

# Oefening: "RPG Inventory & Items"

---

We maken een mini-console RPG-inventaris waarbij verschillende soorten items kunnen worden opgeslagen, beheerd en weergegeven.

## Doelstellingen

Toepassen van de leerstof:

- Basis syntax en klassen
- Compositie
- Inheritance en polymorphism
- Abstracte klassen & interfaces
- Operator overloading
- Pointers & dynamic memory allocation
- Exceptions
- File streams

## Opgave

### 1 Klasse Item

Dit is een abstracte klasse voor items. Er is een virtuele destructor aanwezig en enkele pure virtuele functies:

```
virtual void printInfo() const = 0;  
virtual double value() const = 0;
```

### 2 Klasse Weapon (erft van Item)

Deze klasse bevat de naam en damage hoeveelheid van het wapen. Bij de constructor moet een validatie aanwezig zijn die een exception smijt als de damage hoeveelheid niet groter is dan nul.

- De methode `value()` wordt overschreven en toont de damage \* 10.
- De methode `printInfo()` wordt overschreven en toont de naam en de damage hoeveelheid.
- De operator `==` wordt overladen en vergelijkt de naam én damage.

### 3 Klasse Potion (erft van Item)

Deze klasse bevat de naam en restore hoeveelheid van de potion. Bij de constructor moet een validatie aanwezig zijn die een exception smijt als de restore hoeveelheid niet groter is dan nul.

- De methode `value()` wordt overschreven en toont de restore \* 4.
- De methode `printInfo()` wordt overschreven en toont de naam en de restore hoeveelheid.
- De operator `==` wordt overladen en vergelijkt de naam én restore.

#### 4 Klasse Inventory (compositie van item)

Deze klasse bevat het aantal items en een array van 100 items groot. De constructor initialiseert alle elementen in de array op een `nullptr` en de destructor verwijdert elk dynamisch aangemaakt item via `delete`.

- Er is een methode `void add(Item*)` waarmee een nieuw item kan toegevoegd worden aan de array.
- Er is een methode `void printAll() const` waarmee alle items in de array op het scherm worden weergegeven.
- Er is een methode `bool compare(int, int)` waarmee twee items worden vergeleken. Gebruik `dynamic_cast` om veilig te kunnen werken. Als de items niet van hetzelfde type zijn, wordt een exception gegooid.
- Er is een methode `saveToFile(const std::string& filename)` waarmee alle items in een bestand worden weggeschreven.
- Er is een methode `loadFromFile(const std::string& filename)` waarmee alle items uit een bestand worden ingelezen.

#### 5 Hoofdprogramma (main)

Maakt een Inventory aan op de stack en toon vervolgens in een lus hetvolgende menu:

1. Add weapon
2. Add potion
3. Show all items
4. Compare two items
5. Save inventory to file
6. Load inventory from file
7. Quit

Afhankelijk van de keuze van de gebruiker voorzie je onderstaande functionaliteiten. Als er exceptions optreden, worden die opgevangen en op de console getoond.

- Toevoegen

De gebruiker kan de waarden invullen en er wordt een nieuw item (weapon of potion) aangemaakt op de heap wat dan wordt toegevoegd aan de inventory.

- Toon alles

Toon alle info op de console door de polymorfische `printInfo()` te gebruiken.

- Vergelijken

Je vraagt de indices van de twee items die moeten vergeleken worden en maakt dan gebruik van de `==` operator om dat te doen.

- Wegschrijven naar bestand

Je vraagt aan de gebruiker de bestandsnaam op en schrijft vervolgens per item het itemtype weg gevolgd door de parameters . Bijvoorbeeld:

```
WEAPON Excalibur 18  
POTION Health 40
```

- Inlezen van bestand

Je vraagt aan de gebruiker de bestandsnaam op en leest vervolgens elk item in, maakt daarvan een nieuw object aan op de heap en voegt deze dan toe aan de inventory. Vooraf heb je alle bestaande items uit de inventory verwijderd.

## Restricties

- De gebruiker mag geen spaties gebruiken bij de ingave van een naam.