

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
федеральное государственное автономное образовательное учреждение высшего  
образования  
«Балтийский федеральный университет имени Иммануила Канта»  
Высшая школа компьютерных наук и искусственного интеллекта

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Разработка программного обеспечения**

**Шифр: 02.03.02**

**Направление подготовки: Фундаментальная информатика и информационные технологии  
Профиль: Программная инженерия в искусственном интеллекте**

**Квалификация (степень) выпускника: бакалавр**

Калининград  
2023

## Лист согласования

**Составитель:** Верещагин Михаил Дмитриевич, к.ф.-м.н, директор Высшей школы компьютерных наук и искусственного интеллекта

Рабочая программа утверждена на заседании Учебно-методического совета (УМС)

Протокол № 33 от «27» октября 2023 г.

Профессор, д.ф.-м.н.,  
руководитель ОНК «Институт высоких технологий»

А.В. Юров

Директор высшей школы компьютерных наук  
и искусственного интеллекта

М.Д. Верещагин

Руководитель ОПОП ВО

С.С. Головин

## Содержание

1. Наименование дисциплины «Разработка программного обеспечения».
2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.
3. Место дисциплины в структуре образовательной программы.
4. Виды учебной работы по дисциплине.
5. Содержание дисциплины, в том числе практической подготовки в рамках дисциплины, структурированное по темам.
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.
7. Методические рекомендации по видам занятий
8. Фонд оценочных средств
  - 8.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы в рамках учебной дисциплины
  - 8.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности в процессе текущего контроля
  - 8.3. Перечень вопросов и заданий для промежуточной аттестации по дисциплине
  - 8.4. Планируемые уровни сформированности компетенций обучающихся и критерии оценивания
9. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

## 1. Наименование дисциплины: Разработка программного обеспечения.

**Цель** дисциплины: дать целостные знания о методологии разработки программного обеспечения с использованием языка UML. Дать основные знания об унифицированном процессе разработки программного обеспечения, его итеративности, о фазах процесса и потоках работ процесса, участниках потоков работ и их навыках. Показать значение архитектуры программной системы и ее раннего построения при прохождении итераций через потоки работ. Показать управление итерациями процесса с помощью прецедентов использования программной системы.

## 2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Код компетенции	Результаты освоения образовательной программы (ИДК)	Результаты обучения по дисциплине
ПК-11. Способен принимать участие в управлении проектами по созданию и развитию технологий и систем искусственного интеллекта на стадиях их жизненного цикла (соответствует ОПК-2 Модели)	ПК-11.1. Использует основы управления проектами по созданию и развитию технологий и систем искусственного интеллекта на стадиях их жизненного цикла ПК-11.2. Решает задачи управления проектами по созданию и развитию технологий и систем искусственного интеллекта на стадиях их жизненного цикла	<b>Знать:</b> основы управления проектами по созданию и развитию технологий и систем искусственного интеллекта на стадиях их жизненного цикла <b>Уметь:</b> управлять проектами по созданию и развитию технологий и систем искусственного интеллекта на стадиях их жизненного цикла <b>Владеть:</b> методами решений задач управления проектами по созданию и развитию технологий и систем искусственного интеллекта на стадиях их жизненного цикла

## 3. Место дисциплины в структуре образовательной программы

Разработка программного обеспечения представляет собой дисциплину формируемой участниками части (Б1.В.03) направления подготовки бакалавриата 02.03.02 «Фундаментальная информатика и информационные технологии», профиль «Программная инженерия в искусственном интеллекте».

## 4. Виды учебной работы по дисциплине.

Виды учебной работы по дисциплине зафиксированы учебным планом основной профессиональной образовательной программы по указанному направлению и профилю, выражаются в академических часах. Часы контактной работы и самостоятельной работы обучающегося и часы, отводимые на процедуры контроля, могут различаться в учебных планах ОПОП по формам обучения. Объем контактной работы включает часы контактной аудиторной работы (лекции/практические занятия/ лабораторные работы), контактной внеаудиторной работы (контроль самостоятельной работы), часы контактной работы в период аттестации. Контактная работа, в том числе может проводиться посредством электронной информационно-образовательной среды университета с использованием ресурсов сети Интернет и дистанционных технологий

## 5. Содержание дисциплины, структурированное по темам (разделам)

Исходя из рамок, установленных учебным планом по трудоемкости и видам учебной работы по дисциплине, преподаватель самостоятельно выбирает тематику занятий по формам и количеству часов проведения контактной работы: лекции и иные учебные занятия, предусматривающие преимущественную передачу учебной информации преподавателем и (или) занятия семинарского типа (семинары, практические занятия, практикумы, лабораторные работы, коллоквиумы и иные аналогичные занятия), и (или) групповые консультации, и (или) индивидуальную работу обучающихся с преподавателем, в том числе индивидуальные консультации (по курсовым работам/проектам – при наличии курсовой работы/проекта по данной дисциплине в учебном плане). Рекомендуемая тематика занятий максимально полно реализуется в контактной работе с обучающимися очной формы обучения. В случае реализации образовательной программы в заочной / очно-заочной форме трудоемкость дисциплины сохраняется, однако объем учебного материала в значительной части осваивается обучающимися в форме самостоятельной работы. При этом требования к ожидаемым образовательным результатам обучающихся по данной дисциплине не зависят от формы реализации образовательной программы.

№ п/п	Наименование разделов (тем) дисциплины	Содержание разделов (тем) дисциплин
1.	Тема 1. Основные принципы унифицированного процесса разработки ПО.	Управление процессом с помощью прецедентов использования системы. Понятие архитектуры системы. Ориентация унифицированного процесса на раннюю разработку архитектуры системы. Понятие итерации в разработке программной системы. Итеративность и инкрементальность унифицированного процесса. Поток работ процесса. Фазы процесса.
2.	Тема 2. Управление процессом с помощью прецедентов.	Определение прецедентов использования системы с помощью модели прецедентов на языке UML. Итерация как последовательность разработки моделей на языке UML. Связь различных моделей на языке UML. Пример разработки последовательности моделей, начиная с модели прецедентов.
3.	Тема 3. Разработка архитектуры программной систем.	Процесс ориентированный на разработку архитектуры системы. Необходимость описания архитектуры системы для организации работы участников процесса. Использование вариантов использования системы при разработке архитектуры. Этапы в разработке архитектуры системы. Пример описания архитектуры с помощью графической нотации языка UML.
4.	Тема 4. Итеративность и инкрементальность процесса.	Необходимость итеративной и инкрементальной разработки. Риски в разработке программной системы и их влияние на выбор итерации. Типовая итерация. Инкремент как результат итерации. Итерации в жизненном цикле системы. Развитие моделей программной системы в результате итераций. Пример развития моделей на языке UML.
5.	Тема 5. Поток работ для получения требований к системе.	Прецеденты использования системы. Модель прецедентов использования системы. Актеры как

		<p>средство представления пользователей и внешней среды программной системы. Прототип интерфейса пользователя. Глоссарий терминов используемых при сборе требований. Модель архитектуры, описанная с помощью диаграмм прецедентов языка UML. Участники потока работ по сбору требований: архитектор, системный аналитик, спецификатор сценариев использования, проектировщик интерфейса пользователя.</p> <p>Деятельности потока работ: поиск актеров и сценариев использования, определение приоритетов для сценариев использования, детализация сценариев использования, создание прототипа интерфейса пользователя, структурирование модели сценариев использования.</p>
6.	Тема 6. Поток работ по анализу системы.	<p>Роль модели анализа в жизненном цикле программного обеспечения. Артефакты потока работ: описание архитектуры системы, анализ класса, анализ реализации сценария использования, анализ пакетов. Участники потока работ: архитектор, разработчик прецедентов, разработчик компонентов. Деятельности потока работ: архитектурный анализ, анализ сценариев использования, анализ классов, анализ пакетов.</p>
7.	Тема 7. Поток работ по проектированию системы.	<p>Роль потока работ проектирования в жизненном цикле программного обеспечения. Артефакты потока работ: описание архитектуры (вид потока проектирования), проект класса, проект реализации прецедента, проект подсистемы, проект интерфейса, описание модели внедрения системы. Участники потока работ: архитектор, разработчик прецедентов, разработчик компонентов. Деятельности потока работ: проектирование архитектуры, проектирование прецедентов, проектирование классов, проектирование подсистем.</p>
8.	Тема 8. Поток работ по реализации системы.	<p>Роль потока работ по реализации в жизненном цикле программного обеспечения. Артефакты потока работ: описание архитектуры, компоненты, подсистема реализации, интерфейсы, план интеграции для реализации. Участники потока работ: архитектор, разработчик компонентов, системный интегратор. Деятельности потока работ: реализация архитектуры, интеграция системы, реализация подсистемы, реализация класса, выполнение тестирования для единицы компиляции.</p>
9.	Тема 9. Поток работ по тестированию системы	<p>. Роль потока работ по тестированию в жизненном цикле программного обеспечения. Артефакты потока работ: описание архитектуры, набор тестов, процедура тестирования, компонента тестирования, план тестирования, дефекты, оценка</p>

		теста. Участники потока работ: проектировщик теста, разработчик компонента, тестировщик интеграции, тестировщик системы. Деятельности потока работ: планирования теста, проектирование теста, реализация теста, выполнение тестирования интеграции, выполнение тестирования системы, оценка теста.
--	--	--

## 6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Рекомендуемая тематика учебных занятий *лекционного* типа (предусматривающих преимущественную передачу учебной информации преподавателями):

№ п/п	Наименование разделов (тем) дисциплины	Тема лекции
•	Тема 1. Основные принципы унифицированного процесса разработки ПО.	Лекция 1 «Основные принципы унифицированного процесса разработки ПО»
•	Тема 2. Управление процессом с помощью прецедентов.	Лекция 2 «Управление процессом с помощью прецедентов»
•	Тема 3. Разработка архитектуры программной систем.	Лекция 3 «Разработка архитектуры программной систем»
•	Тема 4. Итеративность и инкрементальность процесса.	Лекция 4 «Итеративность и инкрементальность процесса»
•	Тема 5. Поток работ для получения требований к системе.	Лекция 5 «Поток работ для получения требований к системе»
•	Тема 6. Поток работ по анализу системы.	Лекция 6 «Поток работ по анализу системы»
•	Тема 7. Поток работ по проектированию системы.	Лекция 7 «Поток работ по проектированию системы»
•	Тема 8. Поток работ по реализации системы.	Лекция 8 «Поток работ по реализации системы»
•	Тема 9. Поток работ по тестированию системы.	Лекция 9 «Поток работ по тестированию системы»

### Рекомендуемая тематика практических занятий:

1. Основные принципы унифицированного процесса разработки ПО.
2. Управление процессом с помощью прецедентов.
3. Разработка архитектуры программной систем.
4. Итеративность и инкрементальность процесса.
5. Поток работ для получения требований к системе.
6. Поток работ по анализу системы.
7. Поток работ по проектированию системы.
8. Поток работ по реализации системы.
9. Поток работ по тестированию системы.

### Требования к самостоятельной работе обучающихся

1. Работа с лекционным материалом, предусматривающая проработку конспекта лекций и учебной литературы, по всем темам из п. 6 настоящей рабочей программы.

2. Выполнение домашнего задания, предусматривающего решение задач, выполнение упражнений, выдаваемых на лабораторных занятиях, по всем темам из п. 6 настоящей рабочей программы.

Руководствуясь положениями статьи 47 и статьи 48 Федерального закона от 29 декабря 2012 г. N 273-ФЗ «Об образовании в Российской Федерации» научно-педагогические работники и иные лица, привлекаемые университетом к реализации данной образовательной программы, пользуются предоставленными академическими правами и свободами в части свободы преподавания, свободы от вмешательства в профессиональную деятельность; свободы выбора и использования педагогически обоснованных форм, средств, методов обучения и воспитания; права на творческую инициативу, разработку и применение авторских программ и методов обучения и воспитания в пределах реализуемой образовательной программы и отдельной дисциплины.

Исходя из рамок, установленных учебным планом по трудоемкости и видам учебной работы по дисциплине, преподаватель самостоятельно выбирает тематику занятий по формам и количеству часов проведения контактной работы: лекции и иные учебные занятия, предусматривающие преимущественную передачу учебной информации преподавателем и (или) занятия семинарского типа (семинары, практические занятия, практикумы, лабораторные работы, коллоквиумы и иные аналогичные занятия), и (или) групповые консультации, и (или) индивидуальную работу обучающихся с преподавателем, в том числе индивидуальные консультации (по курсовым работам/проектам – при наличии курсовой работы/проекта по данной дисциплине в учебном плане).

Рекомендуемая тематика занятий максимально полно реализуется в контактной работе с обучающимися очной формы обучения. В случае реализации образовательной программы в заочной / очно-заочной форме трудоемкость дисциплины сохраняется, однако объем учебного материала в значительной части осваивается обучающимися в форме самостоятельной работы. При этом требования к ожидаемым образовательным результатам обучающихся по данной дисциплине не зависят от формы реализации образовательной программы.

## **7. Методические рекомендации по видам занятий**

Лекционные занятия.

В ходе лекционных занятий обучающимся рекомендуется выполнять следующие действия. Вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации по их применению. Задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

Желательно оставить в рабочих конспектах поля, на которых во внеаудиторное время можно сделать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Лабораторные занятия.

На лабораторных занятиях в зависимости от темы занятия выполняется поиск информации по решению проблем, практические упражнения, контрольные работы, выработка индивидуальных или групповых решений, итоговое обсуждение с обменом знаниями, участие в круглых столах, разбор конкретных ситуаций, командная работа, представление портфолио и т.п.

Самостоятельная работа.

Самостоятельная работа осуществляется в виде изучения литературы, эмпирических данных по публикациям и конкретных ситуаций из практики, подготовке индивидуальных работ, работа с лекционным материалом, самостоятельное изучение отдельных тем дисциплины; поиск и обзор литературы и электронных источников; чтение и изучение учебника и учебных пособий.

## 8. Фонд оценочных средств

### 8.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы в рамках учебной дисциплины

Основными этапами формирования указанных компетенций при изучении обучающимися дисциплины являются последовательное изучение содержательно связанных между собой тем учебных занятий. Изучение каждой темы предполагает овладение обучающимися необходимыми компетенциями. Результат аттестации обучающихся на различных этапах формирования компетенций показывает уровень освоения компетенций.

Контролируемые разделы (темы) дисциплины	Индекс контролируемой компетенции (или её части)	Оценочные средства по этапам формирования компетенций
		текущий контроль по дисциплине
Тема 1. Основные принципы унифицированного процесса разработки ПО.	ПК-11	Опрос
Тема 2. Управление процессом с помощью прецедентов.	ПК-11	Опрос
Тема 3. Разработка архитектуры программной систем.	ПК-11	Опрос
Тема 4. Итеративность и инкрементальность процесса.	ПК-11	Опрос
Тема 5. Поток работ для получения требований к системе.	ПК-11	Опрос
Тема 6. Поток работ по анализу системы.	ПК-11	Опрос
Тема 7. Поток работ по проектированию системы.	ПК-11	Опрос
Тема 8. Поток работ по реализации системы.	ПК-11	Опрос
Тема 9. Поток работ по тестированию системы.	ПК-11	Опрос

### 8.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности процессе текущего контроля

В качестве оценочных средств текущего контроля успеваемости предусмотрены:

опрос

Примерные практические контрольные задания для текущего контроля успеваемости.

1. Основные понятия интеграции данных
2. Классификация систем интеграции баз данных
3. Критерии сравнения систем интеграции на примере конкретной системы
4. Поглощение запросов
5. Интерпретация GAV-взглядов
6. Интерпретация LAV-взглядов
7. Каноническая информационная модель
8. Исчисление спецификаций
9. Предметные посредники

10. Запросы к схеме посредника
11. Формулировка задач для решения в среде посредников
12. Унификация моделей данных
13. Преобразование базы данных модели OWL в объектное представление
14. Преобразование графовой базы данных в объектное представление
15. Мультидиалектные логические языки на правилах
16. Примеры виртуальной интеграции в Federation Server

### 8.3. Перечень вопросов и заданий для промежуточной аттестации по дисциплине

#### Вопросы для промежуточного контроля (экзамен)

1. Процесс разработки, ориентированный на управление сценариями использования системы.
2. Для чего требуются варианты использования системы.
3. Получение вариантов использования системы.
4. Этапы анализа, проектирования и реализации вариантов использования системы.
5. Тестирование вариантов использования системы.
6. Процесс ориентированный на разработку архитектуры системы.
7. Понятие архитектуры системы.
8. Необходимость архитектуры системы.
9. Связь вариантов использования системы и ее архитектуры.
10. Этапы в разработке архитектуры системы.
11. Описание архитектуры системы.
12. Итеративность и инкрементальность процесса.
13. Почему необходима итеративная и инкрементальная разработка.
14. Управление рисками при итеративном подходе в разработке системы.
15. Типовая итерация.
16. Инкремент как результат итерации.
17. Итерации в жизненном цикле системы.
18. Развитие моделей в результате итераций.
19. Поток работ для получения требований к системе как сценариев использования системы.
20. Артефакты: Модель сценариев использования; Актер; Сценарий использования; Описание архитектуры; Глоссарий; Прототип интерфейса пользователя.
21. Участники: Системный аналитик; Спецификатор сценариев использования; Проектировщик интерфейса пользователя; Архитектор
22. Деятельности: Поиск актеров и сценариев использования; Определение приоритетов для сценариев использования; Детализация сценариев использования; создание прототипа интерфейса пользователя; Структурирование модели сценариев использования;
23. Поток работ по анализу системы.
24. Роль фазы анализа в жизненном цикле программного обеспечения.
25. Артефакты: Анализ модели; Анализ класса; Анализ реализации сценария использования; Анализ пакета; Описание архитектуры.
26. Участники: Архитектор; Разработчик сценариев использования; Разработчик компонента.
27. Деятельности: Архитектурный анализ; Анализ сценариев использования; Анализ класса; Анализ пакета.
28. Поток работ по проектированию системы.
29. Роль потока работ проектирования в жизненном цикле программного обеспечения.
30. Артефакты: Модель проектирования; Проект класса; Проект реализации сценария использования; Проект подсистемы; Проект интерфейса.

31. Описание архитектуры (вид модели распределения); Модель внедрения системы.
32. Участники: Архитектор; Разработчик сценариев использования.
33. Разработчик компонента.
34. Деятельности: Проектирование архитектуры; Проектирование сценария использования системы.
35. Проектирование класса.
36. Проектирование подсистемы.
37. Поток работ по реализации системы.
38. Роль потока работ по реализации в жизненном цикле программного обеспечения.
- 39.Arteфакты: Модель реализации; Компонента; Подсистема реализации; Интерфейс; Описание архитектуры; План интеграции для реализации.
40. Участники: Архитектор; Разработчик компонента; Системный интегратор.
41. Деятельности: Реализация архитектуры; Интеграция системы; Реализация подсистемы; Реализация класса; Выполнение тестирования для единицы компиляции.
42. Поток работ по тестированию системы.
43. Роль потока работ по тестированию в жизненном цикле программного обеспечения.
44. Arteфакты: Модель тестирования; Набор тестов; Процедура тестирования; Компонента тестирования; План тестирования; Дефекты; Оценка теста.
45. Участники: Проектировщик теста; Разработчик компонента; Тестировщик интеграции; Тестировщик системы.
46. Деятельности: Планирования теста; Проектирование теста; Реализация теста; Выполнение тестирования интеграции; Выполнение тестирования системы; Оценка теста

#### 8.4. Планируемые уровни сформированности компетенций обучающихся и критерии оценивания

Уровни	Содержательное описание уровня	Основные признаки выделения уровня (этапы формирования компетенции, критерии оценки сформированности)	Пятибалльная шкала (академическая) оценка	Двухбалльная шкала, зачет	БРС, % освоения (рейтинговая оценка)
Повышенный	Творческая деятельность	<i>Включает нижестоящий уровень.</i> Умение самостоятельно принимать решение, решать проблему/задачу теоретического и прикладного характера на основе изученных методов, приемов, технологий	отлично	зачтено	86-100
Базовый	Применение знаний и умений в более широких контекстах учебной и профессиональной	<i>Включает нижестоящий уровень.</i> Способность собирать, систематизировать, анализировать и грамотно использовать информацию из самостоятельно	хорошо		71-85

	ьной деятельности, нежели по образцу с большей степени самостоятельности и инициативы	найденных теоретических источников и иллюстрировать ими теоретические положения или обосновывать практику применения			
Удовлетворительный (достаточный)	Репродуктивная деятельность	Изложение в пределах задач курса теоретически и практически контролируемого материала	удовлетворительно		55-70
Недостаточный	Отсутствие признаков удовлетворительного уровня		неудовлетворительно	не зачтено	Менее 55

### 9. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.

#### Основная литература

1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2023. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1895679> (дата обращения: 19.11.2023). – Режим доступа: по подписке.
2. Морозова, Ю. В. Объектно-ориентированный анализ и программирование : учебное пособие / Ю. В. Морозова. - Томск : Эль-Контент, 2018. - 140 с. - ISBN 978-5-4332-0269-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1845889> (дата обращения: 19.11.2023). – Режим доступа: по подписке.

#### Дополнительная литература

1. James Rumbaugh, Ivar Jacobson, Grady Booch. The Unified Modeling Language Reference Manual, 2nd Edition ISBN-13: 978-0321718952
2. Д. Рамбо. Д.Блаха. UML 2.0. Объектно-ориентированное моделирование и разработка. 2-е изд. 2006 год. ISBN: 5-469-00814-2544
3. G.Booch, R.A.Maksimchuk, M.W.Engle, B.J.Young, J.Conallen, K.A.Houston. Object-Oriented Analysis and Design with Applications 3-ed. 2007. Addison-Wesley ISBN 0-201-89551-X
4. Джим Арлоу, Айла Нейштадт, UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. Москва. 2007. Символ-Плюс, ISBN 978-5-93286-094-6
5. Гари Поллис, Лиз Огастин, Крис Лоу, Джас Мадхар. Разработка программных проектов. На основе Rational Unified Process (RUP) Москва. 2011. Бином-Пресс ISBN 978-5-9518-0449-5.
6. Ivar Jacobson, Grady Booch, James Rumbaugh. The Unified Software Development Process. Addison Wesley. 1999.
7. Скотт Амблерю Гибкие технологии: экстремальное программирование и унифицированный процесс разработки. Питер, 2005 г

## **10. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля).**

- НЭБ Национальная электронная библиотека, диссертации и прочие издания
- eLIBRARY.RU Научная электронная библиотека, книги, статьи, тезисы докладов конференций
- Гребенников Электронная библиотека ИД журналы
- ЭБС Консультант студента
- ПРОСПЕКТ ЭБС
- ЭБС ZNANIUM.COM
- ЭБС IBOOKS.RU
- РГБ Информационное обслуживание по МБА
- БЕН РАН
- Электронно-библиотечная система (ЭБС) Кантиана (<https://elib.kantiana.ru/>)

## **11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.**

Программное обеспечение обучения включает в себя:

- система электронного образовательного контента БФУ им. И. Канта – [www.lms.kantiana.ru](http://www.lms.kantiana.ru), обеспечивающую разработку и комплексное использование электронных образовательных ресурсов;
- серверное программное обеспечение, необходимое для функционирования сервера и связи с системой электронного обучения через Интернет;
- корпоративная платформа;
- установленное на рабочих местах обучающихся ПО: Microsoft Windows 10, Microsoft Office Standart 2010, антивирусное программное обеспечение Kaspersky Endpoint Security.
- GNU C++;
- Oracle Java;
- Deductor.
- среда разработки (JVE), компилятор (JVK) и виртуальная машина для исполнения кода (JVM) фирмы Oracle.
- Python 2.7.15 (Anaconda2 5.2.0 64-bit)
- Python 3.6.5 (Anaconda3 5.2.0 64-bit)

## **12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.**

Для проведения занятий лекционного типа, практических и семинарских занятий используются специальные помещения (учебные аудитории), оборудованные техническими средствами обучения – мультимедийной проекционной техникой. Для проведения занятий лекционного типа используются наборы демонстрационного оборудования.

Для проведения лабораторных работ, (практических занятий – при необходимости) используются специальные помещения (учебные аудитории), оснащенные специализированным лабораторным оборудованием: персональными компьютерами с возможностью выхода в интернет и с установленным программным обеспечением, заявленным в п. 11.

Для проведения групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используются специальные помещения (учебные аудитории), оборудованные специализированной мебелью (для обучающихся), меловой / маркерной доской.

Для организации самостоятельной работы обучающимся предоставляются помещения, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Для обучения инвалидов и лиц с ограниченными возможностями здоровья университетом могут быть представлены специализированные средства обучения, в том числе технические средства коллективного и индивидуального пользования.