колко е допринесъл потребителят за форума
 - **Power User** не може да променя репутацията си
 - **Power User** не може да променя своя **Title**
 - Всички други потребители освен текущия могат да дават по 1 **Reputation**, т.е потребител може да даде 1 **Reputation** на текущия потребител само ако имат различен **Username**
- **VIP**
 - **VIP** има всичката информация/функционалност на **User**
 - **VIP** може сам да променя **Title**-а си
- **Admin**
 - **Admin** има всичката информация/функционалност на **VIP, Power User**
 - **Admin** може да променя своя **Username**
 - **Admin** може да променя **Username** на друг потребител

HINTS

- IP address, Username, Password, Title - char *
- Reputation - int
- Примери за криптиране на паролата - XOR, ROR/ROL, NOT, Addition/Subtraction, Reverse bits, etc.
- Промяната на паролата е един метод, на който се подава стара и нова парола
- PowerUser Reputation +1 - на метода в PowerUser се подава кой дава +1 репутация (друг потребител)
- В Admin на метода за промяна на Username на потребител се подава User и нов username