

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 № 11 от 25.12.2024 г.

Рабочая программа дисциплины

Шаблоны проектирования

Уровень высшего образования

Магистратура

Направление подготовки / специальность

02.04.01 - Математика и компьютерные науки

Направленность образовательной программы

Математика и компьютерные науки

Форма обучения

очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.О.10 Шаблоны проектирования относится к обязательной части образовательной программы.

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	Для текущего контроля успеваемости	Для промежуточной аттестации
ОПК-3: Способен самостоятельно создавать прикладные программные средства на основе современных информационных технологий и сетевых ресурсов, в том числе отечественного производства	ОПК-3.1: Обладает фундаментальными знаниями в области прикладного программирования и информационных технологий ОПК-3.2: Умеет использовать их в профессиональной деятельности ОПК-3.3: Имеет практический опыт применения программных средств, используемых при построении математических моделей в естественных науках	ОПК-3.1: Знать методологии разработки программного обеспечения ОПК-3.2: Уметь использовать методологии разработки программного обеспечения и использование их в профессиональной деятельности ОПК-3.3: Иметь практический опыт профессиональной разработки программного обеспечения и использование их в профессиональной деятельности	Задачи Практическая задача Практическое задание	Экзамен: Контрольные вопросы

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	5
Часов по учебному плану	180
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	32
- занятия семинарского типа (практические занятия / лабораторные работы)	64

- КСР	2
самостоятельная работа	46
Промежуточная аттестация	36 Экзамен

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
	о ф о	о ф о	о ф о	о ф о	о ф о
Тема 1. Парадигмы проектирования и реализации ПО	16	4	8	12	4
Тема 2. Средства моделирования и проектирования ПО	16	4	8	12	4
Тема 3. Базовые техники проектирования и типовые решения	18	4	8	12	6
Тема 4. Разработка решений, связанных с инстанцированием сложных систем	18	4	8	12	6
Тема 5. Создание механизмов информационного обмена между объектами	18	4	8	12	6
Тема 6. Разработка высокоуровневых управляемых систем	18	4	8	12	6
Тема 7. Техники функционального расширения систем с минимальными изменениями	18	4	8	12	6
Тема 8. Проектирование с элементами оптимизации	20	4	8	12	8
Аттестация	36				
КСР	2			2	
Итого	180	32	64	98	46

Содержание разделов и тем дисциплины

РАЗДЕЛ 1. Паттерны проектирования и реализации программного обеспечения

Тема 1. Парадигмы проектирования и реализации ПО

Тема 2. Средства моделирования и проектирования ПО

Тема 3. Базовые техники проектирования и типовые решения

Тема 4. Разработка решений, связанных с инстанцированием сложных систем

Тема 5. Создание механизмов информационного обмена между объектами

Тема 6. Разработка высокоуровневых управляемых систем

Тема 7. Техники функционального расширения систем с минимальными изменениями

Тема 8. Проектирование с элементами оптимизации

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Для обеспечения самостоятельной работы обучающихся используются:
Электронные курсы, созданные в системе электронного обучения ННГУ:

Шаблоны проектирования ПО, <https://e-learning.unn.ru/course/view.php?id=4382>.

Иные учебно-методические материалы:

Самостоятельная работа студентов подразделяется на следующие категории:

- Изучение лекционных материалов и учебной литературы (см. п. 6).
- Выполнение домашних практических заданий.

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ОПК-3:

Задача 1. Для заданной модели функционирования некоторой системы построить ее программную реализацию на основе методологии шаблонов проектирования.

Задача 2. По описанию архитектуры построить программный код средствами объектно-ориентированного языка.

Задача 3. Для заданной архитектуры построить ее развитие так, чтобы обеспечить возможность выполнения заданного сценария на уровне клиентского кода средствами объектно-ориентированного языка.

Задача 4. Для заданной модели функционирования некоторой системы построить ее программную реализацию на основе методологии шаблонов проектирования.

Задача 5. По коду построить UML-диаграммы классов для основных элементов архитектуры.

Критерии оценивания (оценочное средство - Задачи)

Оценка	Критерии оценивания
зачтено	Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Результаты работы представлены преподавателю в срок.
не	Выполнены не все практические задания или выполнены не в полном объеме (представлено

Оценка	Критерии оценивания
зачтено	не полное описание этапов выполнения заданий, получен неверный ответ, результаты работы не представлены преподавателю).

5.1.2 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-3:

Задание 1. 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 2. 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 3. 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!

Критерии оценивания (оценочное средство - Практическое задание)

Оценка	Критерии оценивания
зачтено	Выполнена основная часть задания, возможно с незначительными недочетами
не зачтено	Выполнено менее половины задания, есть существенные недочеты

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения 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 ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	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 несущественными 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 базовых навыков. Невозможность оценить наличие навыков вследствие	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые	Имеется минимальный набор навыков для решения стандартных	Продemonстрированы базовые навыки при решении стандартных задач с	Продemonстрированы базовые навыки при решении стандартных задач без	Продemonстрированы навыки при решении нестандартных задач	Продemonстрирован творческий подход к решению нестандартных задач

	отказа обучающегося от ответа	ошибки	х задач с некоторым и недочетами	некоторым и недочетами	ошибок и недочетов	без ошибок и недочетов	
--	-------------------------------	--------	----------------------------------	------------------------	--------------------	------------------------	--

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «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».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ОПК-3

1. Функциональная декомпозиция. Проблема обработки изменяющихся требований.
2. Объектно-ориентированная парадигма. Модели наследования и композиции.
3. Объектно-ориентированная парадигма. Инкапсуляция в механизме полиморфизма.
4. Объектно-ориентированная парадигма. Модели делегирования.
5. Абстрактный класс и интерфейс. Реализации класса и интерфейса. Проектирование на уровне интерфейсов.
6. Сложность систем. Объектная модель. Классификация. Идентификация классов и объектов. Ключевые абстракции и механизмы.

7. Основные этапы разработки объектно-ориентированной системы.
8. UML. Определение. Назначение. Концептуальная модель.
9. Концептуальная модель языка UML. Сущности.
10. Концептуальная модель языка UML. Отношения.
11. Концептуальная модель языка UML. Диаграмма прецедентов и диаграммы взаимодействий.
12. Концептуальная модель языка UML. Статические и динамические модели программных систем на примере диаграмм классов и диаграмм объектов.
13. Паттерны проектирования. Основные термины и понятия. Механизмы повторного использования. Система каталогизации шаблонов проектирования.
14. Методы проектирования, основанные на стратегиях и шаблонах проектирования.
15. Модели инкапсуляции. Стратегия «Инкапсуляции изменчивости» в анализе общности и изменчивости.
16. Стратегии «Композиция предпочтительней наследования» и «Одно правило, одно место».
17. Специфика работа со сложной системой с множеством интерфейсов. Шаблон Фасад (Facade).
18. Проблема совместимости интерфейсов. Шаблон Адаптер (Adapter).
19. Понятия абстракции и реализации. Механизм отделения абстракции от реализации. Шаблон Мост (Bridge).
20. Механизм компоновки объектов в плоские коллекции и иерархические структуры. Манипулирование составными объектами. Шаблон Компоновщик (Composite). Виды Компоновщика.
21. Составные объекты. Организация доступа к элементам составного объекта на базе шаблона Итератор (Iterator).
22. Робастность итераторов плоских коллекций и иерархических структур.
23. Динамическое расширение функциональности объектов. Шаблон Декоратор (Decorator) – как гибкая альтернатива порождению подклассов.
24. Инкапсуляция алгоритма в объект. Механизм «прозрачной» замены алгоритма. Шаблон Стратегия (Strategy).
25. Принципы инстанцирования объектов и систем. Идеология объекта-одиночки (Singleton) в системе объектов. Способы доступа к объекту-одиночке.
26. Конфигурирование и инстанцирование систем объектов. Абстрактная Фабрика (Abstract Factory) и Инструментарий (Kit).

27. Применение решений Фабричного Метода (Factory Method) и Шаблонного Метода (Template Method) в конструировании каркасов приложений с использованием.
28. Клонирование объектов и систем объектов. Поверхностное и глубокое клонирование на базе Прототипа (Prototype).
29. Организация процесса конструирования различных представлений сложного объекта на базе решения Строитель (Builder).
30. Информационный обмен между объектами. Основопологающие принципы. Классификация моделей. Простейшие модели и модель на базе Посредника (Mediator).
31. Модель доставки сообщения на базе решения Цепочка Обязанностей (Chain of Responsibility). Решение без менеджера и с менеджером. Проксирование сообщений.
32. Широковещательные трансляции на базе шаблона Наблюдатель (Observer). Особенности реализации систем типа Субъект-Наблюдатель без менеджера и с менеджером.
33. Объекты с функциональностью Субъекта и Наблюдателя. Проблема учета циклических связей (зависимостей). Решение без менеджера и с менеджером.
34. Идеология представление команды (операции) в виде объекта. Манипулирование командами как объектами. Протоколирование команд.
35. Идеология представление команды (операции) в виде объекта. Манипулирование командами как объектами. Организация макросов (составные команды) на базе шаблона Компонент (Composite).
36. Менеджер команд и универсальные механизмы отката (отмены операций) на базе решений Команда (Command) и Хранитель (Memento).
37. Нарращивание функциональности отдельных объектов (классов) без изменения существующего кода на базе решений Декоратор (Decorator) и Стратегия (Strategy).
38. Двойная диспетчеризация. Динамическое определение новых функций для систем объектов без изменения существующего кода на базе решения Посетитель (Visitor).
39. Представление грамматики языка и интерпретация предложений на базе шаблона Интерпретатор (Interpreter).
40. Планирование вычислительных ресурсов. Идеологии кэширования и отложенной реакции на событие. Объектно-ориентированная организация событийных систем на основе решения Заместитель (Proxy).
41. Идеология разделения объекта и его состояния. Объектно-ориентированная организация систем с большим числом объектов на основе решения Приспособленец (Flyweight).
42. Идеология совмещения в одном объекта разных состояний на основе решения Состояние (State).

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
превосходно	Студент дал развернутый ответ на все вопросы и при этом продемонстрировал знание дополнительного материала.
отлично	Студент дал развернутый ответ на все вопросы.
очень хорошо	Студент дал ответ на все вопросы, возможно с незначительными недочетами.
хорошо	Студент ответил на большую часть вопросов с незначительными недочетами.
удовлетворительно	Студент ответил на большую часть вопросов с существенными недочетами.
неудовлетворительно	При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач.
плохо	Отсутствие знаний материала, отсутствует способность решения стандартных задач.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Лаврищева Екатерина Михайловна. Программная инженерия. Парадигмы, технологии и CASE-средства : Учебник для вузов / Лаврищева Е. М. - 2-е изд. - Москва : Юрайт, 2018. - 280 с. - (Высшее образование). - ISBN 978-5-534-01056-5 : 689.00. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=589076&idb=0>.
2. Астапчук Виктор Андреевич. Корпоративные информационные системы: требования при проектировании : Учебное пособие для вузов / Астапчук В. А., Терещенко П. В. - 2-е изд. - Москва : Юрайт, 2019. - 110 с. - (Высшее образование). - ISBN 978-5-534-08410-8 : 229.00. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=573942&idb=0>.

Дополнительная литература:

1. Григорьев Михаил Викторович. Проектирование информационных систем : Учебное пособие для вузов / Григорьев М. В., Григорьева И. И. - Москва : Юрайт, 2019. - 318 с. - (Высшее образование). - ISBN 978-5-534-01305-4 : 619.00. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=586835&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

www.intuit.ru – Национальный открытый университет
ОС Microsoft Windows (лицензия по подписке Microsoft Imagine)

Microsoft Visual Studio Community Edition 2017 (бесплатная лицензия для обучения
<https://visualstudio.microsoft.com/ru/license-terms/mlt553321/>)

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

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 ФГОС ВО по направлению подготовки/специальности 02.04.01 - Математика и компьютерные науки.

Автор(ы): Старостин Николай Владимирович, доктор технических наук, доцент.

Заведующий кафедрой: Прилуцкий Михаил Хаимович, доктор технических наук.

Программа одобрена на заседании методической комиссии от 02.12.2024, протокол № 5.