

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ  
«ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР  
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ ИМ. М.В. КЕЛДЫША  
РОССИЙСКОЙ АКАДЕМИИ НАУК»**

---

**Утверждена**  
Ученым советом  
ИПМ им. М.В. Келдыша РАН,  
протокол № 14-22 от «10» ноября 2022 г.

# **РАБОЧАЯ ПРОГРАММА**

**УЧЕБНОЙ ДИСЦИПЛИНЫ:**  
**«Технологии разработки программного обеспечения»**

**Научная специальность:**  
**2.3.5 – «Математическое обеспечение вычислительных систем,  
комплексов и компьютерных сетей»**

**Форма обучения**  
очная

**Дисциплина:** «Технологии разработки программного обеспечения»

**Научная специальность:**

2.3.5 – «Математическое обеспечение вычислительных систем, комплексов и компьютерных сетей»

**Форма обучения:** очная

**ИСПОЛНИТЕЛЬ** (разработчик программ):

Якобовский Михаил Владимирович, ИПМ им. М.В. Келдыша РАН, заместитель директора, член-корреспондент РАН.

**РЕЦЕНЗЕНТ:** Крюков Виктор Алексеевич, ИПМ им. М.В. Келдыша РАН, заведующий отделом, доктор физико-математических наук, профессор

**РАБОЧАЯ ПРОГРАММА РЕКОМЕНДОВАНА**

Ученым советом ИПМ им. М.В. Келдыша РАН,  
протокол № 14/22 от «10» ноября 2022 г.

Заведующий аспирантурой \_\_\_\_\_ / Меньшов И.С. /

## Оглавление

АННОТАЦИЯ	4
1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	4
2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	4
3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	6
3.1. Структура дисциплины	6
3.2. Содержание разделов дисциплины	6
3.3. Семинарские занятия	6
4. ТЕКУЩАЯ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ	7
5. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	8
6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	8

## 1 АННОТАЦИЯ

Рабочая программа дисциплины «Технологии разработки программного обеспечения» разработана и составлена на основании ФГТ - «Федеральные государственные требования к структуре программ подготовки научных и научно-педагогических кадров в аспирантуре (адъюнктуре), условиям их реализации, срокам освоения этих программ с учетом различных форм обучения, образовательных технологий и особенностей отдельных категорий аспирантов (адъюнктов)» (Приказ Минобрнауки № 951 от 20.10.2021г.), в соответствии с учебными планами подготовки аспирантов ФГУ «Федеральный исследовательский центр «Институт прикладной математики им. М.В. Келдыша РАН (ИПМ им. М.В. Келдыша РАН) по научным специальностям 2.3.5 - «Математическое обеспечение вычислительных систем, комплексов и компьютерных сетей».

Дисциплина «Технологии разработки программного обеспечения» реализуется в рамках Блока «Образовательный компонент программы подготовки научных и научно - педагогических кадров в аспирантуре ИПМ им. М.В. Келдыша РАН.

Источником материалов для формирования содержания программы являются: материалы конференций, симпозиумов, семинаров, Интернет-ресурсы, научные издания и монографические исследования и публикации.

Общая трудоемкость дисциплины по учебному плану составляет 2 зач.ед. (72 часа), из них лекций – 4 часа, семинарских занятий – 10 часов, практических занятий – 0 часов и самостоятельной работы – 58 часа. Дисциплина реализуется на 2-м курсе, в 1-м семестре, продолжительность обучения – 1 семестр.

Текущая аттестация проводится не менее 2 раз в соответствии с заданиями и формами контроля, предусмотренными настоящей программой.

Промежуточная оценка знания осуществляется в период зачетно-экзаменационной сессии в форме экзамена.

## 2 ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

**Цели и задачи дисциплины «Технологии разработки программного обеспечения»**

**Цель:** освоение фундаментальных знаний и компетенций, которые позволят участвовать в процессах разработки программного обеспечения, а также в проведении научных исследований (экспериментов, наблюдений и количественных измерений) программных продуктов, проектов, процессов, методов и инструментов программной инженерии.

**Задачи:**

- освоить фундаментальные концепции программной инженерии;
- освоить методологии разработки программного обеспечения;
- практическое освоение накопленных по дисциплине знаний при решении профессиональных проблем в реальных (смоделированных) условиях;
- стимулирование к самостоятельной деятельности по освоению дисциплины и формированию необходимых компетенций.

## 3 ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс изучения дисциплины «Технологии разработки программного обеспечения» направлен на формирование определенных знаний, умений и компетенций.

**а) универсальные (УК):** не предусмотрено

**б) общепрофессиональных (ОПК):** не предусмотрено

**в) профессиональных (ПК):** Способность использовать языки программирования и системы программирования (ПК-1), Способность создавать модели и алгоритмы проектирования программных систем (ПК-2).

В результате освоения дисциплины обучающийся должен:

**Иметь представление:**

- о видах деятельности, выполняемых в процессе промышленного программирования и необходимых для успешного выполнения заказов

- о методах обеспечения качества программного обеспечения

**Знать:**

- фундаментальные концепции процесса разработки программного обеспечения, архитектуры программного обеспечения, управления требованиями, конфигурационного управления, тестирования и документирования программного обеспечения;

- основы лицензирования программного обеспечения

- основные виды тестирования программного обеспечения;

- основные методологии разработки программного обеспечения;

**Уметь:**

- строить модели программных проектов и программных продуктов

- работать с системами конфигурационного управления;

**Владеть:**

- инструментальными средствами компьютерного моделирования программных продуктов;

- инструментальными средствами конфигурационного управления;

**Приобрести опыт:**

- для естественнонаучных задач построения математической формулировки

- построения алгоритма решения формализованной задачи и его анализ

## 4 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### *Структура дисциплины*

#### Распределение трудоемкости дисциплины по видам учебных работ

Вид учебной работы	Трудоемкость	
	общая	
	зач.ед.	час.
<b>ОБЩАЯ ТРУДОЕМКОСТЬ</b> по Учебному плану	<b>2</b>	<b>72</b>
Лекции (Л)		<b>4</b>
Практические занятия (ПЗ)	-	-
Семинары (С)		<b>10</b>
Самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к семинарским и практическим занятиям) и самостоятельное изучение тем дисциплины		<b>22</b>
Подготовка к экзамену (проработка и повторение лекционного материала и материала учебников и учебных пособий)		<b>36</b>
<i>Вид контроля: экзамен</i>		

### 4.2 *Содержание разделов дисциплины*

#### Общее содержание дисциплины

№ раздела	Наименование раздела	Содержание раздела	Форма текущей аттестации
-----------	----------------------	--------------------	--------------------------

1.	Введение в дисциплину. Процесс разработки ПО. Управление Требованиями	Понятие “Технологии разработки программного обеспечения”. Программное обеспечение (ПО): определение, свойства. Жизненный цикл ПО. Основные фазы жизненного цикла ПО. Модели жизненного цикла ПО. Понятие процесса разработки ПО. Универсальный процесс. Текущий процесс. Конкретный процесс. Стандартный процесс. Совершенствование процесса. Pull/Push стратегии. Классические модели процесса: водопадная модель, спиральная модель. Фазы и виды деятельности. Управление требованиями. Виды требований: функциональные требования, нефункциональные требования. Свойства требований: ясность и недвусмысленность, полнота и непротиворечивость, необходимый уровень детализации, прослеживаемость, тестируемость и проверяемость, модифицируемость. Формализация требований. Цикл работы с требованиями.	О
2.	Архитектура ПО	Архитектура ПО. Понятие архитектуры ПО. Точка зрения и характеристики точек зрения. Множественность точек зрения при разработке ПО.	О, ДЗ
3.	Конфигурационное управление. Лицензирование ПО.	Понятие конфигурационного управления. Управление версиями. Понятие "ветки" проекта. Управление сборками. Средства версионного контроля. Единицы конфигурационного управления. Понятие baseline. Классификация ПО. Виды лицензий ПО.	О, ДЗ
4.	Обеспечение качества ПО. Документирование ПО	Тестирование ПО. Стандартизация качества. Методы обеспечения качества ПО. Понятие тестирования. Тестирование черного ящика. Тестирование белого ящика. Инструменты тестирования. Критерии тестирования. Виды тестирования. Работа с ошибками. Средства контроля ошибок (bug tracking systems). Контрактное программирование. Основные положения. Обработка исключений. Оценка оттестированности проекта. Метрики и методика интегральной оценки. Модульное и интеграционное тестирование. Модульное и интеграционное тестирование. Интеграционное тестирование и его особенности для объектно-ориентированного программирования. Разновидности тестирования. Системное тестирование. Регрессионное тестирование. Автоматизация тестирования. Издержки тестирования. Индустриальное тестирование. Особенности индустриального тестирования. Документирование и оценка индустриального тестирования. Регрессионное тестирование. Цели и задачи. Условия применения. Классификация тестов и методов отбора. Документирование ПО. Виды программной документации.	О, ДЗ
5.	Методологии разработки ПО.	Обзор методологий разработки ПО. MSF. IT решение. Основные принципы MSF. Модель команды: основные принципы, ролевые кластеры. Масштабирование команды MSF. Модель процесса. Управление компромиссами. СММІ. Понятие СММІ. Уровни зрелости процессов по СММІ. Области усовершенствования. "Гибкие" (agile) методы разработки. Общее описание "гибких" методов разработки ПО. Extreme Programming: общее описание, основные принципы организации процесса. Scrum: общее описание, роли, практики.	О

**Примечание:** О – опрос, Д – дискуссия (диспут, круглый стол, мозговой штурм, ролевая игра), ДЗ – домашнее задание (эссе и пр.). Формы контроля не являются жесткими и могут быть заменены преподавателем на другую форму контроля в зависимости от контингента обучающихся. Кроме того, на занятиях семинарских может проводиться работа с нормативными документами, изданиями средств информации и прочее, что также оценивается преподавателем.

#### 4.3 Лекционные занятия

№ занятия	№ Раздела	Краткое содержание темы занятия	Кол-во часов
1.	1	Понятие “Технологии разработки программного обеспечения”. Программное обеспечение (ПО): определение, свойства. Жизненный цикл ПО. Основные фазы жизненного цикла ПО. Модели жизненного цикла ПО. Понятие процесса разработки ПО. Универсальный процесс. Текущий процесс. Конкретный процесс. Стандартный процесс. Совершенствование процесса. Pull/Push стратегии. Классические модели процесса: водопадная модель, спиральная модель. Фазы и виды деятельности. Управление требованиями. Виды требований: функциональные требования, нефункциональные требования. Свойства требований: ясность и недвусмысленность, полнота и непротиворечивость, необходимый уровень детализации, прослеживаемость, тестируемость и проверяемость, модифицируемость. Формализация требований. Цикл работы с требованиями. Язык UML. Типы UML диаграмм.	2
2.	3,4	Классификация ПО. Виды лицензий ПО. Тестирование ПО. Стандартизация качества. Методы обеспечения качества ПО. Понятие тестирования. Инструменты тестирования. Виды тестирования. Средства контроля ошибок. Оценка оттестированности проекта. Модульное и интеграционное тестирование. Системное тестирование. Регрессионное тестирование. Автоматизация тестирования.	2
<b>ВСЕГО</b>			<b>4</b>

#### 4.4 Семинарские занятия

№ занятия	№ Раздела (темы)	Краткое содержание темы занятия	Кол-во часов
3.	2	Практикум по инструментам компьютерного моделирования программных продуктов	2
4.	3	Практикум по инструментальными средствами конфигурационного управления	2
5.	4	Практикум по тестированию программного обеспечения. Модульное тестирование	2
6.	4	Практикум по тестированию программного обеспечения. Интеграционное тестирование	2
7.	5	Разбор методологий разработки ПО	2
<b>ВСЕГО</b>			<b>10</b>

### 5 ТЕКУЩАЯ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

**Текущая аттестация аспирантов.** Текущая аттестация аспирантов проводится в соответствии с локальным актом ИПМ им. М.В. Келдыша РАН - Положением о текущей, промежуточной и итоговой аттестации аспирантов ИПМ им. М.В. Келдыша РАН по программам высшего образования – программам подготовки научно-педагогических кадров в аспирантуре и является обязательной.

Текущая аттестация по дисциплине проводится в форме опроса, а также оценки вопроса-ответа в рамках участия обучающихся в дискуссиях и различных контрольных мероприятиях по оцениванию фактических результатов обучения, осуществляемых преподавателем, ведущим дисциплину. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины см. ниже.

Объектами оценивания выступают:

- учебная дисциплина – активность на занятиях, своевременность выполнения различных видов заданий, посещаемость занятий;
- степень усвоения теоретических знаний и уровень овладения практическими умениями и навыками по всем видам учебной работы, проводимых в рамках семинаров, практических занятий и самостоятельной работы.

Оценивание обучающегося на занятиях осуществляется с использованием нормативных оценок по 4-х бальной системе (5-отлично, 4-хорошо, 3-удовлетворительно, 2-неудовлетворительно).

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Форма контроля знаний	Вид аттестации	Примечание
проверка домашних заданий, выполненных в течение курса	текущая	Ниже приведены перечни рекомендуемых домашних заданий
проведение проверочного тестирования	промежуточная	Ниже приведены перечни рекомендуемых вопросов
экзамен	итоговая	

Примерный перечень рекомендуемых **домашних заданий** для оценки текущего уровня успеваемости студента:

1. Нарисовать диаграмму одного из следующих паттернов объектно-ориентированного программирования (Singleton , Factory , Factory Method , Builder , Prototype , Object Pool , Object Command , Object Iterator , Object Stratify , Object Null Object , Object Adapter , Object Proxy)

2. Нарисовать UML-диаграмму одного из следующих типов (для заданной преподавателем модельной системы ПО): классов (class diagram), компонент (component diagram), объектов (object diagram), композитных структур (composite structure diagram), развертывания (deployment diagram), пакетов (package diagram), активностей (activity diagram), случаев использования (use case diagram), конечных автоматов (state machine diagram), последовательностей (sequence diagram), схем взаимодействия (interaction overview diagram), коммуникаций (communication diagram)временные диаграмму (timing diagram)

3. Для некоторого модельного проекта, состоящего из набора текстовых файлов осуществить моделирование жизненного цикла проекта в системе контроля версий Git (по заданному преподавателем сценарию) с использованием следующих операций — создание репозитория, добавление и коммит файлов в репозиторий, синхронизация с удалённым репозиторием, создание новой ветки проекта, слияние веток и разрешение конфликтов, перестройка истории коммитов, создание submodule, тегирование версий, создание релизов.

4. Для некоторого модульного модельного проекта, заданного преподавателем, написать файл CMakeLists.txt, в котором будет реализована сборка данного проекта с использованием системы CMake, в которой будут осуществлена работа с внутренними и внешними зависимостями

5. Реализовать ряд юнит-тестов с использованием фреймворка googletest для одного из



следующих наборов функций из стандартной библиотеки C++ на различных входных данных и/или с различными параметрами. Убедиться, что функции работают в соответствии с документацией:

- 5.1. `std::sort`, `std::stable_sort`, `std::is_sorted`, `std::max`, `std::min`
- 5.2. `std::make_heap`, `std::push_heap`, `std::pop_heap`
- 5.3. `std::find`, `std::find_end`, `std::find_first_of`, `std::find_if`, `std::find_if_not`
- 5.4. `std::set_difference`, `std::set_intersection`, `std::set_symmetric_difference`, `std::set_union`
- 5.5. `std::binomial_distribution`, `std::bernoulli_distribution`
- 5.6. Функции `sin`, `sinh`, `cos`, `cosh`, `tan`, `tanh`, `acos`, `acosh`, `asin`, `asinh` из `<math.h>`
- 5.7. Функции для работы с комплексными числами из стандартной библиотеки

6. Реализовать ряд юнит-тестов с использованием фреймворка googletest для одного следующих классов (сами процедуры реализовывать не нужно, только интерфейс):

- 6.1. Класса очереди (`queue`) с функциями `pop`, `push`
- 6.2. Класса кучи (`heap`) с функциями `pop`, `push`
- 6.3. Класса бинарного дерева с функциями `pop`, `push`, `search`

Примерный перечень рекомендуемых **вопросов для промежуточной** оценки текущего уровня успеваемости студента:

1. Напишите определение термина «программное обеспечение»
2. Какими свойствами обладает программное обеспечение и чем отличается от других видов систем, создаваемых человеком?
3. Дайте определение понятия «процесс создания ПО»
4. Что такое «универсальный процесс» разработки ПО?
5. Что такое «текущий процесс» разработки ПО?
6. Что такое «конкретный процесс» разработки ПО?
7. Что такое «стандартный процесс» разработки ПО?
8. В чем заключается деятельность по изменению существующего процесса компании, каковы её цели? В чём отличие Push и Pull стратегий внедрения инноваций в производственные процессы бизнес-компаний?
9. Дайте определение «модели процесса»
10. Дайте определение «фазы» процесса
11. Дайте определение «вида деятельности» в процессе разработки ПО
12. В чём заключается «Водопадная модель» процесса разработки ПО?
13. В чём заключается «Спиральная модель» процесса разработки ПО?
14. Что является «Рабочим продуктом» процесса разработки ПО?
15. Дайте определение понятия «Проекта» по разработке ПО
16. В чём заключается вид деятельности «Управление проектами»?
17. Дайте определение понятий «Stakeholders», «Project score» и «Компромиссов» в терминах управления проектами
18. Дайте определение понятия «Архитектура ПО»
19. В чём заключается множественность точек зрения на архитектуру ПО. В чём её причина?
20. Какие существуют типы структурных UML диаграмм?
21. Какие существуют типы поведенческих UML диаграмм?
22. Чем характеризуются функциональные требования?
23. Чем характеризуются нефункциональные требования?
24. Какие существуют свойства требований?
25. Какие существуют варианты формализации требований?

26. Какие виды деятельности осуществляются при работе с требованиями?
27. В чём заключается деятельность по конфигурационному управлению? Чем вызвана её необходимость?
28. В чём заключается управление версиями?
29. В чём заключается управление сборками?
30. Что является единицей конфигурационного управления?
31. Что такое система управления версиями?
32. Дайте определение понятию «Baseline»
33. Какие существуют методы обеспечения качества ПО?
34. Что такое тестирование чёрного ящика и белого ящика?
35. Чем вызвана необходимость к определению критерий тестирования
36. В чём заключается модульное тестирование?
37. В чём заключается системное тестирование?
38. В чём заключается регрессионное тестирование?
39. В чём заключается нагрузочное тестирование?
40. В чём заключается стрессовое тестирование?
41. В чём заключается приёмочное тестирование?
42. Что такое «средство контроля ошибок» (bug tracking system)?
43. Опишите основные принципы MSF
44. Опишите основные принципы CMMI
45. Чем характеризуются «гибкие» (agile) методологии разработки ПО? В чём их отличие от классических методологий?
46. В чём заключается методология Scrum? Какие существуют практики в рамках данной методологии?
47. В чём заключается методология Extreme Programming?

**Итоговая аттестация аспирантов.** Итоговая аттестация аспирантов по дисциплине проводится в соответствии с локальным актом ИПМ им. М.В. Келдыша РАН – Положением о текущей, промежуточной и итоговой аттестации аспирантов ИПМ им. М.В. Келдыша РАН по программам высшего образования – программам подготовки научно-педагогических кадров в аспирантуре и является обязательной.

Итоговая аттестация по дисциплине осуществляется в форме экзамена в период зачетно-экзаменационной сессии в соответствии с Графиком учебного процесса. Обучающийся допускается к зачету в случае выполнения аспирантом всех учебных заданий и мероприятий, предусмотренных настоящей программой. В случае наличия учебной задолженности (пропущенных занятий и (или) невыполненных заданий) аспирант отрабатывает пропущенные занятия и выполняет задания.

Оценивание обучающегося на промежуточной аттестации осуществляется с использованием нормативных оценок на экзамене – неудовлетворительно, удовлетворительно, хорошо, отлично.

#### Оценивание аспиранта на итоговой аттестации в форме экзамена

Оценка	Требования к знаниям и критерии выставления оценок
Неудовлетворительно	основное содержание учебного материала не раскрыто; допущены грубые ошибки в определении понятий и при использовании терминологии; не даны ответы на дополнительные вопросы.
Удовлетворительно	владение основным объемом знаний по дисциплине; проявляются затруднения в самостоятельных ответах, неточные формулировки; в процессе ответов допускаются ошибки по существу вопросов; экзаменуемый способен решать лишь наиболее легкие задачи,

	владеет только обязательным минимумом базовых концепций.
Хорошо	владение знаниями дисциплины почти в полном объеме программы (имеются пробелы знаний только в некоторых, особенно сложных разделах); самостоятельно и отчасти при наводящих вопросах даются полноценные ответы на вопросы билета; не всегда выделяется наиболее существенное, но нет серьезных ошибок в ответах; экзаменуемый умеет решать легкие и средней тяжести задачи;
Отлично	владение знаниями предмета в полном объеме учебной программы, достаточно глубокое осмысление дисциплины; экзаменуемый самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы билета, подчеркивал при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделять в нем главное: устанавливать причинно-следственные связи;

## 6 УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### *Основная литература*

1. Е. А. Жоголев. Технология программирования / Москва : Научный Мир, 2004. - 216 с., ISBN 5-89176-265-X
2. М. М. Горбунов-Посадов. Расширяемые программы. Москва : Политих, 1999. - 336 с., ISBN 5-901118-01-4

### *Дополнительная литература и Интернет-ресурсы*

1. В. В. Бахтизин, Л. А. Глухова. Технология разработки программного обеспечения : учеб. пособие / Минск : БГУИР, 2010. – 267 с. : ил., ISBN 978-985-488-512-4
2. Джеймс Рамбо, М. Блаха. UML 2.0. Объектно-ориентированное моделирование и разработка / Питер, 2007. — 544 с., ISBN 5-469-00814-2
3. Бертран Мейер. Объектно-ориентированное конструирование программных систем, 2-е издание / Русская редакция, 2005. - 1204 с., ISBN 5-7502-0255-0
4. Технологии командной разработки программного обеспечения информационных систем. <http://www.intuit.ru/studies/courses/4806/1054/lecture/16115>
5. Дисциплина «Технологии разработки программного обеспечения» / Кафедра КСПТСПБПУ, 2015. <http://aivt.ftk.spbstu.ru/course/se>
6. Курс “Введение в программную инженерию” на сайте INTUIT.ru <http://www.intuit.ru/studies/courses/497/353/info>
7. Курс “Основы тестирования программного обеспечения” на сайте INTUIT.ru <http://www.intuit.ru/studies/courses/48/48/info>

## 7 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для обеспечения интерактивных методов обучения для чтения лекций требуется аудитория с мультимедиа (возможен вариант с интерактивной доской).

Для проведения дискуссий и круглых столов, возможно, использование аудиторий со специальным расположением столов и стульев.