КУРС **«**Шаблоны проектирования**».** Контрольная работа№ **1**.

Этап **1**. **Архитектура**. Опишите интерфейс IPrinter (печатающий) – содержит методы

печати строковых данных. Реализуйте интерфейс IPrinter в классе PrinterDefault,

который осуществляет вывод строковых данных на консоль. Опишите интерфейс

IPrintable (печатаемый) – содержит единственный метод Print (распечатать), который на

вход получает объект IPrinter. Реализуйте интерфейс IPrintable в классах Word (слово) и

Sign (символ). Класс Word – инкапсулирует строковую переменную (тип string,

инициируется в конструкторе); использует объект IPrinter для печати значения строковой

переменной. Класс Sign – инкапсулирует символьную переменную (тип char,

инициируется в конструкторе); использует объект IPrinter для печати значения

символьной переменной. Создать класс Text (текст), который инкапсулирует массив

печатаемых объектов IPrintable (инициируется в конструкторе). Реализовать в классе

Text интерфейс IPrintable – вызовы перенаправляются элементам массива.

Этап **2**. **Клиентский код**. Создайте объект класса Text, передав в конструкторе массив

объектов классов Word и Sign. Посредством объекта класса PrinterDefault организуйте

вывод на консоль данных, инкапсулированных в объектах классов Word и Sign. Смотри

пример клиентского кода (язык C#):

Text txt = new Text(

new Word("Тестируем"), new Sign(" "),

new Word("мою"), new Sign(" "),

new Word("архитектуру"), new Sign("!"));

txt.Print(new PrinterDefault());

Результат работы кода – на консоль отправлен текст:

Тестируем мою архитектуру!

Этап **3**. **Модификация** **1**. Создайте класс PrinterSpecial как наследника PrinterDefault

так, чтобы данные, которые инкапсулированы в объектах класса Word, выводились на

консоль в круглых скобках. Смотри пример клиентского кода (язык C#):

txt.Print(new PrinterSpecial());

Результат работы кода – на консоль отправлен текст:

(Тестируем) (мою) (архитектуру)!

Этап **4**. **Модификация** **2**. Интерфейс IPrinterDelegate наследуйте от интерфейса

IPrinter. Дополните интерфейс IPrinterDelegate методом, который позволит

классифицировать отношения между объектами IPrinterDelegate и объектами IPrintable

как отношение делегирования (объекты IPrinterDelegate – делегаты, объекты IPrintable –

уполномоченные). Создайте пример конкретного класса семейства IPrinterDelegate и

приведите пример клиентского кода.

Этап **5**. **Реинжиниринг**. По коду постройте UML-диаграмму классов архитектуры

созданной программы. При помощи UML-диаграммы последовательностей покажите

динамику взаимодействий объекта-делегата (объекты IPrinterDelegate) и объекта-

уполномоченного (объекты IPrintable).

СИСТЕМА ОЦЕНОК. За каждый выполненный этап +1 балл.