

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**

Проектирование и архитектура программных систем

---

Уровень высшего образования

Бакалавриат

---

Направление подготовки / специальность

09.03.04 - Программная инженерия

---

Направленность образовательной программы

Разработка программно-информационных систем

---

Форма обучения

очная

---

г. Нижний Новгород

2025 год начала подготовки

## 1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.11 Проектирование и архитектура программных систем 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-12: Способен осуществлять концептуальное, функциональное и логическое проектирование систем и подсистем малого и среднего масштаба и сложности.	<p>ПК-12.1: Знает методы концептуального, функционального и логического проектирования</p> <p>ПК-12.2: Знает типовые архитектурные стили и паттерны проектирования</p> <p>ПК-12.3: Умеет применять архитектурные стили и паттерны проектирования при решении типовых задач</p> <p>ПК-12.4: Владеет навыками разработки проекта программной системы с учетом возможностей и ограничений</p>	<p>ПК-12.1: ЗНАТЬ знать способы передачи, приёма, обработки, анализа и хранения информации.</p> <p>ПК-12.2: Знать типовые архитектурные стили и паттерны проектирования</p> <p>ПК-12.3: Уметь применять архитектурные стили и паттерны проектирования при решении типовых задач</p> <p>ПК-12.4: Владеть навыками разработки проекта программной системы с учетом возможностей ее дальнейшего развития, а также повторного использования некоторых ее компонент в других проектах</p>	Собеседование Практическое задание	Экзамен: Контрольные вопросы
ПК-13: Способен вырабатывать требования по модернизации систем и подсистем малого и среднего масштаба и сложности	<p>ПК-13.1: Знает основы управления изменениями требований</p> <p>ПК-13.2: Умеет анализировать влияния изменений при проектировании и модернизации систем</p>	ПК-13.1: Знать методiku анализа предметной области и выявления требований к системе, основы управления изменениями требований	Собеседование Практическое задание	Экзамен: Контрольные вопросы

	ПК-13.3: Владеет навыками создания и изучения запросов на изменение требований к системе	ПК-13.2: Уметь применять на практике методику выявления требований и документировать требования с использованием языка UML при проектировании и модернизации систем  ПК-13.3: Владеть навыками разработки формальных требований к программной системе, основанных на потребностях заинтересованных лиц		
--	--	--	--	--

### 3. Структура и содержание дисциплины

#### 3.1 Трудоемкость дисциплины

	очная
<b>Общая трудоемкость, з.е.</b>	<b>4</b>
<b>Часов по учебному плану</b>	<b>144</b>
в том числе	
<b>аудиторные занятия (контактная работа):</b>	
- занятия лекционного типа	16
- занятия семинарского типа (практические занятия / лабораторные работы)	16
- КСР	2
<b>самостоятельная работа</b>	<b>74</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

Рынок разработки ПО в России и в мире, состояние, тенденции, проблемы, перспективы; успешные и неудачные проекты; роль технологий и методологий разработки ПО.	8	2	2	4	4
Методологии проектирования и разработки ПО. Обзор методологий: RUP, MSF, CMM, CMMI, Agile, SCRUM. Базовые принципы разработки ПО на примере Rational Unified Process.	12	2	2	4	8
. 6 лучших практик разработки ПО, их вклад в решение типовых проблем, жизненный цикл разработки ПО на примере RUP – итерации, фазы, вехи, дисциплины.	12	2	2	4	8
От теории к практике. Требования, их выявление и организация, диаграммы вариантов использования. Примеры. Анализ требований.	12	2	2	4	8
Понятие и роль архитектуры, классификация архитектурных стилей – файл-сервер, клиент-сервер, компонентная, послойная, многозвенная архитектура и другие. Достоинства и недостатки, области применения рассмотренных стилей.	12	2	2	4	8
Примеры типовых решений. Архитектурные паттерны. Паттерны на примере. Архитектурные паттерны MVC, MVP, MVVM, DDD. Примеры.	10	2	2	4	6
. Подсистемы. Объектный подход. Абстракция, инкапсуляция, наследование, полиморфизм; SOLID-принципы. Примеры.	20	2	2	4	16
От требований к архитектуре: разработка архитектуры приложения на примере. Подсистемы и интерфейсы, взаимодействие подсистем.	20	2	2	4	16
Аттестация	36				
КСР	2			2	
Итого	144	16	16	34	74

### Содержание разделов и тем дисциплины

1. Рынок разработки ПО в России и в мире, состояние, тенденции, проблемы, перспективы; успешные и неудачные проекты; роль технологий и методологий разработки ПО.
2. Методологии проектирования и разработки ПО. Обзор методологий: RUP, MSF, CMM, CMMI, Agile, SCRUM. Базовые принципы разработки ПО на примере Rational Unified Process.
3. 6 лучших практик разработки ПО, их вклад в решение типовых проблем, жизненный цикл разработки ПО на примере RUP – итерации, фазы, вехи, дисциплины.
4. От теории к практике. Требования, их выявление и организация, диаграммы вариантов использования. Примеры. Анализ требований.
5. Понятие и роль архитектуры, классификация архитектурных стилей – файл-сервер, клиент-сервер, компонентная, послойная, многозвенная архитектура и другие. Достоинства и недостатки, области применения рассмотренных стилей.
6. Примеры типовых решений. Архитектурные паттерны. Паттерны на примере. Архитектурные паттерны MVC, MVP, MVVM, DDD. Примеры.
7. Подсистемы. Объектный подход. Абстракция, инкапсуляция, наследование, полиморфизм; SOLID-принципы. Примеры.
8. От требований к архитектуре: разработка архитектуры приложения на примере. Подсистемы и интерфейсы, взаимодействие подсистем.

### 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:

Анализ и проектирование ПО, <https://e-learning.unn.ru/course/view.php?id=6132>),.

## **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 - Собеседование) для оценки сформированности компетенции ПК-12:**

1. Рынок разработки ПО в России и в мире, состояние, тенденции, проблемы, перспективы; успешные и неудачные проекты; роль технологий и методологий разработки ПО.
2. 6. Понятие и роль архитектуры, классификация архитектурных стилей – файл-сервер, клиент-сервер. Достоинства и недостатки, области применения рассмотренных стилей.
3. 7. Понятие и роль архитектуры, классификация архитектурных стилей –компонентная, послойная, многослойная архитектура. Достоинства и недостатки, области применения рассмотренных стилей.
4. 8. Архитектурный паттерн MVC. Примеры.
5. 9. Архитектурный паттерн MVP. Примеры.
6. 10. Архитектурный паттерн MVVM. Примеры.
7. 11. Архитектурный паттерн DDD. Примеры.
8. 12. Подсистемы. Объектный подход. Абстракция, инкапсуляция, наследование, полиморфизм.
9. 13. SOLID-принципы. Примеры.

#### **5.1.2 Типовые задания (оценочное средство - Собеседование) для оценки сформированности компетенции ПК-13:**

1. 2. Методологии проектирования и разработки ПО: RUP, MSF, CMM, CMMI. Базовые принципы разработки ПО на примере Rational Unified Process.
2. Методологии проектирования и разработки ПО: Agile, SCRUM.
3. 6 лучших практик разработки ПО, их вклад в решение типовых проблем, жизненный цикл разработки ПО на примере RUP – итерации, фазы, вехи, дисциплины.
4. От теории к практике. Требования, их выявление и организация, диаграммы вариантов использования. Примеры. Анализ требований.

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

Оценка	Критерии оценивания
превосходно	Студент дал развернутый ответ на все вопросы и при этом продемонстрировал знание дополнительного материала

Оценка	Критерии оценивания
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.

### 5.1.3 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ПК-12:

Задания для групповых работ

- Проектирование–1: проектирование программного продукта. Выбор и применение архитектурных стилей и паттернов.
- Проектирование–2: проектирование программного продукта. Представление разработанной архитектуры.
- Проектирование–3: проектирование программного продукта. Выделение подсистем.
- Проектирование–4: проектирование программного продукта. Взаимодействие подсистем, проектирование интерфейсов подсистем.

### 5.1.4 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ПК-13:

Задания для групповых работ:

- Анализ требований: выявление и организация требований для задачи, стоящей перед командой. Подготовка пакета UML-диаграмм.
- Реализация проекта: проектирование программного продукта. Реализация подсистем.
- Представление и демонстрация: проектирование программного продукта. Финальное представление проекта и демонстрация работы программного прототипа.

Тематика групповых работ для оценивания результатов обучения:

- 1) Система обработки метеоинформации (редактор карт)
- 2) Система обработки метеоинформации (задание климата)

### Критерии оценивания (оценочное средство - Практическое задание)

Оценка	Критерии оценивания
зачтено	Выполнена основная часть задания, возможно с незначительными недочетами

Оценка	Критерии оценивания
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 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	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 ПК-12

1. Рынок разработки ПО в России и в мире, состояние, тенденции, проблемы, перспективы; успешные и неудачные проекты; роль технологий и методологий разработки ПО.
2. 6. Понятие и роль архитектуры, классификация архитектурных стилей – файл-сервер, клиент-сервер. Достоинства и недостатки, области применения рассмотренных стилей.
3. 7. Понятие и роль архитектуры, классификация архитектурных стилей –компонентная, послойная, многозвенная архитектура. Достоинства и недостатки, области применения рассмотренных стилей.
4. 8. Архитектурный паттерн MVC. Примеры.
5. 9. Архитектурный паттерн MVP. Примеры.
6. 10. Архитектурный паттерн MVVM. Примеры.
7. 11. Архитектурный паттерн DDD. Примеры.
8. 12. Подсистемы. Объектный подход. Абстракция, инкапсуляция, наследование, полиморфизм.
9. 13. SOLID-принципы. Примеры.

#### 5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-13



1. 2. Методологии проектирования и разработки ПО: RUP, MSF, CMM, CMMI. Базовые принципы разработки ПО на примере Rational Unified Process.
2. Методологии проектирования и разработки ПО: Agile, SCRUM.
3. 6 лучших практик разработки ПО, их вклад в решение типовых проблем, жизненный цикл разработки ПО на примере RUP – итерации, фазы, вехи, дисциплины.
4. От теории к практике. Требования, их выявление и организация, диаграммы вариантов использования. Примеры. Анализ требований.

### **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	При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач.

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

Основная литература:

1. Вендров Александр Михайлович. Проектирование программного обеспечения экономических информационных систем : учеб. для студентов эконом. вузов, обучающихся по специальностям "Прикладная информатика (по обл.)" и "Прикладная математика и информатика". - 2-е изд., перераб. и доп. - М. : Финансы и статистика, 2006. - 544 с. : ил. - ISBN 5-279-02937-8 : 180.00., 94 экз.

Дополнительная литература:

1. Введение в программную инженерию / Кознов Д.В. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=662933&idb=0>.

2. Методы и средства инженерии программного обеспечения / Петрухин В.А., Лаврищева Е.М. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663012&idb=0>.

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

- Что такое архитектура программного обеспечения?  
<https://msdn.microsoft.com/ru-ru/hh144976.aspx>
- Среда разработки семейства Microsoft Visual Studio, лицензия по подписке Microsoft Imagine.
- UML diagrams online. <https://creately.com/Draw-UML-and-Class-Diagrams-Online>, свободное ПО.

## **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/специальности 09.03.04 - Программная инженерия.

Автор(ы): Чумаков Виктор Викторович.

Заведующий кафедрой: Баркалов Константин Александрович, доктор технических наук.

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