

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 президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 г.

**Рабочая программа дисциплины**

Шаблоны проектирования

---

Уровень высшего образования

Бакалавриат

---

Направление подготовки / специальность

09.03.03 - Прикладная информатика

---

Направленность образовательной программы

Прикладная информатика в области принятия решений

---

Форма обучения

очная

---

г. Нижний Новгород

2024 год начала подготовки

## 1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.06 Шаблоны проектирования относится к части, формируемой участниками образовательных отношений 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
ПК-4: Способен проводить исследование и описание процессов принятия решений в конкретной предметной (проблемной) области с применением современных информационных технологий, в том числе основанных на моделях и методах искусственного интеллекта	<p>ПК-4.1: Демонстрирует знание современных моделей и методов интеллектуальной поддержки процессов принятия решений.</p> <p>ПК-4.2: Демонстрирует умение применять системный подход к исследованию и описанию предметной (проблемной) области, формированию требований к ИС (ИИС) с учетом возможностей интеллектуальных технологий.</p> <p>ПК-4.3: Имеет практический опыт исследования и описания конкретной предметной области, разработки технического задания, эскизного и технического проектов ИС (ИИС).</p>	<p>ПК-4.1: Знать типовые процессы создания, развертывания и ввода в эксплуатацию системных архитектурных решений, основанных на паттернах проектирования.</p> <p>ПК-4.2: 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.</p> <p>ПК-4.3: 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,</p>	<p>Задания</p> <p>Тест</p>	<p>Экзамен:</p> <p>Контрольные вопросы</p>

		адаптации и тестирования новых системных архитектурных решений.		
ПК-5: Способен проектировать интеллектуальные ИС (ИИС) по видам обеспечения	<p>ПК-5.1: Демонстрирует знание современных технологий проектирования ИИС.</p> <p>ПК-5.2: Демонстрирует умение проектировать архитектуру ИИС по видам обеспечения.</p> <p>ПК-5.3: Имеет практический опыт проектирования конкретной ИИС по видам обеспечения.</p>	<p>ПК-5.1:</p> <p>Знать основные понятия, принципы и методы программирования ООП, языки программирования. Знать типовые решения (паттерны проектирования): фасад, адаптер, мост, компоновщик, итератор, декоратор, стратегия, одиночка, прототип, абстрактная фабрика, фабричный метод, шаблонный метод, строитель, цепочка обязанностей, команда, хранитель, посетитель, интерпретатор, заместитель, приспособленец, состояние.</p> <p>ПК-5.2:</p> <p>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.</p> <p>ПК-5.3:</p> <p>Имеет практический подготовки проектных решений и создания ИС на базе паттернов: фасад, адаптер, мост, компоновщик,</p>	<p>Задания</p> <p>Контрольная работа</p> <p>Тест</p>	<p>Экзамен:</p> <p>Контрольные вопросы</p> <p>Задания</p>

		итератор, декоратор, стратегия, одиночка, прототип, абстрактная фабрика, фабричный метод, шаблонный метод, посредник, цепочка обязанностей, команда, хранитель, посетитель, интерпретатор, заместитель, приспособленец, состояние.		
--	--	---	--	--

### 3. Структура и содержание дисциплины

#### 3.1 Трудоемкость дисциплины

	очная
<b>Общая трудоемкость, з.е.</b>	<b>5</b>
<b>Часов по учебному плану</b>	<b>180</b>
в том числе	
<b>аудиторные занятия (контактная работа):</b>	
- занятия лекционного типа	32
- занятия семинарского типа (практические занятия / лабораторные работы)	64
- КСР	2
<b>самостоятельная работа</b>	<b>46</b>
<b>Промежуточная аттестация</b>	<b>36</b>
	<b>Экзамен</b>

#### 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/ лабора- торные работы), часы	Всего	
	о ф о	о ф о	о ф о	о ф о	о ф о
Базовые паттерны проектирования	68	16	32	48	20
Системные архитектурные решения	74	16	32	48	26
Аттестация	36				
КСР	2			2	
Итого	180	32	64	98	46

## Содержание разделов и тем дисциплины

### РАЗДЕЛ 1. Базовые паттерны проектирования

Тема 1. Объектно-ориентированная парадигма

Тема 2. Введение в объектно-ориентированный анализ и базис языка визуального моделирования

Тема 3. Шаablонный метод проектирования

Тема 4. Шаablоны Фасад и Адаптер

Тема 5. Шаablон Мост

Тема 6. Шаablоны Компоновщик и Итератор

Тема 7. Шаablоны Декоратор и Стратегия

### РАЗДЕЛ 2. Системные архитектурные решения

Тема 8. Инстанцирование систем

Тема 9. Информа-цион-ный обмен

Тема 10. Управле-ние системой

Тема 11. Функциональное расширение систем

Тема 12. Целевое проектирование систем

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.

На проведение практических занятий / лабораторных работ в форме практической подготовки отводится: очная форма обучения - 6 ч.

### 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:

- электронный курс "Шаablоны проектирования" (<https://e-learning.unn.ru/course/view.php?id=4382>).

Иные учебно-методические материалы: Практическая подготовка направлена на формирование и развитие практических навыков в соответствии с профилем ОП: применение стандартов, норм и правил (в том числе установленные самостоятельно) при оформлении технической документации на различных стадиях проектирования и поддержки жизненного цикла информационных систем; способность вводить в эксплуатацию и осуществлять сопровождение ИС на всех этапах ее жизненного цикла, включая ее презентацию и начальное обучение пользователей.

Справочник по шаablонам проектирования. <http://www.oodesign.com/>

### 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 - Задания) для оценки сформированности компетенции ПК-4:

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.

Типовое 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.

### **5.1.2 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ПК-5:**

Типовое задание 1. Построить проект системного архитектурного решения с учетом применения в рамках конкретного контекста. Предоставить UML-диаграммы, документирующие ключевые архитектурные решения. Проверка задания.

Типовое задание 2. Реализовать заданный проект системного архитектурного решения в рамках конкретного контекста. На базе системного решения построить клиентский программный код, демонстрирующий принципы его работы. Проверка задания.

Типовое задание 3. Осуществить экспертизу качества системного архитектурного решения, выделить проблемы, разработать варианты их преодоления с помощью техники рефакторинга. Проверка задания.

### **Критерии оценивания (оценочное средство - Задания)**

Оценка	Критерии оценивания
превосходно	Все практические задания (задачи, лабораторные работы) выполнены в полном объеме и в срок, при этом применен творческий подход к решению нестандартных задач. Описаны все этапы выполнения заданий, код и результаты работы представлены преподавателю.

Оценка	Критерии оценивания
отлично	Все практические задания (задачи, лабораторные работы) выполнены в полном объеме и в срок. Описаны все этапы выполнения заданий, код и результаты работы представлены преподавателю.
очень хорошо	Выполнены основные этапы решения задания (задачи, лабораторной работы) или задача решена с незначительными недочетами. Код и результаты работы представлены преподавателю в срок.
хорошо	Выполнены часть этапов решения задания (задачи, лабораторной работы) или задача решена с недочетами. Код и результаты работы представлены преподавателю в срок.
удовлетворительно	Выполнены часть этапов решения задания (задачи, лабораторной работы) или задача решена с существенными недочетами. Код и результаты работы представлены преподавателю, но с отклонениями от сроков.
неудовлетворительно	Выполнены не все практические задания (задачи, лабораторной работы) или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, код работает некорректно, результаты работы не представлены преподавателю).
плохо	Студент не приступал к выполнению заданий (задач, лабораторной работы).

### 5.1.3 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-4:

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

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

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

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

### 5.1.4 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-5:

1. Метод функциональной декомпозиции позволяет подготовить код к изменениям:

- а) Практически нет.
- б) Да, это стандартный метод проектирования сложных систем.

2. Главные принципы ООП:

- а) Концепция инкапсулированной сущности.
- б) Концепция класса, механизмы наследования и полиморфизма.
- с) Концепция метода, механизмы инкапсуляции, наследования и полиморфизма.
- д) Концепция объекта, механизмы инкапсуляции, наследования и полиморфизма.

### Критерии оценивания (оценочное средство - Тест)

Оценка	Критерии оценивания
превосходно	95-100%
отлично	90-95%
очень хорошо	85-90%
хорошо	80-85%
удовлетворительно	70-80%
неудовлетворительно	30-70%
плохо	0-30%

### 5.1.5 Типовые задания (оценочное средство - Контрольная работа) для оценки сформированности компетенции ПК-5:

**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("архитектуры"), new Sign("!")); txt.Print(new PrinterDefault());
```

Результат работы кода – на консоль отправлен текст: Тестируем мою архитектуру!

### Критерии оценивания (оценочное средство - Контрольная работа)

Оценка	Критерии оценивания
превосходно	Студент получил верный ответ во всех заданиях. При этом студент продемонстрировал знание дополнительного материала.
отлично	Студент получил верный ответ во всех заданиях.
очень хорошо	Студент получил верный ответ в большинстве заданий.
хорошо	Студент решил большую часть задач с незначительными недочетами.
удовлетворительно	Студент решил большую часть задач с существенными недочетами.
неудовлетворительно	Студент допускает грубые ошибки в решении стандартных задач.
плохо	Отсутствие знаний материала, отсутствует способность решения стандартных задач.

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

### Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатор 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	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	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.

	ответа			ошибок	нных ошибок		
<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	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме	Продемонстрированы 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 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	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

### Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	<b>превосходно</b>	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
	<b>отлично</b>	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	<b>очень хорошо</b>	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	<b>хорошо</b>	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	<b>удовлетворительно</b>	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	<b>неудовлетворительно</b>	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	<b>плохо</b>	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 ПК-4

Инстанцирование систем. Принципы. Идеология объекта-одиночки (Singleton) в системе объектов. Способы доступа к объекту-одиночке.
Инстанцирование систем. Конфигурирование. Абстрактная Фабрика (Abstract Factory) и Инструментарий (Kit).
Инстанцирование систем. Каркасы приложений. Применение решений Фабричного Метода (Factory Method) и Шаблонного Метода (Template Method).
Инстанцирование систем. Клонирование объектов и систем объектов. Поверхностное и глубокое клонирование на базе Прототипа (Prototype).
Инстанцирование систем. Организация процесса конструирования различных представлений сложного объекта на базе решения Строитель (Builder).
Информационный обмен. Основополагающие принципы. Классификация моделей. Простейшие модели и модель на базе Посредника (Mediator).
Информационный обмен. Модель доставки сообщения на базе решения Цепочка Обязанностей (Chain of Responsibility). Решение без менеджера и с менеджером. Проксирование сообщений.
Информационный обмен. Широковещательные трансляции на базе шаблона Наблюдатель (Observer). Особенности реализации систем типа Субъект-Наблюдатель без менеджера и с менеджером.
Информационный обмен. Объекты с функциональностью Субъекта и Наблюдателя. Проблема учета циклических связей (зависимостей). Решение без менеджера и с менеджером.
Управление системой. Идеология представление команды (операции) в виде объекта. Манипулирование командами как объектами. Протоколирование команд.
Управление системой. Идеология представление команды (операции) в виде объекта. Манипулирование командами как объектами. Организация макросов (составные команды) на базе шаблона Компонировщик (Composite).
Управление системой. Менеджер команд и универсальные механизмы отката (отмены операций) на базе решений Команда (Command) и Хранитель (Memento).
Функциональное расширение систем. Нарращивание функциональности отдельных объектов (классов) без изменения существующего кода на базе решений Декоратор (Decorator) и Стратегия (Strategy).
Функциональное расширение систем. Двойная диспетчеризация. Динамическое определение новых

функций для систем объектов без изменения существующего кода на базе решения Посетитель (Visitor).
Целевое проектирование систем. Оптимизация по времени работы. Идеологии кэширования и отложенной реакции на событие. Объектно-ориентированная организация событийных систем на основе решения Заместитель (Proxy).
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).
Целевое проектирование систем. Функциональный дизайн. Представление грамматики языка и интерпретация предложений на базе шаблона Интерпретатор (Interpreter).
Целевое проектирование систем. Автоматный (конечный автомат) дизайн. Идеология совмещения в одном объекта разных состояний на основе решения Состояние (State).

### 5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-5

Функциональная декомпозиция. Проблема обработки изменяющихся требований.
Объектно-ориентированная парадигма. Модели наследования и композиции.
Объектно-ориентированная парадигма. Инкапсуляция в механизме полиморфизма.
Объектно-ориентированная парадигма. Модели делегирования.
Абстрактный класс и интерфейс. Реализации класса и интерфейса. Проектирование на уровне интерфейсов.
Сложность систем. Объектная модель. Классификация. Идентификация классов и объектов. Ключевые абстракции и механизмы.
Основные этапы разработки объектно-ориентированной системы.
UML. Определение. Назначение. Концептуальная модель.
Концептуальная модель языка UML. Сущности.
Концептуальная модель языка UML. Отношения.
Концептуальная модель языка UML. Диаграмма прецедентов и диаграммы взаимодействий.
Концептуальная модель языка UML. Статические и динамические модели программных систем на примере диаграмм классов и диаграмм объектов.
Шаблоны проектирования. Основные термины и понятия. Механизмы повторного использования.

Система каталогизации шаблонов проектирования.
Методы проектирования, основанные на стратегиях и шаблонах проектирования.
Модели инкапсуляции. Стратегия «Инкапсуляции изменчивости» в анализе общности и изменчивости.
Стратегии «Композиция предпочтительней наследования» и «Одно правило, одно место».
Специфика работа со сложной системой с множеством интерфейсов. Шаблон Фасад (Facade).
Проблема совместимости интерфейсов. Шаблон Адаптер (Adapter).
Понятия абстракции и реализации. Механизм отделения абстракции от реализации. Шаблон Мост (Bridge).
Механизм компоновки объектов в плоские коллекции и иерархические структуры. Манипулирование составными объектами. Шаблон Компоновщик (Composite). Виды Компоновщика.
Составные объекты. Организация доступа к элементам составного объекта на базе шаблона Итератор (Iterator).
Робастность итераторов плоских коллекций и иерархических структур.
Динамическое расширение функциональности объектов. Шаблон Декоратор (Decorator) – как гибкая альтернатива порождению подклассов.
Инкапсуляция алгоритма в объект. Механизм «прозрачной» замены алгоритма. Шаблон Стратегия (Strategy).

### **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

Оценка	Критерии оценивания
	задач.

### 5.3.3 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ПК-5

**ЗАДАЧА 1.** Отобразите на диаграмме классов код

**class A {**

int a;

public:

A(int x) { a = x; }

virtual int geta() { return a; }

};

**class B : public A {**

int b;

public:

B(int x, int y) : B(y) { b = x; }

virtual int geta() { return a+b; }

};

**Критерии оценивания (оценочное средство - Задания)**

Оценка	Критерии оценивания
превосходно	Все практические задания выполнены в полном объеме и в срок, при этом применен творческий подход к решению нестандартных задач. Описаны все этапы выполнения заданий, код и результаты работы представлены преподавателю.
отлично	Все практические задания выполнены в полном объеме и в срок. Описаны все этапы выполнения заданий, код и результаты работы представлены преподавателю.
очень хорошо	Выполнены основные этапы решения задания или задача решена с незначительными недочетами. Код и результаты работы представлены преподавателю в срок.

Оценка	Критерии оценивания
хорошо	Выполнены часть этапов решения задания или задача решена с недочетами. Код и результаты работы представлены преподавателю в срок.
удовлетворительно	Выполнены часть этапов решения задания или задача решена с существенными недочетами. Код и результаты работы представлены преподавателю, но с отклонениями от сроков.
неудовлетворительно	Выполнены не все практические задания или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, код работает некорректно, результаты работы не представлены преподавателю).
плохо	Студент не приступал к выполнению заданий .

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

Основная литература:

1. Унгер А. Ю. Шаблоны объектно-ориентированного проектирования в языке C++ / Унгер А. Ю. - Москва : РТУ МИРЭА, 2022. - 67 с. - Книга из коллекции РТУ МИРЭА - Инженерно-технические науки., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=861259&idb=0>.

Дополнительная литература:

1. Достовалов Д. Н. Объектно-ориентированный анализ и проектирование. Задачи и примеры на C++ : учебное пособие / Достовалов Д. Н., Лауферман О. В. - Новосибирск : НГТУ, 2022. - 74 с. - Утверждено Редакционно-издательским советом университета в качестве учебного пособия. - Книга из коллекции НГТУ - Инженерно-технические науки. - ISBN 978-5-7782-4708-6., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=860502&idb=0>.

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

1. Операционные системы семейства MicrosoftWindows, лицензия по подписке MicrosoftImagine.
2. Браузер Google Chrome, предоставляется бесплатно на условиях лицензионных соглашений на программное обеспечение с открытым исходным кодом.
3. Среда разработки семейства MicrosoftVisualStudio, лицензия по подписке MicrosoftImagine.

## 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 ОС ННГУ по направлению подготовки 09.03.03 - Прикладная информатика.

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

Заведующий кафедрой: Прилуцкий Михаил Хаимович, доктор технических наук.

Программа одобрена на заседании методической комиссии от 13.12.2023, протокол № 3.