Design Patterns and Frameworks	**Exercise Sheet No. 3**
Dr.-Ing. Max Leuthäuser	Software Technology Group
INF 2081	Institute for SMT
`http://st.inf.tu-dresden.de/teaching/dpf`	Department of Computer Science
	Technische Universität Dresden
	01062 Dresden

Optimization and Control Flow Patterns

Task 3.1: Tetris Light

A Tetris field is essentially a big collection of stones, some representing empty fields, and some representing parts of a Tetris block, either just falling or landed at the bottom of the play ground some time ago. Whenever a block lands, it dissembles into its individual stones which are considered independent from then on. It makes sense, therefore, to represent each stone by an individual object in an object-oriented Tetris application. Doing so naïvely results in a large amount of objects being allocated, deleted, and moved all the time. What design pattern can be applied to optimize this, taking advantage of the fact that the stones are really very similar?

Task 3.2: The One and Only Tetris

2a)

Imagine a design with a factory for creating the stones in a Tetris application. It makes sense to restrict the number of instances of such a factory that can exist in an application at any one time to at most one. This way we can easily ensure that every part of the application uses the same factory.

Which design pattern can we use for this? What does the corresponding code look like?

2b)

An important variation point for the Factory pattern is the specific factory class. We have used this in a prior task to provide for different maze configurations. How can we maintain this variability while also ensuring that no more than one instance of the factory will be used?

Task 3.3: Commanding Tetris

In a Tetris application many different things may happen at any one time: the current stone falls, new stones are generated, user input causes the current stone to move and turn, etc. One way to deal with this multitude of events is by using the Command pattern and providing a central loop which takes commands from a queue and executes them.

3a)

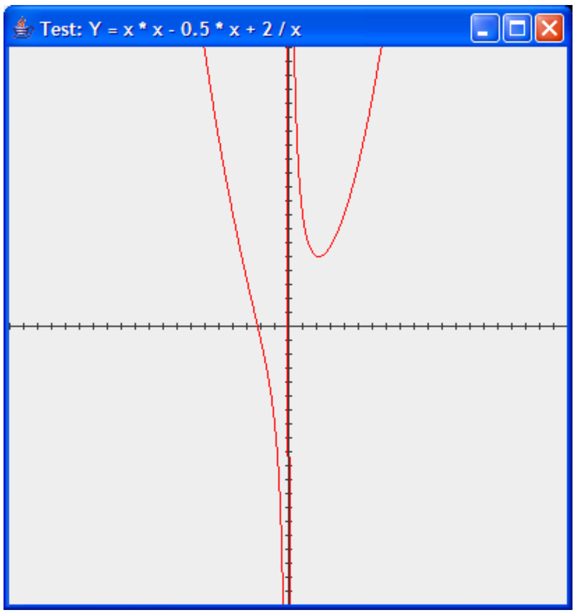
What is the Command pattern and how is it structured?

3b)

Design a Tetris control loop using the COMMAND pattern.

Task 3.4: Function Plotter

Design a Swing-based function plotter component that plots the curve corresponding to an expression in a variable ‘x’ in a range of -10.0 to $+10.0$ (the following figure shows a screen shot of such a component in action).



Use the INTERPRETER pattern to make the function expression a parameter of the new component.

$\boxed{\text{4a)}}$

What is the INTERPRETER pattern? What is its structure?

$\boxed{\text{4b)}}$

Develop a design for the new component using the INTERPRETER design pattern. What additional technology do you need to make this system functional?

Hint: To redefine the drawing behaviour of a swing component override `JComponent`’s `paintComponent()` operation and use the operations provided by the `Graphics` parameter so manipulate the output.