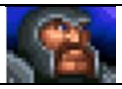
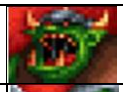
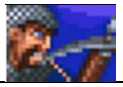
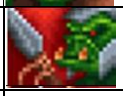
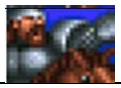
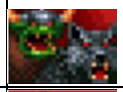
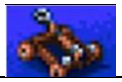
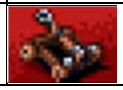
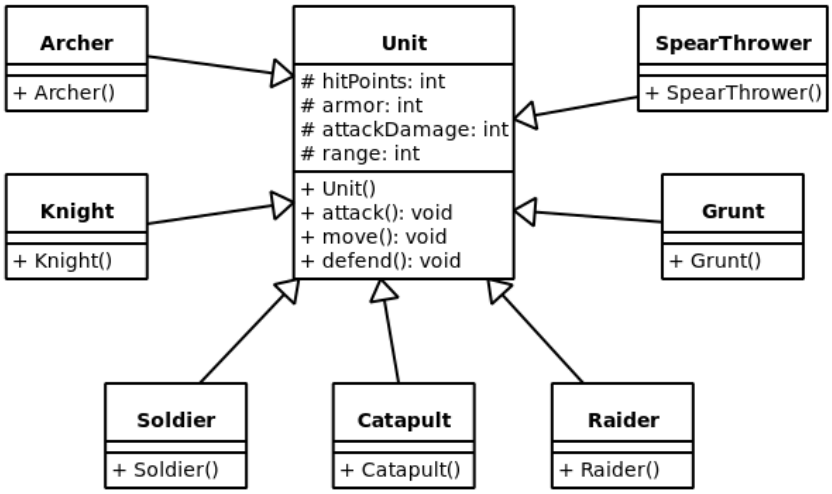
Course: Programming Paradigms and Technologies
Topic: Design Patterns

The Warcraft game: Humans vs Orcs© is a strategy game in real time where two species fight between them: humans and orcs. Every species has the same number of military units, but they are specialised for every specie. We can see a summary in the following table:

Type	Species	
	Humans	Orcs
Infantry	 Soldier	 Grunt
Archer	 Archer	 Spear Thrower
Horseman	 Knight	 Raiders
SiegeMachine	 Catapult	 Catapult

These characters have to be represented in memory according to the next hierarchy of classes:



Course: Programming Paradigms and Technologies
Topic: Design Patterns

From the above hierarchy, it has been modelled an artificial intelligence (A.I.) to manage the Orcs. A fragment of that class AI**Orcs** is on the right, where we can observe a method which creates a combat unit made of several units of each type.

```
// Artificial Intelligence for Orcs
public class AIOrcs {

/**
  * Generation of a group of attack for A.I.
  * @return An attack group with Orcs
*/
public Unit[] createGroupOfAttack()
{
        // Array of Orcs Units
        Unit[] groupOfAttack = new Unit[10];

        // 4 x Infantry
        for(int x = 0; x < 4; x++)
                groupOfAttack[x] = new Grunt();
        // 3 x archer
        for(int x = 4; x < 7; x++)
                groupOfAttack[x] = new SpareThrower();
        // 2 x horseman
        groupOfAttack[7] = new Raider();
        groupOfAttack[8] = new Raider();
        // 1 x siege machine
        groupOfAttack[9] = new Catapult();

        return groupOfAttack;
}
```

Programmers could see that this A.I. works very well, so they have decided to use it for both Orcs and Humans.

Therefore, this class will be renamed as **AICommon**. Then, it will be added a constructor **AICommon(String species)** to create orcs units ("orcs") or human units ("humans") depending on the string specie. We want to apply the most suitable design pattern to avoid creating a conditional sentence for every **new** when a specific type of unit is created.

Answer to the following questions:

1. From the design patterns which have been studied in the lectures, which would be the most appropriate in order to facilitate to the class **AICommon** to choose the specie, and then create a specific number of units to be managed in every object? **Justify the answer.**
2. After applying this pattern, could it be possible to generate units for both species in the same attack group? **Justify the answer.**
3. Which structure of classes has to be added, how will it integrate with the classes which already exist? Draw the diagram. Indicate the function for every new class.
4. Implement the class AICommon with its new constructor and its method createGroupOfAttack().
5. Would it be very difficult to add a third type of species to the game? For example:

   Species No-Dead (Skeleton/LancerZombie/HorsemanNoHead/CannonOfPlague)

   Which changes should be done?

6. Would it be very difficult to add a new type of units for each specie? For example:

Type	**Species**		
	Humans	**Orcs**	**No-Dead**
Magician	 Conjurer	 Sorcerer	 Necromancer

   Which modification should be done?