Enkapsulācija:

Character un Enemie klases: zmantota enkapsulācija, lai saglabātu un pārvaldītu "health” un citas īpašības kā privātus mainīgos (piemēram, veselība, weapon). Šiem propertie var piekļūt un tos mainīt, izmantojot publiskās metodes.

Enemy klase izmanto agresijas un armor mainīgos lielumus, kas ir privāti vai aizsargāti, lai iekšējās darbības būtu slēptas no citām klasēm.

Mantošana:

Enemie klase ir mantota no Character klases. Tas nozīmē, ka Enemie var izmantot Character propertijus un metodes, piemēram, GetHit() un Attack(), bez nepieciešamības tās atkārtoti definēt. Tas var arī ignorēt tādas metodes kā Attack(), lai mainītu vai paplašinātu uzvedību.

Polimorfisms:

Attack() metode ir polimorfisma piemērs. Lai gan Character klase definē vispārīgu Attack() metodi, apakšklases, piemēram, Tank un Berserker, ignorē metodi, lai nodrošinātu savu specifisko uzvedību.

public override int Attack()  
{

}

Izsaucot Attack() Enemy tipa objektam, atkarībā no faktiskā objekta veida tiek izsaukta pareizā metodes versija (no Tank, Berserker vai Enemy).

Abstrakcija:

Weapon klase ir abstrakta, kas nozīmē, ka tā nodrošina pamatstruktūru (metode GetDamage()), bet faktisko ieviešanu atstāj atvasinātām klasēm (piemēram, FireSword vai BasicWeapon).

Realizētajs papilduzdevums: 3 dažādi pretinieki. +pat tas nebija minēts uzdevumā, es izdarīju tā, lai spēle nebeigtos tad, kad spēlētajs nogalina 3 pretiniekus, spēle ģenerēs jaunus pretiniekus.

Artūrs Umbraško