

COS782 Assignment 2: Generic version of the Strategy design pattern

Daniël van Zyl & Lukas Anthonissen
u21446459 & u21434345

June 2024

1 Introduction

The Strategy design pattern is a behavioural pattern that enables an object to change its behaviour based on the current internal state of the object. It achieves this by defining and encapsulating a family of algorithms and allowing them to be interchanged based on the internal state of the object. The strategy pattern has three participants namely: Context, Strategy and ConcreteStrategy. The *Context* hold a reference to a strategy object, the *Strategy* define the interface for the *ConcreteStrategy* objects and the *ConcreteStrategies* implement the algorithm defined by the *Strategy*. To make a generic version of the strategy pattern, we decided to use the idea of functors to implement the concrete strategies.

2 Generic Strategy

2.1 Functor

2.1.1 Template Type Definitions

The functor is defined with a class type defining the *ReturnType* of the Functor, as well as a list of class types defining the *ParameterTypes* of the Functor.

2.1.2 Private Members

- **func_:** The saved function of type *ReturnType(ParameterTypes...)*

2.1.3 Public Members

- **Constructor:** The constructor of the function receives a function of type *ReturnType(ParameterTypes...)* and assigns it to the private *func_* member.

- **ReturnType operator():** Returns the result of executing the *func_* member.

2.2 Strategy

2.2.1 Template Type Definitions

The functor is defined with a class type defining the *Key* of the strategy map, a class type defining the *StrategyErrorPolicy* and the *ReturnType* and *ParameterTypes* of the Functors used as strategies.

2.2.2 Private Members

- **FunctorType:** Is a type defined as a Functor with the return type of *ReturnType* and parameter types of *ParameterTypes*.
- **strategies:** A map containing key-value pairs of unique keys and their respective strategies in the form of Functors.
- **current_strategy:** The key of the current selected strategy.

2.2.3 Public Members

- **Constructor(Key initial_key, FunctorType initial_strategy):** The constructor of the function receives a key-value pair of types *Key* and *FunctorType* respectively, adds them to the strategy map and sets the *current_strategy* to the key.
- **void register_strategy(Key key, FunctorType strategy):** Receives a key-value pair of types *Key* and *FunctorType* respectively and adds them to the strategy map.
- **bool remove_strategy(Key key):** Receives a *Key* whose entry must be removed from the strategy map. If the entry is the last one, nothing is removed and false is returned, else the entry is removed and if the current key is equal to the *Key* parameter, the current strategy is set to the first strategy in the map.
- **void set_strategy(Key key):** Receives a key to set as the current strategy. If no entry exists in the strategy map with the same key, a *StrategyErrorPolicy* exception is thrown.
- **ReturnType operator()(ParameterTypes... args):** Returns the executed value of the currently selected strategy.
- **ReturnType execute()(ParameterTypes... args):** Returns the executed value of the currently selected strategy.

- **ReturnType execute_by_key()(Key key, ParameterTypes... args):** Returns the executed value of the strategy with the same key as passed in the parameters. If not such entry exists, return the executed value of the first entry in the map.

2.3 Error Policy

2.3.1 Template Type Definitions

The error policy allow the user to add custom generic exceptions that can be thrown when an error occur in the strategy.

2.3.2 Private Members

- **message:** The message that will be displayed when the exception is thrown of type *char**.
- **key:** The key of the strategy that threw the exception of type *Key**.

2.3.3 Public Members

- **Default Constructor:** Set the message to a default value.
- **Constructor(const char *message):** Set the message member variable to the passed in parameter.
- **getKey():** Returns the key member variable.
- **what():** Print out the message when the exception get thrown.

2.3.4 Protected Members

- **OnNoStrategy(const Key& key):** The function that handles the error. It set the key member variable and it throws an exception with a custom message.