

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский Нижегородский государственный университет  
им. Н.И. Лобачевского»**

**Институт информационных технологий, математики и механики**

---

УТВЕРЖДЕНО  
решением Ученого совета ННГУ  
протокол  
№ 13 от 30.11.2022 г.

**Рабочая программа дисциплины**

**Паттерны проектирования и реализации программного обеспечения**

---

Уровень высшего образования  
**магистратура**

---

Направление подготовки  
**09.04.03 Прикладная информатика**

---

Магистерская программа  
**Прикладная информатика в области принятия решений**

---

Форма обучения  
**очная**

---

Квалификация (степень)  
**Магистр**

---

Нижегород

2023

## 1. Место дисциплины в структуре ОПОП

Дисциплина относится к части, формируемой участниками образовательных отношений

№ варианта	Место дисциплины в учебном плане образовательной программы	Стандартный текст для автоматического заполнения в конструкторе РПД
2	Блок 1. Дисциплины (модули) Часть, формируемая участниками образовательных отношений	Дисциплина Б1.В.02 Паттерны проектирования и реализации программного обеспечения относится к части ООП направления подготовки 09.04.03 Прикладная информатика, формируемой участниками образовательных отношений.

## 2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции* (код, содержание индикатора)	Результаты обучения по дисциплине**	
УК-2. Способен управлять проектом на всех этапах его жизненного цикла	УК-2.1. Демонстрирует знание основных этапов жизненного цикла ИТ-проекта.	Знать типовые процессы создания, развертывания и ввода в эксплуатацию системных архитектурных решений, основанных на паттернах проектирования.	собеседование
	УК-2.2. Демонстрирует умение разрабатывать и анализировать альтернативные варианты планирования этапов проекта для достижения намеченных целей.	Уметь организовывать репозиторий с архитектурными артефактами, исходными кодами системных решений ИС. Уметь вводить в дизайн общего решения ИС новые системные решения, сопровождать их и адаптировать к контексту их использования. Уметь выполнять отладку и тестирования системных архитектурных решений.	собеседование
	УК-2.3. Демонстрирует наличие практического опыта принятия решений на различных этапах конкретных проектов.	Иметь практический опыт инсталляции и настройки: средств организации репозитория, средств разработки ПО. Владеть средствами контроля версий, компиляции и сборки исполняемых файлов ИС, отладки и тестирования. Иметь практический опыт создания, интеграции, адаптации и тестирования новых системных архитектурных решений.	задача
ПК-6. Способен управлять процессами и проектами по созданию (модификации) информации	ПК-6.1. Демонстрирует знание способов управления процессами и проектами по созданию (модификации) информационных ре-	Знать основные понятия, принципы и методы программирования ООП, языки программирования. Знать типовые решения (паттерны проектирования): фасад, адаптер, мост, компоновщик, итератор, деко-	собеседование

онных ресурсов (сайт, портал) Интернет.	сурсов (сайт, портал) Интернет.	ратор, стратегия, одиночка, прототип, абстрактная фабрика, фабричный метод, шаблонный метод, строитель, цепочка обязанностей, команда, хранитель, посетитель, интерпретатор, заместитель, приспособленец, состояние.	
	ПК-6.2. Демонстрирует умение планировать и организовывать разработку процессов и проектов по созданию (модификации) информационных ресурсов (сайт, портал), применять инструментальные средства.	Уметь выделять архитектурные задачи, формулировать требования к их решениям, описывать контракты и выделять интерфейсы, выбирать проектные решения на базе паттернов проектирования, адаптировать их специфике применения рамках конкретного контекста. Уметь документировать ключевые архитектурные решения средствами UML. Уметь воплощать архитектурные решения в коде и выполнять их отладку. Уметь оценивать и улучшать архитектурные решения с помощью техники рефакторинга.	собеседование
	ПК-6.3. Имеет практический опыт планирования и организации деятельности по созданию (модификации) информационных ресурсов (сайт, портал) Интернет.	Имеет опыт практической подготовки проектных решений и создания ИС на базе паттернов: фасад, адаптер, мост, компоновщик, итератор, декоратор, стратегия, одиночка, прототип, абстрактная фабрика, фабричный метод, шаблонный метод, посредник, цепочка обязанностей, команда, хранитель, посетитель, интерпретатор, заместитель, приспособленец, состояние.	задача

### 3. Структура и содержание дисциплины

#### 3.1. Трудоемкость дисциплины

	<b>очная форма обучения</b>
<b>Общая трудоемкость</b>	<b>5 ЗЕТ</b>
<b>Часов по учебному плану</b>	<b>180</b>
<b>в том числе</b>	
<b>контактная работа:</b>	<b>66</b>
- занятия лекционного типа	32
- занятия семинарского типа	32
- текущий контроль (КСР)	2
<b>самостоятельная работа</b>	<b>78</b>
<b>Промежуточная аттестация – экзамен</b>	<b>36</b>

### 3.2. Содержание дисциплины

Наименование и краткое содержание разделов и тем дисциплины	Всего (часы)	в том числе				
		контактная работа (работа во взаимодействии с преподавателем), часы				Самостоятельная работа обучающегося, часы
		из них				
		Занятия лекционного типа	Занятия семинарского типа	Занятия лабораторного типа	Всего контактных часов	
РАЗДЕЛ 1. Базовые паттерны проектирования						
Тема 1. Объектно-ориентированная парадигма	8	2	2		4	1
Тема 2. Введение в объектно-ориентированный анализ и базис языка визуального моделирования	16	4	4		8	7
Тема 3. Шаблонный метод проектирования	10	2	2		4	7
Тема 4. Шаблоны Фасад и Адаптер	10	2	2		4	7
Тема 5. Шаблон Мост	10	2	2		4	7
Тема 6. Шаблоны Компоновщик и Итератор	10	2	2		4	7
Тема 7. Шаблоны Декоратор и Стратегия	10	2	2		4	7
РАЗДЕЛ 2. Системные архитектурные решения						
Тема 8. Инстанцирование систем	13	3	3		9	7
Тема 9. Информационный обмен	13	3	3		9	7
Тема 10. Управление системой	13	3	3		9	7
Тема 11. Функциональное расширение систем	13	3	3		9	7
Тема 12. Целевое проектирование систем	16	4	4		12	7
Текущий контроль (КСР)	2				2	
Итого	108	32	32		98	78

Текущий контроль успеваемости реализуется в рамках занятий семинарского типа, контрольных работ и домашних заданий.

Промежуточная аттестация проходит в традиционной форме (экзамен).

#### 4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Для самоконтроля у студента имеется возможность изучения материала в дистанционном управляемом курсе (требуется авторизация): <https://e-learning.unn.ru/course/view.php?id=4381>

Самостоятельная работа студента при изучении дисциплины «Паттерны проектирования и реализации программного обеспечения» включает выполнение заданий под контролем преподавателя, решение домашних заданий и подготовку к экзамену.

Тематика самостоятельной работы

В рамках раздела 1 самостоятельная работа предполагает выполнение типовых заданий. Каждое типовое задание может быть сформулировано в контексте теоретической части тем раздела 1. Выполнение типовых задач связано с выделением архитектурных задач, формулировкой требований, описанием контрактов и интерфейсов, выбором проектных решений на базе паттернов проектирования, их адаптации к конкретному контексту, документировании средствами UML, реализации в коде, отладке, тестировании, применении техники рефакторинга для улучшения имеющихся решений, демонстрации принципов работы паттернов проектирования в рамках в клиентской части приложения. Контекст типовых заданий определяется теоретическим материалом тем 1 – 7 раздела 1.

Перечень типовых заданий:

Типовое задание 1. По описанию архитектуры построить программный код средствами объектно-ориентированного языка. Проверка задания.

Типовое задание 2. Для заданной архитектуры построить ее развитие так, чтобы обеспечить возможность выполнения заданного сценария на уровне клиентского кода средствами объектно-ориентированного языка. Проверка задания.

Типовое задание 3. Для заданной модели функционирования некоторой системы построить ее программную реализацию на основе методологии шаблонов проектирования. Проверка задания.

Типовое задание 4. По коду построить UML-диаграммы классов для основных элементов архитектуры. Проверка задания.

Типовое задание 5. По коду клиентской части построить UML-диаграммы последовательностей/объектов. Проверка задания.

Типовое задание 6. Для заданного шаблона проектирования построить UML-диаграмму классов. Проверка задания.

Типовое задание 7. Для заданного шаблона проектирования построить UML-диаграмму последовательностей, демонстрирующей принципы его работы. Проверка задания.

Типовая задание 8. Для заданного шаблона проектирования построить программный код, демонстрирующий принципы его работы. Проверка задания.

Типовое задание 9. В рамках заданной архитектуры для заданного сценария реализовать клиентский код средствами объектно-ориентированного языка. Проверка задания.

В рамках раздела 2 самостоятельная работа предполагает выполнение типовых заданий. Каждое типовое задание может быть сформулировано в контексте теоретической части тем раздела 2. Выполнение каждой задачи связано с инсталляцией, настройкой, работой со средствами: контроля версий, компиляцией, сборки, отладки и тестирования ИС. Суть типовых заданий заключается в интеграции в ИС новых системных архитектурных решений – теоретический материал тем 8 – 12 раздела 2. Перечень типовых заданий:

Типовая задание 10. Построить проект системного архитектурного решения с учетом применения в рамках конкретного контекста. Предоставить UML-диаграммы, документирующие ключевые архитектурные решения. Проверка задания.

Типовая задание 11. Реализовать заданный проект системного архитектурного решения в рамках конкретного контекста. На базе системного решения построить клиентский программный код, демонстрирующий принципы его работы. Проверка задания.

Типовая задание 12. Осуществить экспертизу качества системного архитектурного решения, выделить проблемы, разработать вариант их преодоления с помощью техники рефакторинга. Проверка задания.

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.2.

## 5. Фонд оценочных средств для промежуточной аттестации по дисциплине, включающий:

### 5.1. Описание шкал оценивания результатов обучения по дисциплине

Уровень сформированности компетенций (индикатора достижения компетенций)	Шкала оценивания сформированности компетенций						
	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	Не зачтено		Зачтено				
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибки.	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько незначительных ошибок	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок	Уровень знаний в объеме, превышающем программу подготовки
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания но не в полном объеме	Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами	Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами	Продemonстрированы все основные умения, решены все основные задачи с отдельными незначительными недочетами, выполнены все задания в полном объеме	Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов

<u>Навыки</u>	Отсутствие владения материалом. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами	Продemonстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продemonстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продemonстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продemonстрирован творческий подход к решению нестандартных задач
---------------	--	---	---	---	--	--	---

### Шкала оценки при промежуточной аттестации

Оценка		Уровень подготовки
Зачтено	Превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно»
	Отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
	Очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
	Хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»
	Удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
Не зачтено	Неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно», ни одна из компетенций не сформирована на уровне «плохо»
	Плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

## 5.2. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения

### 5.2.1. Контрольные вопросы

Вопросы	Код формируемой компетенции
1. Функциональная декомпозиция. Проблема обработки изменяющихся требований.	ПК-6
2. Объектно-ориентированная парадигма. Модели наследования и композиции.	ПК-6
3. Объектно-ориентированная парадигма. Инкапсуляция в механизме полиморфизма.	ПК-6

4. Объектно-ориентированная парадигма. Модели делегирования.	ПК-6
5. Абстрактный класс и интерфейс. Реализации класса и интерфейса. Проектирование на уровне интерфейсов.	ПК-6
6. Сложность систем. Объектная модель. Классификация. Идентификация классов и объектов. Ключевые абстракции и механизмы.	ПК-6
7. Основные этапы разработки объектно-ориентированной системы.	ПК-6
8. UML. Определение. Назначение. Концептуальная модель.	ПК-6
9. Концептуальная модель языка UML. Сущности.	ПК-6
10. Концептуальная модель языка UML. Отношения.	ПК-6
11. Концептуальная модель языка UML. Диаграмма прецедентов и диаграммы взаимодействий.	ПК-6
12. Концептуальная модель языка UML. Статические и динамические модели программных систем на примере диаграмм классов и диаграмм объектов.	ПК-6
13. Шаблоны проектирования. Основные термины и понятия. Механизмы повторного использования. Система каталогизации шаблонов проектирования.	ПК-6
14. Методы проектирования, основанные на стратегиях и шаблонах проектирования.	ПК-6
15. Модели инкапсуляции. Стратегия «Инкапсуляции изменчивости» в анализе общности и изменчивости.	ПК-6
16. Стратегии «Композиция предпочтительней наследования» и «Одно правило, одно место».	ПК-6
17. Специфика работа со сложной системой с множеством интерфейсов. Шаблон Фасад (Facade).	ПК-6
18. Проблема совместимости интерфейсов. Шаблон Адаптер (Adapter).	ПК-6
19. Понятия абстракции и реализации. Механизм отделения абстракции от реализации. Шаблон Мост (Bridge).	ПК-6
20. Механизм компоновки объектов в плоские коллекции и иерархические структуры. Манипулирование составными объектами. Шаблон Компоновщик (Composite). Виды Компоновщика.	ПК-6
21. Составные объекты. Организация доступа к элементам составного объекта на базе шаблона Итератор (Iterator).	ПК-6
22. Робастность итераторов плоских коллекций и иерархических структур.	ПК-6
23. Динамическое расширение функциональности объектов. Шаблон Декоратор (Decorator) – как гибкая альтернатива порождению подклассов.	ПК-6
24. Инкапсуляция алгоритма в объект. Механизм «прозрачной» замены алгоритма. Шаблон Стратегия (Strategy).	ПК-6
25. Инстанцирование систем. Принципы. Идеология объекта-одиночки (Singleton) в системе объектов. Способы доступа к объекту-одиночке.	УК-2
26. Инстанцирование систем. Конфигурирование. Абстрактная Фабрика (Abstract Factory) и Инструментарий (Kit).	УК-2
27. Инстанцирование систем. Каркасы приложений. Применение решений Фабричного Метода (Factory Method) и Шаблонного Метода (Template Method).	УК-2
28. Инстанцирование систем. Клонирование объектов и систем объектов. Поверхностное и глубокое клонирование на базе Прототипа (Prototype).	УК-2
29. Инстанцирование систем. Организация процесса конструирования различных пред-	УК-2



ставлений сложного объекта на базе решения Строитель (Builder).	
30. Информационный обмен. Основополагающие принципы. Классификация моделей. Простейшие модели и модель на базе Посредника (Mediator).	УК-2
31. Информационный обмен. Модель доставки сообщения на базе решения Цепочка Обязанностей (Chain of Responsibility). Решение без менеджера и с менеджером. Проксирование сообщений.	УК-2
32. Информационный обмен. Широковещательные трансляции на базе шаблона Наблюдатель (Observer). Особенности реализации систем типа Субъект-Наблюдатель без менеджера и с менеджером.	УК-2
33. Информационный обмен. Объекты с функциональностью Субъекта и Наблюдателя. Проблема учета циклических связей (зависимостей). Решение без менеджера и с менеджером.	УК-2
34. Управление системой. Идеология представление команды (операции) в виде объекта. Манипулирование командами как объектами. Протоколирование команд.	УК-2
35. Управление системой. Идеология представление команды (операции) в виде объекта. Манипулирование командами как объектами. Организация макросов (составные команды) на базе шаблона Компоновщик (Composite).	УК-2
36. Управление системой. Менеджер команд и универсальные механизмы отката (отмены операций) на базе решений Команда (Command) и Хранитель (Memento).	УК-2
37. Функциональное расширение систем. Нарращивание функциональности отдельных объектов (классов) без изменения существующего кода на базе решений Декоратор (Decorator) и Стратегия (Strategy).	УК-2
38. Функциональное расширение систем. Двойная диспетчеризация. Динамическое определение новых функций для систем объектов без изменения существующего кода на базе решения Посетитель (Visitor).	УК-2
39. Целевое проектирование систем. Оптимизация по времени работы. Идеологии кэширования и отложенной реакции на событие. Объектно-ориентированная организация событийных систем на основе решения Заместитель (Proxy).	УК-2
40. Целевое проектирование систем. Оптимизация по памяти. Идеология разделения объекта и его состояния. Объектно-ориентированная организация систем с большим числом объектов на основе решения Приспособленец (Flyweight).	УК-2
41. Целевое проектирование систем. Функциональный дизайн. Представление грамматики языка и интерпретация предложений на базе шаблона Интерпретатор (Interpreter).	УК-2
42. Целевое проектирование систем. Автоматный (конечный автомат) дизайн. Идеология совмещения в одном объекта разных состояний на основе решения Состояние (State).	УК-2

## 5.2.2. Типовые тестовые задания для оценки сформированности компетенции

### Типовые тестовые задания для оценки сформированности компетенции ПК-6

- Метод функциональной декомпозиции позволяет подготовить код к изменениям.
  - Практически нет. (+)
  - Да, это стандартный метод проектирования сложных систем.
- Главные принципы ООП.
  - Концепция инкапсулированной сущности.
  - Концепция класса, механизмы наследования и полиморфизма.
  - Концепция метода, механизмы инкапсуляции, наследования и полиморфизма.

d) Концепция объекта, механизмы инкапсуляции, наследования и полиморфизма. (+)

3. Шаблоны проектирования это –

- a) технология проектирования систем, построенных на принципах ООП. (+)
- b) стандартная методология адаптация кода программ к последующим изменениям.
- c) база данных типовых проектных решений на все случаи жизни.

4. Техника выбора паттерна-кандидата на решения конкретной задачи заключается

- a) в анализе разделов «Назначения» и «Структура».
- b) в анализе разделов «Назначения» и «Результаты».
- c) в анализе разделов «Назначения» и «Применимость». (+)

5. Паттерн проектирования ФАСАД используется для

- a) для упрощения работы клиентского кода со сложной системой (+)
- b) для скрытия работы некоторой системы от клиентского кода.
- c) для создания удобных пользовательских интерфейсов.

6. Для предотвращения проблемы бесконтрольного роста числа классов в программной системе вы обратите внимание на какую группу паттернов:

- a) ОДИНОЧКА и КОМАНДА.
- b) ФАБРИКА и ПОСРЕДНИК.
- c) МОСТ и ДЕКОРАТОР. (+)

7. Для исключения дублирования методов в разных классах вы будете осуществлять рефакторинг, опираясь на какую группу паттернов:

- a) АДАПТЕР и ПРОТОТИП.
- b) ШАБЛОННЫЙ МЕТОД и СТРАТЕГИЯ. (+)
- c) КОМПОНОВЩИК и ИТЕРАТОР.

8. Для предельной экономии оперативной памяти какая группа паттернов привлечет ваше внимание:

- a) ПРИСПОСОБЛЕНЕЦ и АБСТРАКТНАЯ ФАБРИКА. (+)
- b) ПОСРЕДНИК и ПОСЕТИТЕЛЬ.
- c) КОМПОНОВЩИК и ИТЕРАТОР.

### **Типовые тестовые задания для оценки сформированности компетенции УК-2**

1. В стандартный механизм отмены операций (инкрементальная модель) вы интегрируете следующие решения:

- a) МОСТ, ФАСАД.
- b) КОМАНДА, ПРОТОТИП, ШАБЛОННЫЙ МЕТОД, ОДИНОЧКА. (+)
- c) ПАССИВНЫЙ НАБЛЮДАТЕЛЬ и техника разделения булевой переменной.

2. Механизм «ленивых» операций Вы скорее всего будете создавать на базе решений

- a) МОСТ, ФАСАД.
- b) КОМАНДА, ПРОТОТИП, ШАБЛОННЫЙ МЕТОД, ОДИНОЧКА.
- c) ПАССИВНЫЙ НАБЛЮДАТЕЛЬ и техника разделения булевой переменной. (+)

3. Декомпозицию архитектуры на слои Вы будете осуществлять при помощи решений

- a) МОСТ, ФАСАД. (+)

- б) КОМАНДА, ПРОТОТИП, ШАБЛОННЫЙ МЕТОД, ОДИНОЧКА.
- с) ПАССИВНЫЙ НАБЛЮДАТЕЛЬ и техника разделения булевой переменной.

4. Какая техника и основанное на ней решение позволяет создать дополнительный функционал за пределами имеющегося класса.

- а) ПРИСПОБОБЛЕНЕЦ и техника разделения объекта и его состояния.
- б) ПОСЕТИТЕЛЬ и техника двойной диспетчеризации. (+)
- с) ПАССИВНЫЙ НАБЛЮДАТЕЛЬ и техника разделения булевой переменной.

### 5.2.3. Типовые задачи для оценки сформированности компетенции

Текущий контроль усвоения материала проводится путем проведения контрольных работ во время лабораторных занятий, домашних заданий для самостоятельной работы и проверки выполнения домашних заданий. Ниже представлены примеры заданий контрольной работы.

**Задание 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`, иницируется в конструкторе); использует объект `IPrinter` для печати значения строковой переменной. Класс `Sign` – инкапсулирует символьную переменную (тип `char`, иницируется в конструкторе); использует объект `IPrinter` для печати значения символьной переменной. Создать класс `Text` (текст), который инкапсулирует массив печатаемых объектов `IPrintable` (иницируется в конструкторе). Реализовать в классе `Text` интерфейс `IPrintable` – вызовы перенаправляются элементам массива.

**Задание 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!**

**Задание 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)!**

**Задание 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` – уполномоченные (делегаты), объекты `IPrintable` – поручители). Создайте пример конкретного класса семейства `IPrinterDelegate` и приведите пример клиентского кода.

**Задание 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`).

## **6. Учебно-методическое и информационное обеспечение дисциплины**

а) Программное обеспечение и Интернет-ресурсы

1. Справочник по шаблонам проектирования. <http://www.oodesign.com/>
2. Архитектурное проектирование программного обеспечения. Бесплатный курс для самообразования на официальном сайте Интуит.ру. <http://www.intuit.ru/studies/courses/3509/751/info>
3. Объектно-ориентированный анализ и программирование. Бесплатный курс для самообразования на официальном сайте Интуит.ру. <http://www.intuit.ru/studies/courses/491/347/info>
4. Проектирование информационных систем. Бесплатный курс для самообразования на официальном сайте Интуит.ру. <http://www.intuit.ru/studies/courses/491/347/info>

## **7. Материально-техническое обеспечение дисциплины**

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой (лекционного и семинарского типа), оснащенные оборудованием и техническими средствами обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ 09.04.03 Прикладная информатика.

Автор            д.т.н., профессор Н.В. Старостин

Рецензент      д.т.н., профессор Ю.С. Федосенко

Заведующий кафедрой            М.Х. Прилуцкий

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики

30.11.2022 года, протокол № 3