**Using Inheritance & Abstraction – Tasks & Ideas**

**Ideas for applications with abstraction / interfaces (column C)**

Here are some ideas for small applications where you show grades of Abstraction:

**Hospital Management Tool:**

Simulate a hospital management tool: this system has patients, doctors, nurses, assistants. It has departments and buildings.

Define a set of typical attributes for these classes.

Define a hierarchy of classes. Which classes are abstract? Which classes should have objects?

**Simulation of Predator and Prey:**

Create a simulation of animals which are either predators or prey. Let the user decide what he wants to be.

Depending on the animal the user chooses, this animal has a certain position in the hierarchy.

**Simulation of Social Network:**

Implement a small application which simulates users on a social network. See separate worksheet.

**McDonalds Simulation:**

McDonalds is a very successful fast-food chain which operates worldwide (in most countries). Simulate the different types of Burgers, thereby also making the distinction between countries. (note: BigMac is the same type in every country, but has considerable price differences).

What would be an abstract type of burger? And what properties does a burger have?

Note: consider using the Decorator Design Pattern to implement variations of hamburgers…. (see Decorator Design Pattern hand-out for example)

Organigram of your Business:

Large businesses usually work with hierarchies, which can be mapped /simulated in a small application. You then have an abstract layer of departments and concrete instances / classes of the people below. Make the application interactive, so you can add / delete staff from departments. (and possibly also create , delete departments)

Zork Game Revisited:

Zork was one mini project you could follow in Part A of this module. Now you can extend this and create a larger world where inheritance & abstraction is included.

What classes become abstract?

Other Ideas:

Also check the Java Book, chapters 19, 20 & 21 on inheritance, polymorphism & abstractions for exercises & ideas.