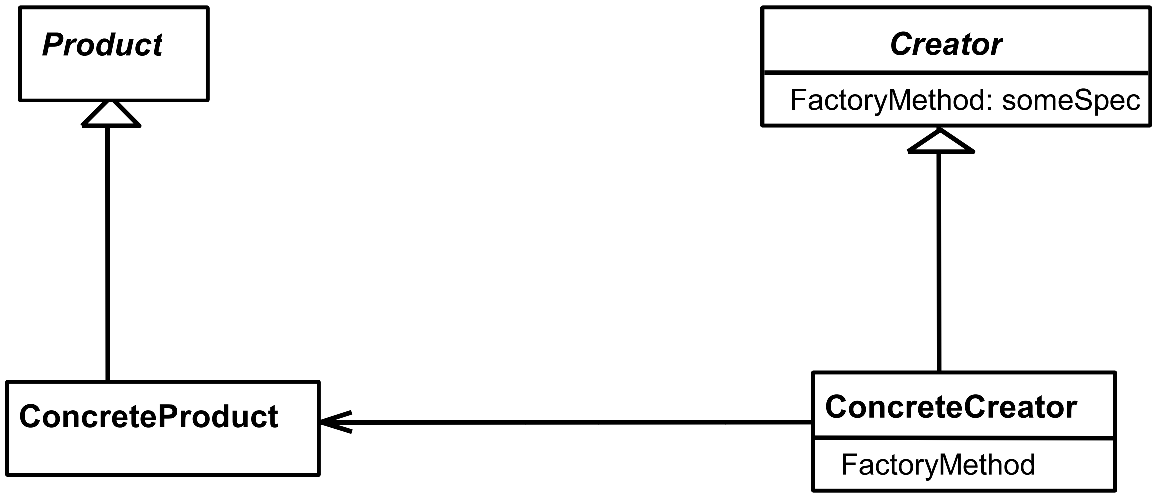
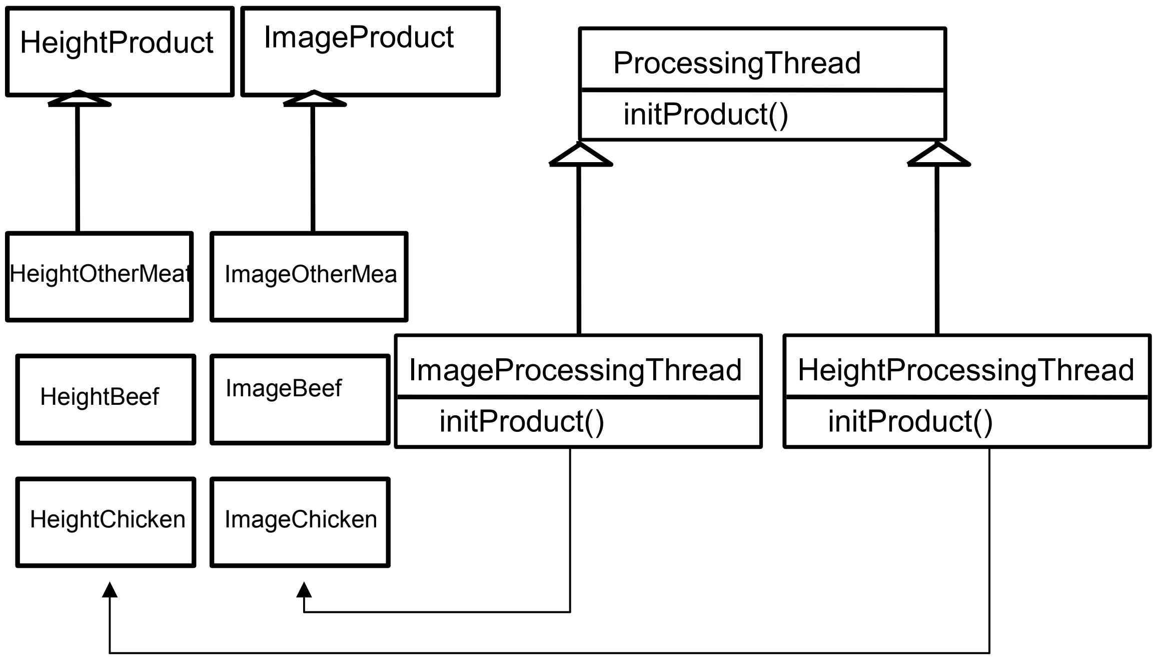


UML Diagram from GT Squeak Slides

1. Creational Design Patterns allow a system to be “independent of how its objects are created, composed, and represented.” Factory methods design pattern is a design in this category. Factory methods design is used to create an interface for which generic abstract class methods can be overridden to decide how a new object should be created. If a programmer needs to do specific processing on an image, but the processing depends on the content, the programmer can create a factory class that parses an xml file to decide which concrete implementation of the abstract image processor should be instantiated. A more involved example could be a system where either height readings or image processing is conducted for a food processing plant depending on the user selected xml configuration file that gets loaded. The height readings need also to be conducted one way for beef, another for chicken, ect. Also the image processing is also specific to the type of product that needs to be evaluated.



2. Pair programming is a combined development system for code in which one person is the driver or the one writing the code actively engaged into the specifics of a particular method or class while the other is the navigator thinking ahead about how the method or class fits together with the entirety of the program. Every half-hour or pre-determined time duration switch roles. Logical errors and careless mistakes can be avoided with two people evaluating the code. Pair programming fits into the XP paradigm by promoting values such as communication between team members and increased productivity. It allows each member in the pair to understand the code fully and give feedback since each member may have certain strengths that can be combined to complement their partner.

Unit testing is an important aspect of XP programming. Refactoring occurs on code to simplify the design or clarify variables and other elements without changing functionality so that the code can be more flexible and extendible in the future. It can be easy to break your code when refactoring so unit testing is useful to see if the code's functionality is intact when organizational changes are made and in case you do break something it is easier to know precisely where you introduced a bug. This saves time and the programmer from potential headaches.

3. Garbage collection allows the programmer from having to worry about the memory that he uses in the program so it saves him a significant amount of time. Determining when variables are no longer used and when they need to be deleted is time consuming. Also, it keeps the programmer from committing pesky runtime errors such as referencing a deleted memory location that may be used by another program. Some of these errors may go undetected, but produce weird or unexpected results. A disadvantage of garbage collection is a slight decrease in the speed of performance. Reference counting works well for systems with low memory that need memory freed as soon as possible with minimum overhead. However, cyclic references in which a reference directly or indirectly refers to itself will never equal zero. So in a simple implementation these objects will never be freed up.