

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
президиумом Ученого совета ННГУ
от 14.12.2021 г. протокол №4)

Рабочая программа дисциплины

Шаблоны проектирования

Уровень высшего образования
магистратура

Направление подготовки
02.04.01 Математика и компьютерные науки

Направленность образовательной программы
Математика и компьютерные науки

Форма обучения
очная

Нижний Новгород
2022

1. Место дисциплины в структуре ООП

Дисциплина «Б1.О.10 «Шаблоны проектирования» относится к обязательной части Блока 1 «Дисциплины (модули)» направления подготовки 02.04.01 «Математика и компьютерные науки» направленность «Математика и компьютерные науки». Дисциплина преподается во 2 семестре. Трудоемкость дисциплины составляет 4 зачетные единицы, 180 час, экзамен.

№ варианта	Место дисциплины в учебном плане образовательной программы	Стандартный текст для автоматического заполнения в конструкторе РПД
1	Блок 1. Дисциплины (модули) Обязательная часть	Дисциплина «Б1.О.10 «Шаблоны проектирования» относится к обязательной части ООП направления подготовки 02.04.01 «Математика и компьютерные науки».

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**	
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	<u>Знать</u> методологии разработки программного обеспечения	Задача (практическое задание)
	ОПК-3.2. Умеет использовать их в профессиональной деятельности	<u>Уметь</u> использовать методологии разработки программного обеспечения и использование их в профессиональной деятельности	Задача (практическое задание)
	ОПК-3.3. Имеет практический опыт применения программных средств, используемых при построении математических моделей в естественных науках	<u>Иметь практический опыт</u> профессиональной разработки программного обеспечения и использование их в профессиональной деятельности	Задача (практическое задание)

3. Структура и содержание дисциплины

3.1. Трудоемкость дисциплины

	Очная форма обучения
Общая трудоемкость	5 ЗЕТ
Часов по учебному плану	180

в том числе	
аудиторные занятия (контактная работа):	98
- занятия лекционного типа	32
- занятия семинарского типа	0
- занятия лабораторного типа	64
- текущий контроль (КСРИФ)	2
самостоятельная работа	46
Промежуточная аттестация – экзамен	36

3.2. Содержание дисциплины

Наименование и краткое содержание разделов и тем дисциплины (модуля), форма промежуточной аттестации по дисциплине (модулю)	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	Консультации и контроль	Всего		
Очная	Очная	Очная	Очная	Очная	Очная	Очная	Очная
РАЗДЕЛ 1. Паттерны проектирования и реализации программного обеспечения							
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	18	4		8		12	6

высокоуровневых управляемых систем							
Тема 7. Техники функционального расширения систем с минимальными изменениями	18	4		8		12	6
Тема 8. Проектирование с элементами оптимизации	18	4		8		12	6
В т.ч. текущий контроль (КСРИФ)	2					2	
Итого:	180	32		64	32	98	46
Промежуточная аттестация экзамен							

Практические занятия (лабораторные работы) организуются, в том числе в форме практической подготовки, которая предусматривает участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

Практическая подготовка предусматривает: выполнение задач и практических заданий, подготовку к вопросам к экзаменам.

На проведение практических занятий (лабораторных работ) в форме практической подготовки отводится 64 часа.

Практическая подготовка направлена на формирование и развитие:

- практических навыков в соответствии с профилем ОП: создание и сопровождение архитектуры программных средств, разработка и тестирование программного обеспечения;
- компетенции – ОПК-3.

Текущий контроль успеваемости реализуется в рамках занятий лабораторного типа, групповых или индивидуальных консультаций.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа студентов подразделяется на следующие категории:

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

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

Для обеспечения самостоятельной работы обучающихся используется электронный курс (Шаблоны проектирования ПО, <https://e-learning.unn.ru/course/view.php?id=4382>), созданный в системе электронного обучения ННГУ <https://e-learning.unn.ru/>.

5. Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю), 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 ошибок.	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 (ОПК-3)

Вопросы к экзамену

Вопрос
<ol style="list-style-type: none"> 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 Компоновщик (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).

5.3. Типовые задачи для оценки 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.

5.4. Типовые практические задания для оценки 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!

6. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Лаврищева Е.М. Программная инженерия. Парадигмы, технологии и CASE-средства: учебник для вузов. – 2 издание, испр. – М.: Издательство Юрайт, 2018 – 280 с. <https://biblio-online.ru/viewer/programmnyaya-inzheneriya-paradigmy-tehnologii-i-case-sredstva-444952#page/1>
2. Астапчук В.А., Корпоративные информационные системы: требования при проектировании: учеб. пособие для вузов – 2 изд., испр. и доп. - М.: Издательство Юрайт, 2019 – 110 с. <https://biblio-online.ru/viewer/korporativnye-informacionnye-sistemy-trebovaniya-pri-proektirovanii-444114#page/1>

б) дополнительная литература:

3. Григорьев М.В., Григорьева И.И Проектирование информационных систем: учеб. пособие для вузов - М.: Издательство Юрайт, 2019 – 318 с. <https://biblio-online.ru/viewer/proektirovanie-informacionnyh-sistem-434436#page/1>

в) 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

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой, оснащенные оборудованием и техническими средствами обучения: компьютерный класс, проектор, экран.

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 3++ направления подготовки 02.04.01 «Математика и компьютерные науки».

Автор: д.т.н., профессор кафедры ИАНИ Старостин Н.В.

Рецензент: к.т.н., доцент кафедры МОСТ, Мееров И.Б.

Заведующий кафедрой ИАНИ: д.т.н, проф., Прилуцкий М.Х.

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики от 01.12.2021 года, протокол № 2.