# Дисциплина «Паттерны проектирования»

Дисциплина рекомендована для направлений: 01.03.02 Прикладная математика и информатика, 02.03.02 Фундаментальная информатика и информационные технологии, 02.03.03 Математическое обеспечение и администрирование информационных систем, 09.03.03 Прикладная информатика. Курс – 3–4. Общее число часов: 72 ч.

Целью дисциплины является формирование современных теоретических знаний, приобретение умений и навыков, позволяющих владеть на практике основными шаблонами проектирования – типичными способами решения часто встречающихся проблем при проектировании программ.

Задачи курса:

* приобретение теоретических знаний в области разработки шаблонов проектирования;
* приобретение практических навыков объектно-ориентированного проектирования прикладных задач с использованием шаблонов проектирования;

# Место дисциплины в структуре образовательной программы

Дисциплина «Паттерны проектирования» является логическим продолжением дисциплины «Проектирование Информационных Систем». Входными знаниями для освоения данной дисциплины являются знания, умения и опыт, накопленный студентами в процессе изучения дисциплины «Введение в ООП» и «Проектирование Информационных Систем».

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование раздела | всего | Количество часов | | |
|
| Лекции | Лабораторные занятия | СРС |
| 1 | Введение в Паттерны | 3 | 3 |  |  |
| 2 | Основные идеи паттернов | 5 | 2 | 2 | 1 |
| 3 | Порождающие паттерны | 9 | 3 | 3 | 3 |
| 4 | Структурные паттерны | 21 | 7 | 7 | 7 |
| 5 | Поведенческие паттерны | 30 | 10 | 10 | 10 |
| 6 | Системные шаблоны | 4 | 3 | 1 |  |
|  | Итого по дисциплине | **72** | 28 | 23 | 21 |

Умение студента выполнить и защитить практическую работу демонстрирует владение следующих компетенций:

**ОК 1** Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

**ОК 2** Организовывать собственную деятельность, определять методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

**ОК 5** Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности.

**ОК 7** Ставить цели, мотивировать деятельность подчиненных, организовывать и контролировать их работу с принятием на себя ответственности за результат выполнения заданий.

**ОК 8** Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

**ПК 1.2** Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты знаний:

1. Знать:

* принципы построения обобщенных моделей проектирования программного кода;
* основные шаблоны проектирования.

1. Уметь:

* проектировать систему классов для заданной предметной области с использованием шаблонов проектирования;
* адаптировать шаблоны проектирования к условиям постановки задачи в новой для себя предметной области.

1. Владеть:

* основными программными библиотеками, построенными на основе методологии шаблонов проектирования.

# Содержание дисциплины

**Тема 1. Введение в паттерны**

Понятие паттерна проектирования

Описание паттернов проектирования

**Тема 2. Основные идеи паттернов**

Классификация паттернов проектирования

**Тема 3. Порождающие паттерны**

Паттерн Абстрактная фабрика

Паттерн Строитель

Паттерн Фабричный метод

Паттерн Отложенная инициализация

Паттерн Мультитон

Паттерн Объектный пул

Паттерн Получение ресурса есть инициализация

Паттерн Прототип

Паттерн Одиночка

**Тема 4. Структурные паттерны**

Паттерн Адаптер

Паттерн Мост

Паттерн Компоновщик

Паттерн Декоратор

Паттерн Фасад

Паттерн Единая точка входа

Паттерн Приспособленец

Паттерн Заместитель

**Тема 5. Паттерны поведения**

Паттерн Цепочка обязанностей

Паттерн Команда

Паттерн Интерпретатор

Паттерн Итератор

Паттерн Посредник

Паттерн Наблюдатель

**Тема 6. Системные шаблоны**

# Фонд оценочных средств для проведения текущего контроля

# Перечень задач текущего контроля по темам:

Задачи по теме «Основные идеи паттернов»

1. История появления шаблонов;
2. Плюсы и минусы шаблонного подхода к проектированию и программированию;
3. Применение одного из основных шаблонов на практике:
   * Делегирование;
   * Функциональный дизайн;
   * Интерфейс;
   * Неизменяемый интерфейс;
   * Интерфейс–маркер;
   * Контейнер свойств;
   * Канал событий.

Задачи по теме «Порождающие паттерны»

1. Назначение порождающих паттернов;
2. Применение трех паттернов на практике:
   * Абстрактная фабрика;
   * Строитель;
   * Фабричный метод;
   * Отложенная инициализация;
   * Мультитрон;
   * Объектный пул;
   * Прототип;
   * Получение ресурса есть инициализация;
   * Одиночка.

Задачи по теме «Структурные паттерны»

1. Назначение структурных паттернов;
2. Применение трех паттернов на практике:
   * Адаптер;
   * Мост;
   * Компоновщик;
   * Декоратор;
   * Фасад;
   * Единая точка входа;
   * Приспособленец;
   * Заместитель.

Задачи по теме «Поведенческие паттерны»

1. Назначение поведенческих паттернов;
2. Применение семи паттернов на практике:
   * Цепочка обязанностей;
   * Команда (Транзакция);
   * Интерпретатор;
   * Итератор;
   * Посредник;
   * Хранитель;
   * Null Object;
   * Наблюдатель (издатель – подписчик);
   * Слуга;
   * Спецификация;
   * Состояние;
   * Стратегия;
   * Шаблонный метод;
   * Посетитель;
   * Простая политика;
   * Слушатель событий;
   * Одноразовый посетитель;
   * Иерархический посетитель.

# Список контрольных вопросов

1. Понятие паттерна проектирования
2. Описание паттернов проектирования
3. Классификация паттернов проектирования
4. Паттерн Абстрактная фабрика
5. Паттерн Строитель
6. Паттерн Фабричный метод
7. Паттерн Отложенная инициализация
8. Паттерн Мультитон
9. Паттерн Объектный пул
10. Паттерн Получение ресурса есть инициализация
11. Паттерн Прототип
12. Паттерн Одиночка
13. Паттерн Адаптер
14. Паттерн Мост
15. Паттерн Компоновщик
16. Паттерн Декоратор
17. Паттерн Фасад
18. Паттерн Единая точка входа
19. Паттерн Приспособленец
20. Паттерн Заместитель
21. Паттерн Цепочка обязанностей
22. Паттерн Команда
23. Паттерн Интерпретатор
24. Паттерн Итератор
25. Паттерн Посредник
26. Паттерн Наблюдатель

# Перечень упражнений для самостоятельной работы

Привести пример реализации паттерна проектирования и описать процесс его создания:

1. Паттерн Абстрактная фабрика
2. Паттерн Строитель
3. Паттерн Фабричный метод
4. Паттерн Отложенная инициализация
5. Паттерн Мультитрон
6. Паттерн Объектный пул
7. Паттерн Получение ресурса есть инициализация
8. Паттерн Прототип
9. Паттерн Одиночка
10. Паттерн Адаптер
11. Паттерн Мост
12. Паттерн Компоновщик
13. Паттерн Декоратор
14. Паттерн Фасад
15. Паттерн Единая точка входа
16. Паттерн Приспособленец
17. Паттерн Заместитель
18. Паттерн Цепочка обязанностей
19. Паттерн Команда
20. Паттерн Интерпретатор
21. Паттерн Итератор
22. Паттерн Посредник
23. Паттерн Наблюдатель

# Пример выполнения упражнения

Создаем класс Адаптера.

В этом классе реализуем метод, который принимает на входе объект в незнакомом формате, а возвращает объект нужного нам формата.

В этом методе реализуем логику преобразования объекта одного формата в другой.

Пример реализации на языке Java:

interface Chief {

Object makeBreakfast();

Object makeDinner();

}

public class Plumber { // Adaptee

public Object getScrewNut() { ... }

public Object getGasket() { ... }

}

public class ChiefAdapter extends Plumber implements Chief { // Adapter

public Object makeBreakfast() {

return getGasket();

}

public Object makeDinner() {

return getScrewNut();

}

}

public class Client { // Client

public static void eat(Object dish) { ... }

public static void main(String[] args) {

Chief ch = new ChiefAdapter();

eat(ch.makeBreakfast());

eat(ch.makeDinner());

}

}

# Комплекс тестовых заданий для проверки знаний, полученных в процессе изучения курса

|  |  |  |
| --- | --- | --- |
|  | Описание | Ответ |
| 1 | Выберите верное утверждение:  А) Мост относится к порождающим паттернам  Б) Приспособленец относится к структурным паттернам  В) Команда относится к структурным паттернам  Г) Прототип относится к структурным паттернам | Б |
| 2 | Выберите верное утверждение:  А) Фабричный метод относится к порождающим паттернам  Б) Прототип относится к структурным паттернам  В) Стратегия относится к структурным паттернам  Г) Декоратор относится к поведенческим паттернам | А |
| 3 | Что такое агрегация?  А) Нет правильного ответа  Б) Изменение в одной сущности может влиять на состояние или поведение другой сущности  В) Отношение наследования или реализации интерфейса  Г) описывает связь «часть»–«целое», в котором «часть» может существовать отдельно от «целого» | Г |
| 4 | К какому из видов относится шаблон «Мост»?  А) Порождающие  Б) Структурные  В) Базовые  Г) Поведенческие | Б |
| 5 | Позволяет объекту изменять своё поведение в зависимости от внутреннего состояния. О каком паттерне идёт речь?  А) Состояние  Б) Стратегия  В) Декоратор  Г) Абстрактная фабрика | В |
| 6 | Предоставляет способ последовательного доступа к элементам множества, независимо от его внутреннего устройства. О каком паттерне идёт речь?  А) Итератор  Б) Строитель  В) Интерпретатор  Г) Прототип | А |
| 7 | К какому из видов относится шаблон «Итератор»?  А) Порождающие  Б) Структурные  В) Поведенческие  Г) Базовые | В |
| 8 | Помещает объекты в древовидную структуру, представляя их в виде иерархии. Позволяет клиенту одинаково обращаться как к отдельному объекту, так и к целому поддереву. О каком паттерне идёт речь?  А) Адаптер  Б) Компоновщик  В) Строитель  Г) Мост | Б |
| 9 | Какое из описаний подходит паттерну «Прокси»?  А) Предоставляет замену другого объекта для контроля доступа к нему  Б) Определяет алгоритм, некоторые этапы которого делегируются подклассам  В) Позволяет подклассам переопределить эти этапы, не меняя структуру алгоритма  Г) Обеспечивает слабую связь, избавляя объекты от необходимости прямо ссылаться друг на друга | А |
| 10 | Разделяет создание сложного объекта и инициализацию его состояния так, что одинаковый процесс построения может создать объекты с разным состоянием. О каком паттерне идёт речь?  А) Мост  Б) Компоновщик  В) Адаптер  Г) Строитель | Г |
| 11 | Предоставляет интерфейс для создания групп связанных или зависимых объектов, не указывая их конкретный класс. О каком паттерне идёт речь?  А) Декоратор  Б) Состояние  В) Стратегия  Г) Абстрактная фабрика | Г |
| 12 | Что из этого описывает поведенческие шаблоны?  А) Правильного ответа нет  Б) Изменяют интерфейс уже существующих объектов или его реализацию  В) Шаблоны проектирования, которые абстрагируют процесс инстанцирования  Г) Определяют взаимодействие между объектами, увеличивая таким образом его гибкость | Г |
| 13 | Гарантирует, что класс имеет только один экземпляр и представляет глобальную точку доступа к нему. О каком паттерне идёт речь?  А) Фасад  Б) Одиночка  В) Приспособленец  Г) Фабричный метод | Б |
| 14 | Что из этого описывает порождающие шаблоны?  А) Шаблоны проектирования, которые абстрагируют процесс инстанцирования  Б) Определяют взаимодействие между объектами, увеличивая таким образом его гибкость  В) Изменяют интерфейс уже существующих объектов или его реализацию  Г) Правильного ответа нет | А |
| 15 | Предоставляет единый интерфейс к группе интерфейсов подсистемы. Определяет высокоуровневый интерфейс, делая подсистему проще для использования. О каком паттерне идёт речь?  А) Одиночка  Б) Фабричный метод  В) Фасад  Г) Приспособленец | В |
| 16 | Выберите верное утверждение  А) Мост относится к порождающим паттернам  Б) Компоновщик относится к порождающим паттернам  В) Декоратор относится к порождающим паттернам  Г) Строитель относится к порождающим паттернам | Г |
| 17 | Что такое композиция?  А) Отношение наследования или реализации интерфейса  Б) Подвид агрегации, в которой «части» не могут существовать отдельно от «целого»  В) Описывает связь «часть»–«целое», в котором «часть» может существовать отдельно от «целого»  Г) Изменение в одной сущности может влиять на состояние или поведение другой сущности | Б |
| 18 | Какие из перечисленных паттернов относятся к структурным?  А) Цепочка обязанностей, Хранитель  Б) Компоновщик, Декоратор  В) Строитель, Абстрактная фабрика  Г) Прототип, Хранитель | Б |
| 19 | К какому из видов относится шаблон «Посетитель»?  А) Базовые  Б) Структурные  В) Поведенческие  Г) Порождающие | В |
| 20 | К какому из видов относится шаблон «Прототип»?  А) Базовые  Б) Порождающие  В) Структурные  Г) Поведенческие | Б |

Общее число заданий: 20

Проходной порог: 60%

# Оценивание работы студента в течение семестра

|  |  |
| --- | --- |
| Описание | Баллы |
| Выполнены в срок необходимые задачи по теме, студент может пояснить причину применения того или иного паттерна в своей задаче, может предложить паттерн для выбранной преподавателем задачи, объяснить свой выбор. | 5+  (+1 за каждую доп. задачу) |
| Выполнено в срок 75% задач по теме или 100% задач не в срок, студент может пояснить причину применения того или иного паттерна в своей задаче. | 4+  (+1 за каждую доп. задачу) |
| Выполнено в срок 50% задач по теме или 75% задач не в срок. | 2 |
| Выполнено менее 50% задач по теме. | 0 |

|  |  |
| --- | --- |
| Оценка | |
| Незачет | Зачет |
| Студент набрал менее 13 баллов. | Студент набрал 14 – 20 баллов. |

# Помощь студенту

Список рекомендованной литературы:

* *Зандстра М.* PHP. Объекты, шаблоны и методики программирования. — 5-е изд.. — СПб.: «Диалектика», 2019. — С. 736. — ISBN 978-5-907144-54-5.
* *Фаулер, Мартин.* Рефакторинг кода на JavaScript: улучшение проекта существующего кода. — 2-е изд.. — СПб.: «Диалектика», 2019. — С. 464. — ISBN 978-5-907144-59-0.
* *Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.* Приемы объектно-ориентированного проектирования. Паттерны проектирования = PHP Objects, Patterns and Practice, Third Edition. — 3-е издание. — М.: «Вильямс», 2015. — С. 368. — ISBN 978-5-496-00389-6.
* *Джейсон Мак-Колм Смит.* Элементарные шаблоны проектирования = Elemental Design Patterns. — М.: «Вильямс», 2012. — 304 с. — ISBN 978-5-8459-1818-5.
* *Фаулер, Мартин, Бек, Кент, Брант, Джон, Опдайк, Уильям, Робертс, Дон.* Рефакторинг: улучшение проекта существующего кода. — М.: «Диалектика», 2019. — 448 с. — ISBN 978-5-9909445-1-0.
* *Мартин Фаулер.* Шаблоны корпоративных приложений (Signature Series) = Patterns of Enterprise Application Architecture (Addison-Wesley Signature Series). — М.: «Вильямс», 2012. — 544 с. — ISBN 978-5-8459-1611-2.
* *Марк Гранд.* Шаблоны проектирования в JAVA. Каталог популярных шаблонов проектирования, проиллюстрированных при помощи UML = Patterns in Java, Volume 1. A Catalog of Reusable Design Patterns Illustrated with UML. — М.: «Новое знание», 2004. — С. 560. — ISBN 5-94735-047-5.
* *Крэг Ларман.* Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. — М.: «[Вильямс](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1)», 2006. — С. 736. — [ISBN 0-13-148906-2](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/0131489062).
* *Стив Макконнелл.* Совершенный код = Code complete. — СПб.: Питер, 2005. — С. 896. — (Мастер-класс). — [ISBN 5-7502-0064-7](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/5750200647), 5-469-00822-3.

**Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины**

* CIT-Forum. Обзор паттернов проектирования - http://citforum.ru/SE/project/pattern/
* Source Making (eng) - <http://sourcemaking.com/>
* Вики ИТ Мехмат ЮФУ. Курс Паттерны проектирования - http://it.mmcs.sfedu.ru/wiki/Страница\_курса\_Паттерны\_проектирования
* Паттерны проектирования раздел на сайте Cpp-Reference - http://cpp-reference.ru/patterns/