

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 № 10 от 02.12.2024 г.

**Рабочая программа дисциплины**

Паттерны проектирования и реализации программного обеспечения

---

Уровень высшего образования  
Магистратура

---

Направление подготовки / специальность  
09.04.03 - Прикладная информатика

---

Направленность образовательной программы  
Проектирование и автоматизация производства изделий микроэлектроники

---

Форма обучения  
очная

---

г. Нижний Новгород

2025 год начала подготовки

## 1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.02 Паттерны проектирования и реализации программного обеспечения 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
УК-2: Способен управлять проектом на всех этапах его жизненного цикла	<p>УК-2.1: Демонстрирует знание основных этапов жизненного цикла ИТ-проекта</p> <p>УК-2.2: Демонстрирует умение разрабатывать и анализировать альтернативные варианты планирования этапов проекта для достижения намеченных целей</p> <p>УК-2.3: Демонстрирует наличие практического опыта принятия решений на различных этапах конкретных проектов</p>	<p>УК-2.1: Знать типовые процессы создания, развертывания и ввода в эксплуатацию системных архитектурных решений, основанных на паттернах проектирования.</p> <p>УК-2.2: 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>УК-2.3: 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, адаптации и тестирования новых системных архитек-</p>	<p>Задачи</p> <p>Тест</p>	<p>Экзамен:</p> <p>Контрольные вопросы</p>

		турных решений.		
ПК-6: Способен управлять процессами и проектами по созданию (модификации) информационных ресурсов (сайт, портал) Интернет	<p>ПК-6.1: Демонстрирует знание способов управления процессами и проектами по созданию (модификации) информационных ресурсов (сайт, портал) Интернет</p> <p>ПК-6.2: Демонстрирует умение планировать и организовывать разработку процессов и проектов по созданию (модификации) информационных ресурсов (сайт, портал), применять инструментальные средства</p> <p>ПК-6.3: Имеет практический опыт планирования и организации деятельности по созданию (модификации) информационных ресурсов (сайт, портал) Интернет</p>	<p>ПК-6.1: 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-6.2: 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-6.3: Имеет опыт практический подготовки проектных решений и создания ИС на базе паттернов: фасад, адаптер, мост, компоновщик, итератор, декоратор, стратегия, одиночка,</p>	Задачи Тест	Экзамен: Контрольные вопросы

		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

	<b>очная</b>
<b>Общая трудоемкость, з.е.</b>	<b>6</b>
<b>Часов по учебному плану</b>	<b>216</b>
в том числе	
<b>аудиторные занятия (контактная работа):</b>	
- занятия лекционного типа	32
- занятия семинарского типа (практические занятия / лабораторные работы)	32
- КСР	2
<b>самостоятельная работа</b>	<b>114</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/лабораторные работы), часы	Всего	
	о ф о	о ф о	о ф о	о ф о	о ф о
Тема 1. Объектно-ориентированная парадигма	8	2	2	4	4
Тема 2. Введение в объектно-ориентированный анализ и базис языка визуального моделирования	18	4	4	8	10
Тема 3. Шаблонный метод проектирования	14	2	2	4	10
Тема 4. Шаблоны Фасад и Адаптер	14	2	2	4	10
Тема 5. Шаблон Мост	14	2	2	4	10
Тема 6. Шаблоны Компоновщик и Итератор	14	2	2	4	10
Тема 7. Шаблоны Декоратор и Стратегия	14	2	2	4	10

Тема 8. Инстанцирование систем	16	3	3	6	10
Тема 9. Информационный обмен	16	3	3	6	10
Тема 10. Управление системой	16	3	3	6	10
Тема 11. Функциональное расширение систем	16	3	3	6	10
Тема 12. Целевое проектирование систем	18	4	4	8	10
Аттестация	36				
КСР	2			2	
Итого	216	32	32	66	114

### 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. Шаблонный метод проектирования

Тема 4. Шаблоны Фасад и Адаптер

Тема 5. Шаблон Мост

Тема 6. Шаблоны Компоновщик и Итератор

Тема 7. Шаблоны Декоратор и Стратегия

#### РАЗДЕЛ 2. Системные архитектурные решения

Тема 8. Инстанцирование систем

Тема 9. Информационный обмен

Тема 10. Управление 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 - 2 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:

Электронные курсы, созданные в системе электронного обучения ННГУ:

Паттерны проектирования и реализации программного обеспечения,

<https://e-learning.unn.ru/enrol/index.php?id=4381>.

## 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 УК-2:

**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.

**Задание 2. 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 3. 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).

#### 5.1.2 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ПК-6:

**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.

#### 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 зачтено	Выполнены не все практические задания или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, получен неверный ответ, результаты работы не представлены преподавателю).

### **5.1.3 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции УК-2:**

1. Метод функциональной декомпозиции позволяет подготовить код к изменениям.

- a) Практически нет.
- b) Да, это стандартный метод проектирования сложных систем.

2. Главные принципы ООП.

- a) Концепция инкапсулированной сущности.
- b) Концепция класса, механизмы наследования и полиморфизма.
- c) 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.

### **5.1.4 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-6:**

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

- a) 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.

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

а) 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.

### Критерии оценивания (оценочное средство - Тест)

Оценка	Критерии оценивания
зачтено	как минимум 80% правильных ответов в тесте
не зачтено	менее 80% правильных ответов в тесте

### 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 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 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	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 УК-2

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 Компоновщик (Composite).
Управление системой. Менеджер команд и универсальные механизмы отката (отмены операций) на

базе решений Команда (Command) и Хранитель (Memento).
Функциональное расширение систем. Наращивание 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-6

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.

Оценка	Критерии оценивания
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. 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. Шаблоны проектирования для облачной среды / Дэвис К. - Москва : ДМК-пресс, 2020., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=665082&idb=0>.

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

1. Справочник по шаблонам проектирования. <http://www.oodesign.com/>
2. Архитектурное проектирование программного обеспечения. Бесплатный курс для самообразования на официальном сайте Интуит.py. <http://www.intuit.ru/studies/courses/3509/751/info>
3. Объектно-ориентированный анализ и программирование. Бесплатный курс для самообразования на официальном сайте Интуит.py. <http://www.intuit.ru/studies/courses/491/347/info>
4. Проектирование информационных систем. Бесплатный курс для самообразования на официальном сайте Интуит.py. <http://www.intuit.ru/studies/courses/491/347/info>
5. Операционные системы семейства MicrosoftWindows, лицензия по подписке MicrosoftImagine.
6. Браузер Google Chrome, предоставляется бесплатно на условиях лицензионных соглашений на программное обеспечение с открытым исходным кодом.
7. 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, специализированным оборудованием: 1. Учебно-лабораторный интерактивный комплекс «Математическое и программное обеспечение проектирования изделий микроэлектроники» (корпус 6, ауд. 116), 2. Учебно-лабораторный интерактивный комплекс «Математическое и программное обеспечение управления высокотехнологичным производством» (корпус 6, ауд. 120)

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/специальности 09.04.03 - 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 02.12.2024, протокол № 5.