

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Структурное подразделение «Брикс кафедры»

УТВЕРЖДЕНА:
на заседании кафедры
Протокол №15 от 18 марта 2025 г.

Рабочая программа дисциплины

**«ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ / SOFTWARE
DEVELOPMENT TECHNOLOGIES»**

Направление: 09.03.01 Информатика и вычислительная техника

Искусственный интеллект и компьютерные науки /Artificial Intelligence and Computer
Science

Квалификация: Бакалавр

Форма обучения: очная

Документ подписан простой
электронной подписью
Составитель программы:
Эррахел Назих
Дата подписания: 17.06.2025

Документ подписан простой
электронной подписью
Утвердил: Киреев Анна
Павловна
Дата подписания: 17.06.2025

Документ подписан простой
электронной подписью
Согласовал: Афанасьев
Александр Диомидович
Дата подписания: 17.06.2025

Год набора – 2025

Иркутск, 2025 г.

1 Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

1.1 Дисциплина «Технологии разработки программного обеспечения / Software Development Technologies» обеспечивает формирование следующих компетенций с учётом индикаторов их достижения

Код, наименование компетенции	Код индикатора компетенции
ПКС-1 Способен проводить самостоятельные исследования в области искусственного интеллекта	ПКС-1.2, ПКС-1.3
ПКС-3 Способность разрабатывать программные продукты, используя современные технологии в области искусственного интеллекта	ПКС-3.1
ПКС-4 Способность выполнять процесс поиска, диагностики ошибок и оптимизации программного обеспечения в области искусственного интеллекта	ПКС-4.1, ПКС-4.3
ПКС-5 Способен реализовывать проекты в сфере искусственного интеллекта	ПКС-5.2, ПКС-5.4

1.2 В результате освоения дисциплины у обучающихся должны быть сформированы

Код индикатора	Содержание индикатора	Результат обучения
ПКС-3.1	Умеет разрабатывать и применять процедуры проверки работоспособности программного обеспечения	Знать Основы тестирования ПО: модульное, интеграционное, системное тестирование. Методы верификации и валидации программного обеспечения. Инструменты автоматизации тестирования (например, JUnit, Selenium). Уметь Разрабатывать тестовые сценарии и тест-кейсы для проверки функциональности ПО. Применять процедуры тестирования для выявления ошибок в программных системах. Анализировать результаты тестирования и предлагать улучшения. Владеть Навыками разработки и выполнения тестовых сценариев. Инструментами тестирования и анализа качества ПО. Техниками документирования результатов тестирования.
ПКС-4.1	Знает требования к современному программному обеспечению для решения прикладных задач	Знать Функциональные и нефункциональные требования к ПО (производительность, надежность, масштабируемость). Стандарты разработки ПО

		(например, ISO/IEC 25010). Особенности требований к ПО в области искусственного интеллекта. Уметь Анализировать требования к ПО для решения прикладных задач. Составлять спецификации требований на основе технического задания. Оценивать соответствие ПО заявленным требованиям. Владеть Навыками анализа и документирования требований к ПО. Техниками работы с техническими заданиями и стандартами качества. Профессиональной терминологией на русском и английском языках в области разработки ПО.
ПКС-5.2	Умеет составлять формализованные описания решений поставленных задач в соответствии с требованиями технического задания	Знать Основы языка UML: сущности, отношения, типы диаграмм. Структура и требования к техническому заданию. Принципы формализации решений задач в области разработки ПО. Уметь Разрабатывать UML-диаграммы (вариантов использования, классов, последовательностей) для описания решений. Составлять формализованные описания решений в соответствии с техническим заданием. Документировать решения в виде пояснительных записок. Владеть Навыками построения UML-диаграмм с использованием CASE-средств (например, StarUML). Техниками составления технической документации. Методами проверки соответствия решений требованиям технического задания.
ПКС-1.2	Сформирован навык постановки цели исследования и ее декомпозиции	Знать Основы научного метода и этапы проведения исследования. Методы декомпозиции задач в области искусственного интеллекта и разработки ПО. Принципы постановки целей исследования (SMART:

		конкретность, измеримость, достижимость, релевантность, ограниченность во времени). Уметь Формулировать цели исследования в области искусственного интеллекта и разработки ПО. Выполнять декомпозицию целей на подзадачи с учетом их приоритетности и последовательности. Анализировать результаты декомпозиции для планирования исследования. Владеть Навыками структурирования задач с использованием методов декомпозиции. Техниками постановки целей и их документирования в виде формализованных описаний. Инструментами для планирования исследований (например, Mind Maps, Gantt charts).
ПКС-1.3	Способен определить сферы применения результатов научно-исследовательской работы в профессиональной деятельности	Знать Основные области применения искусственного интеллекта в профессиональной деятельности. Методы анализа потребностей рынка и индустрии для применения результатов исследований. Примеры практического использования результатов исследований в разработке ПО. Уметь Определять потенциальные области применения результатов исследований в разработке ПО и ИИ. Анализировать релевантность результатов исследования для решения практических задач. Составлять рекомендации по внедрению результатов исследования в профессиональную деятельность. Владеть Навыками анализа и сопоставления результатов исследований с профессиональными задачами. Техниками составления отчетов о применении результатов

		исследований. Инструментами для оценки экономической и технической эффективности внедрения результатов.
ПКС-4.3	Умеет анализировать сообщения об ошибках в программных средствах	Знать Типы ошибок в ПО (синтаксические, логические, runtime-ошибки). Методы диагностики и классификации ошибок в программных системах. Инструменты для анализа логов и сообщений об ошибках (например, Bugzilla, Jira). Уметь Анализировать сообщения об ошибках для определения их причин. Классифицировать ошибки по типу, критичности и влиянию на систему. Предлагать решения для устранения выявленных ошибок. Владеть Навыками работы с системами управления ошибками. Техниками анализа и интерпретации логов и отчетов об ошибках. Методами документирования процесса устранения ошибок.
ПКС-5.4	Способен проводить проектную работу в сфере профессиональной деятельности	Знать Этапы проектной работы: анализ, проектирование, реализация, тестирование, внедрение. Принципы управления проектами в разработке ПО (Agile, Scrum, Waterfall). Роль проектной документации в профессиональной деятельности. Уметь Планировать этапы проектной работы в области разработки ПО. Реализовывать проектные задачи с использованием объектно-ориентированного подхода. Координировать работу в команде для выполнения проектных задач. Владеть Навыками управления проектами с использованием современных методологий. Инструментами разработки ПО (например, IDE, системы контроля версий). Техниками составления и защиты

		проектной документации.
--	--	-------------------------

2 Место дисциплины в структуре ООП

Изучение дисциплины «Технологии разработки программного обеспечения / Software Development Technologies» базируется на результатах освоения следующих дисциплин/практик: «Программирование / Programming»

Дисциплина является предшествующей для дисциплин/практик: «Основы проектной деятельности / Introduction to Project Development», «Технологии внедрения решений искусственного интеллекта / Artificial Intelligence Implementation Technologies»

3 Объем дисциплины

Объем дисциплины составляет – 4 ЗЕТ

Вид учебной работы	Трудоемкость в академических часах (Один академический час соответствует 45 минутам астрономического часа)	
	Всего	Семестр № 4
Общая трудоемкость дисциплины	144	144
Аудиторные занятия, в том числе:	72	72
лекции	36	36
лабораторные работы	0	0
практические/семинарские занятия	36	36
Самостоятельная работа (в т.ч. курсовое проектирование)	36	36
Трудоемкость промежуточной аттестации	36	36
Вид промежуточной аттестации (итогового контроля по дисциплине)	Зачет	Зачет

4 Структура и содержание дисциплины

4.1 Сводные данные по содержанию дисциплины

Семестр № 4

№ п/п	Наименование раздела и темы дисциплины	Виды контактной работы						СРС		Форма текущего контроля
		Лекции		ЛР		ПЗ(СЕМ)				
		№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	
1	2	3	4	5	6	7	8	9	10	11
1	Жизненный цикл программного продукта, модели жизненного цикла, сферы их применения	1, 2	4			1	4	5	4	Тест
2	Основные подходы к разработке программного обеспечения	3	4			2	4	4	4	Устный опрос
3	Язык UML.	4, 5	8			3, 4	8	5, 5	8	Решение

	Диаграммы UML, их назначение и правила составления									задач
4	Применение выполнения этапов анализа и проектирования ПС	6	4			5	4	2	4	Отчет по лабораторной работе
5	Создание объектно-ориентированного программного обеспечения	7	4			6, 7	8	1, 5	8	Отчет по лабораторной работе
6	Современный подход к проверке при создании программного обеспечения (ПО)	8	4			8, 9	8	5	4	Решение задач
7	Тенденции развития технологий разработки ПО	9	4					3	4	Доклад
8	Средства оценки качества программных продуктов	10	4							Тест
	Промежуточная аттестация								36	Зачет
	Всего		36				36		72	

4.2 Краткое содержание разделов и тем занятий

Семестр № 4

№	Тема	Краткое содержание
1	Жизненный цикл программного продукта, модели жизненного цикла, сферы их применения	Понятие жизненного цикла ПО, этапы (анализ, проектирование, реализация, тестирование). Модели: каскадная, спиральная, с промежуточным контролем. Их преимущества, недостатки и области применения.
2	Основные подходы к разработке программного обеспечения	Сравнение структурного и объектно-ориентированного подходов. Принципы функциональной и объектной декомпозиции, их применение в разработке ПО.
3	Язык UML. Диаграммы UML, их назначение и правила составления	Введение в UML: сущности, отношения, диаграммы (вариантов использования, классов, последовательностей, деятельности). Правила построения и применения для моделирования ПО.
4	Применение выполнения этапов анализа и проектирования ПС	Процессы анализа (функциональные и нефункциональные требования) и проектирования (архитектура ПО). Использование UML для документирования этапов.
5	Создание объектно-ориентированного	Основы ООП: абстрагирование, инкапсуляция, модульность. Разработка ПО на Python с

	программного обеспечения	использованием паттернов проектирования.
6	Современный подход к проверке при создании программного обеспечения (ПО)	Методы тестирования, верификации и валидации ПО. Особенности тестирования объектно-ориентированных систем и проектирование тестов.
7	Тенденции развития технологий разработки ПО	Современные тенденции: DevOps, CI/CD, микросервисы, автоматизация тестирования, использование ИИ в разработке ПО.
8	Средства оценки качества программных продуктов	Методы и метрики оценки качества ПО (функциональность, надежность, производительность). Применение стандартов (ISO/IEC 25010).

4.3 Перечень лабораторных работ

Лабораторных работ не предусмотрено

4.4 Перечень практических занятий

Семестр № 4

№	Темы практических (семинарских) занятий	Кол-во академических часов
1	Анализ кейса по выбору модели жизненного цикла	4
2	Сравнение подходов к разработке ПО	4
3	Построение UML-диаграмм вариантов использования	4
4	Разработка UML-диаграмм классов и последовательностей	4
5	Анализ требований для программной системы	4
6	Реализация паттерна Singleton на Python	4
7	Реализация паттерна Observer на Python	4
8	Тестирование объектно-ориентированного ПО на Python	4
9	Анализ современных тенденций разработки ПО	4

4.5 Самостоятельная работа

Семестр № 4

№	Вид СРС	Кол-во академических часов
1	Выполнение тренировочных и обучающих тестов	4
2	Написание отчета	4
3	Подготовка к зачёту	4
4	Подготовка презентаций	4
5	Решение специальных задач	20

В ходе проведения занятий по дисциплине используются следующие интерактивные методы обучения: Для вовлечения студентов и формирования компетенций в области разработки ПО применяются следующие интерактивные методы: 1. Проблемная лекция: Анализ реальных задач разработки ПО (например, выбор модели жизненного цикла) с обсуждением и выводами студентов. 2. Задания-провокации: Поиск и исправление ошибок в UML-диаграммах или коде (например, нарушение инкапсуляции в C++/Java). 3. Лекция-визуализация: Демонстрация UML-диаграмм или процесса тестирования с использованием StarUML и видео. 4. Кейс-технологии: Решение кейсов по проектированию ПО, включая создание UML-диаграмм для систем с ИИ. 5. Групповая работа: Командная разработка программы с применением паттернов проектирования (например, Singleton) и защитой результатов.

5 Перечень учебно-методического обеспечения дисциплины

5.1 Методические указания для обучающихся по освоению дисциплины

5.1.1 Методические указания для обучающихся по практическим занятиям

Дисциплина предусматривает лекции раз в две недели, лабораторные работы раз в две недели. Изучение курса завершается экзаменом.

Успешное изучение курса требует посещения лекций, активной работы на лабораторных работах, выполнения всех учебных заданий преподавателя, ознакомления с основной и дополнительной литературой.

Во время лекции студент должен вести краткий конспект.

Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий. При этом необходимо пометить материалы конспекта, которые вызывают затруднения для понимания. При этом обучающийся должен стараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если ему самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или ближайшей лекции.

Обучающемуся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

5.1.2 Методические указания для обучающихся по самостоятельной работе:

При подготовке к зачёту в дополнение к изучению конспектов лекций, учебных пособий и слайдов, необходимо пользоваться учебной литературой, рекомендованной к настоящей программе. При подготовке к экзамену нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала и самостоятельно решить по несколько типовых задач из каждой темы. При решении задач всегда необходимо

уметь качественно интерпретировать итог решения.

Самостоятельная работа студентов (СРС) по дисциплине играет важную роль в ходе всего учебного процесса. Методические материалы и рекомендации для обеспечения СРС готовятся преподавателем и являются неотъемлемой частью программы.

6 Фонд оценочных средств для контроля текущей успеваемости и проведения промежуточной аттестации по дисциплине

6.1 Оценочные средства для проведения текущего контроля

6.1.1 семестр 4 | Решение задач

Описание процедуры.

Студенты решают задачи, связанные с построением UML-диаграмм (вариантов использования, классов, последовательностей) в StarUML или разработкой тестовых сценариев для проверки ПО с использованием Python (библиотека unittest) в Anaconda. Решения оформляются в виде отчета (MS Word, ГОСТ 7.32-2017), включающего описание задачи, результат (диаграммы/сценарии) и пояснения. Отчет загружается на <https://el.istu.edu/>. Проверка проводится преподавателем с возможной устной защитой.

Критерии оценивания.

Корректность и полнота решения задачи (диаграммы или тестовые сценарии) (40%).

Соответствие решения требованиям задания (30%).

Качество оформления отчета (20%).

Обоснование подхода при защите или в пояснениях (10%).

6.1.2 семестр 4 | Отчет по лабораторной работе

Описание процедуры.

Студенты выполняют лабораторные работы, включающие анализ требований и проектирование ПО с использованием UML-диаграмм (в StarUML) или реализацию паттернов проектирования (Singleton, Observer) на Python в Anaconda с модульным тестированием (unittest). Результаты оформляются в отчет (MS Word, ГОСТ 7.32-2017), содержащий цель, описание, результаты (диаграммы/код/тесты) и выводы. Отчет загружается на <https://el.istu.edu/>. Защита проводится в форме презентации или устного доклада.

Критерии оценивания.

Точность и полнота выполненного задания (UML-диаграммы, код, тесты) (40%).

Соответствие результатов требованиям технического задания (30%).

Качество оформления отчета и документации (20%).

Обоснование решений на защите (10%).

6.1.3 семестр 4 | Доклад

Описание процедуры.

Студенты готовят доклад на тему современных тенденций разработки ПО (DevOps, CI/CD, микросервисы, использование ИИ). Доклад оформляется в виде презентации (MS PowerPoint) и сопровождается кратким отчетом (MS Word, ГОСТ 7.32-2017). Материалы загружаются на <https://el.istu.edu/>. Доклад представляется устно (5–7 минут) с ответами на вопросы.

Критерии оценивания.

Полнота и актуальность содержания доклада (40%).

Логичность и структура презентации (30%).

Качество оформления материалов (презентация и отчет) (20%).

Уверенность и ясность изложения, ответы на вопросы (10%).

6.1.4 семестр 4 | Устный опрос

Описание процедуры.

Студенты отвечают устно на вопросы по структурному и объектно-ориентированному подходам к разработке ПО, включая принципы декомпозиции и их применение. Подготовка осуществляется с использованием лекционных материалов и ресурсов на <https://el.istu.edu/>. Опрос проводится индивидуально в аудитории.

Критерии оценивания.

Точность и полнота ответов по подходам и декомпозиции (50%).
Понимание преимуществ и недостатков подходов (30%).
Ясность и логичность изложения (20%).

6.1.5 семестр 4 | Тест

Описание процедуры.

Студенты выполняют тест, включающий вопросы по жизненному циклу ПО (модели, этапы, применение) и метрикам качества ПО (функциональность, надежность, стандарты ISO/IEC 25010). Тест проводится в письменной или электронной форме через <https://el.istu.edu/>. Подготовка основана на лекциях и рекомендованной литературе.

Критерии оценивания.

Правильность ответов на вопросы по моделям жизненного цикла и метрикам качества (50%).
Полнота и точность теоретических знаний (30%).
Корректность использования терминологии (20%).

6.2 Оценочные средства для проведения промежуточной аттестации

6.2.1 Критерии и средства (методы) оценивания индикаторов достижения компетенции в рамках промежуточной аттестации

Индикатор достижения компетенции	Критерии оценивания	Средства (методы) оценивания промежуточной аттестации
ПКС-3.1	Способность разрабатывать корректные и полные тестовые сценарии. Умение применять процедуры тестирования для обеспечения работоспособности ПО. Качество анализа и интерпретации результатов тестирования.	Практическое задание: разработка и выполнение тестового сценария для заданной программной системы. Защита лабораторной работы с демонстрацией результатов тестирования. Письменный тест по методам

		тестирования и верификации ПО.
ПКС-4.1	Полнота и точность анализа требований к ПО. Способность составлять структурированные спецификации требований. Умение использовать профессиональную терминологию в документации.	Письменный тест: вопросы по функциональным и нефункциональным требованиям к ПО. Практическое задание: составление спецификации требований на основе кейса. Анализ технического задания на английском языке с переводом ключевых терминов.
ПКС-5.2	Полнота и корректность UML-диаграмм, отражающих решение задачи. Соответствие формализованных описаний требованиям технического задания. Качество и структурированность пояснительной записки.	Практическое задание: разработка UML-диаграмм для заданного технического задания. Защита пояснительной записки с демонстрацией формализованного описания. Письменный тест по правилам построения UML-диаграмм.
ПКС-1.2	Способность четко формулировать цели исследования, соответствующие принципам SMART. Корректность и полнота декомпозиции целей на подзадачи. Умение обосновать выбор методов и инструментов для достижения целей исследования.	Исследовательское задание: разработка плана исследования с формулировкой цели и ее декомпозицией на подзадачи. Презентация результатов декомпозиции с обоснованием

		подхода. Письменный отчет с описанием целей и структуры исследования.
ПКС-1.3	Способность выявить и обосновать области применения результатов исследования. Полнота и точность анализа релевантности результатов для профессиональной деятельности. Качество рекомендаций по внедрению результатов в практику.	Практическое задание: анализ кейса с определением сфер применения результатов исследования. Письменный отчет с рекомендациями по использованию результатов исследования в профессиональной деятельности. Устный опрос по примерам применения исследований в области ИИ и разработки ПО.
ПКС-4.3	Способность точно определять причины ошибок на основе сообщений. Корректность классификации ошибок и обоснованность предложенных решений. Качество документирования процесса анализа и устранения ошибок.	Практическое задание: анализ логов программы с составлением отчета об ошибках. Защита лабораторной работы с демонстрацией процесса диагностики ошибок. Устный опрос по типам ошибок и методам их анализа.
ПКС-5.4	Способность структурировать и выполнять этапы проектной работы. Качество реализации проектных задач и соответствие их целям проекта. Умение работать в команде и защищать результаты проектной работы.	Практическое задание: разработка и реализация небольшого проекта с использованием

		ООП и UML. Защита проектной работы с демонстрацией результатов и документации. Презентация плана проекта с описанием методологии управления.
--	--	--

6.2.2 Типовые оценочные средства промежуточной аттестации

6.2.2.1 Семестр 4, Типовые оценочные средства для проведения зачета по дисциплине

6.2.2.1.1 Описание процедуры

Студенты сдают зачет в форме устного или письменного ответа на вопросы из билета, охватывающие все темы дисциплины. Подготовка проводится с использованием лекций, практических материалов и ресурсов на <https://el.istu.edu/>. Зачет проводится в аудитории или онлайн через el.istu.edu.

Вопросы к билету:

1. Опишите этапы жизненного цикла ПО и сравните каскадную и спиральную модели.
2. Объясните различия между структурным и объектно-ориентированным подходами к разработке ПО.
3. Назовите основные типы UML-диаграмм и их назначение. Постройте диаграмму вариантов использования для заданной системы.
4. Опишите процесс анализа и проектирования ПО, включая роль UML.
5. Реализуйте паттерн Singleton на Python и объясните его применение.
6. Опишите методы тестирования ПО (модульное, интеграционное, системное) и их особенности.
7. Перечислите современные тенденции разработки ПО (DevOps, микросервисы) и их преимущества.
8. Объясните метрики качества ПО по стандарту ISO/IEC 25010.

6.2.2.1.2 Критерии оценивания

Зачтено	Не зачтено
Корректные и полные ответы на вопросы билета (не менее 70% правильных)	Менее 70% правильных ответов или значительные ошибки в ключевых

ответов).	понятиях.
Демонстрация понимания теоретических основ и практических навыков (UML, Python, тестирование).	Отсутствие понимания основных тем (жизненный цикл, UML, ООП, тестирование).
Четкое изложение и использование профессиональной терминологии.	Неспособность объяснить решения или применить знания на практике.

7 Основная учебная литература

1. Теория информационных процессов и систем. Моделирование информационных систем на языке UML [Электронный ресурс] : методические указания к лабораторному практикуму и самостоятельной работе студентов / Иркут. гос. техн. ун-т, 2008. - 56.
2. Сосинская С. С. Технологии разработки программного обеспечения : электронный курс / С. С. Сосинская, 2020

8 Дополнительная учебная литература и справочная

1. Рамбо Дж. UML 2.0. Объектно-ориентированное моделирование и разработка / Дж. Рамбо, М. Блаха, 2007. - 540.
2. Полонников Револьд Иванович. Методы оценки показателей надежности программного обеспечения / Револьд Иванович Полонников, Алексей Владимирович Никандров, 1992. - 77.
3. Игнатьев А. В. Тестирование программного обеспечения : учебное пособие / А. В. Игнатьев, 2023. - 56.
4. Дэвис К. Шаблоны проектирования для облачной среды. Проектирование программного обеспечения, устойчивого к изменениям / К. Дэвис ; пер. с англ. Д. А. Беликов, 2020. - 388.

9 Ресурсы сети Интернет

1. <http://library.istu.edu/>
2. <https://e.lanbook.com/>

10 Профессиональные базы данных

1. <http://new.fips.ru/>
2. <http://www1.fips.ru/>

11 Перечень информационных технологий, лицензионных и свободно распространяемых специализированных программных средств, информационных справочных систем

1. Microsoft Windows High Performance Computing (HPC) Server 2008
2. Microsoft Office Professional Plus 2013

12 Материально-техническое обеспечение дисциплины

1. Нойтбук Dell Alenware 13 Core i5 6200U 8Gb.1Tb,nVidia GeForce GTX 960M 2Gb,13,3,