



ООП – Практикум: Домашна работа №2

Задача 1: Инвентар/Inventory

Имплементирайте инвентар на герой от игра. Той може бъде два типа, **Backpack** и **Equipment**. В backpack-а могат да се сложат два типа предмети - **Material** и **Money**, докато в **Equipment** могат да се сложат - **Weapon** и **Armor**. За целта ще са ви нужни следните класове:

Inventory - бива два типа - Equipment или Backpack.

Трябва да предефинирате:

- operator <<, който ще отпечата данните на класа в него (да има същата функционалност като метод print())
- operator ==, който сравнява две Inventory-та по тяхната тежест (по количеството заети слотове/slots)
- operator !=, който сравнява две Inventory-та по тяхната тежест (по количеството заети слотове/slots)

Backpack - бива два типа – **Money** или **Materials**.

Класът ще съдържа и член-данна *slots*, която ще служи за ограничаване на максималния капацитет на торбата – 16 слота при материали и неограничен брой при пари.

Трябва да имплементирате следните методи:

- Метод, който проверява дали торбата е празна
- Метод, който проверява дали торбата е пълна
- Метод, който занулява (изчиства) торбата
- Метод, който добавя към торбата (пари или материал)
- Метод, който премахва от торбата (пари или материал)

Equipment - два типа – **Armor** и **Weapon**.

Като член-данна, той ще има динамичен масив, който е от един от двата възможни типа.

Класът ще съдържа и член-данна *slots*, която ще служи за ограничаване на максималния капацитет на гардероба – 24 слота.

Трябва да имплементирате следните методи:

- Метод, който проверява дали equipment е празен
- Метод, който проверява дали equipment е пълен
- Метод, който занулява (изчиства) equipment
- Метод, който добавя към equipment





ООП – Практикум: Домашна работа №2

- Метод, който премахва от equipment
- Метод, който връща най-добрия equipment.
- `operator==` и `operator!=`, сравняващи 2 equipment-а по техния GearScore/WeaponScore.

Materials - четири типа - herbs, ores, cloth, essence

Като член-данна ще се съдържа и количеството от всеки тип.

За да се запълни един слот в чантата, едно от следните условия трябва да бъде изпълнено:

- Количество до 10 есенции включително се брои за 1 слот
- Количество до 20 herbs, ores, cloth включително се брои за 1 слот

Money

Ще съдържа две член-данни – gold и silver.

Когато количеството на среброто достигне 100, автоматично се конвертира едно ниво нагоре (bronze към silver, silver към gold).

Note: Забележете, че е НЕВЪЗМОЖНО да имате повече от 99 сребро в даден момент, защото автоматично ще трябва да се качи едно ниво нагоре.

Пример: Ако към 0 gold 99 silver добавим 3 silver, получаваме 1 gold 2 silver

Бонус: Реализирайте оптимално Money, като използвате само една член-данна.

Armor

Ще съдържа:

- тип - cloth, leather, mail
- defense - цяло число
- набор от ефекти, като всеки ефект се състои от име и сила (положително число). **Note: Не трябва да се прави отделен клас/структура Effect.** Всяка отделна броня заема точно 1 слот.
- Gearscore - съдържа стойностите на всички ефекти и по този начин изчислява рейтинга на самата броня.

Имплементирайте нужните според вас методи за дадения клас.

Weapon:

Ще съдържа:

- тип – one-handed, two-handed
- вид – axe, sword, dagger, mace, staff, etc.
Note: Staff например може да е само two-handed, а dagger - one-handed.
- hit damage - приема най-ниска и най-висока възможна сила.





ООП – Практикум: Домашна работа №2

- набор от ефекти, като всеки ефект се състои от име и сила(положително число). **Note: Не трябва да се прави отделен клас/структура Effect.** Всяко оръжие заема точно толкова слотове, колкото ръце са нужни за използването му.
- WeaponScore - изчислява, като се съберат средният hit-damage и стойностите на всички ефекти и се умножат в зависимост от типа - one-hand $\times 0.75$ / two-handed $\times 1.5$

Трябва да имплементирате следните методи:

- Метод, който променя вида на оръжието
Note: Съобразете как това влияе на типа.
- Метод, който променя типа на оръжието.
- Метод, който добавя ефект
- Метод, който премахва ефект
- Метод, който променя силата на даден ефект.

Допълнително имплементирайте нужните според вас методи за дадения клас.





ООП – Практикум: Домашна работа №2

Пояснение:

- Опитайте се да напишете максимално ефективен код, както по отношение на брой редове, така и по отношение на време за изпълнение. Помислете къде може да се намали броят на повторенията на циклите или да се намали броят на променливите, които използвате за решаване на задачата. Реализирайте задачите спазвайки добрите ООП практики, за които се говори по време на лекции и упражнения.
- Файловете с решенията може да съдържат само стандартните символи с кодове от 0127 (не се разрешава използване на кирилица, например в стринговете или коментарите!).
- Предадените от вас решения трябва да могат да се компилират успешно на Visual C++ или GCC.
- **Разрешено** е да ползвате класове от библиотеката STL като `std::string`, `std::vector`, `std::stack` и др
- Първото нещо във всеки от файловете, които предавате, трябва да бъде коментарен блок, който носи информация за съдържанието на файла. Този коментар трябва да изглежда точно така, както е показано по-долу, като в него попълните информация за Вас. За улеснение, просто копирайте дадения по-долу блок и попълнете в него необходимите данни, вместо текста, маркиран с ъглови скоби. Обърнете внимание, че на първия ред след наклонената черта има две звезди и че във файловете не може да се съдържат символи на кирилица.

```
/**
 * Solution to homework assignment 2
 * Object Oriented Programming Course
 * Faculty of Mathematics and Informatics of Sofia University
 * Summer semester 2020/2021 *
 * @author <вашето име> m
 * @idnumber <вашият факултетен номер>
 * @task <номер на задача>
 * @compiler <използван компилатор - GCC или VC>
 */
```





ООП – Практикум: Домашна работа №2

Например един попълнен блок за студент с име Иван Иванов, Ф.н:12345, който предава задача 2, компилирана с GCC, трябва да изглежда така:

```
/**
 * Solution to homework assignment 2
 * Object Oriented Programming Course
 * Faculty of Mathematics and Informatics of Sofia University
 * Summer semester 2020/2021
 *
 * @author Ivan Ivanov
 * @idnumber 12345
 * @task 1
 * @compiler GCC
 */
```

Изисквания за предаване:

- Всички задачи ще бъдат проверени автоматично за преписване. Файловете с голямо съвпадение ще бъдат проверени ръчно и при установено плагиатство ще бъдат анулирани.
- 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 12345);

Пример за .zip архив за домашно: 2_SI_1_1_12345.zip

- Архивът да съдържа само изходен код (.cpp и .h/.hpp 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;
- Качване на архива на посоченото място в Moodle;

Софийски университет "Св. Климент Охридски"

Факултет по математика и информатика

