



Федеральное государственное бюджетное образовательное учреждение
высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
АРХИТЕКТУРНО-СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ

Кафедра Информационных систем и технологий

УТВЕРЖДАЮ
Начальник учебно-методического управления

«29» июня 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Кросс-платформенное программирование

направление подготовки/специальность 09.03.03 Прикладная информатика

направленность (профиль)/специализация образовательной программы Прикладная информатика

Форма обучения очная

Санкт-Петербург, 2023

1. Цели и задачи освоения дисциплины (модуля)

Целью освоения дисциплины «Кроссплатформенное программирование» является формирование компетенций обучающегося в области разработки переносимых приложений.

Задачами освоения дисциплины являются изучение различных языков и инструментов разработки КППО, основных техник проектирования, отладки и тестирования КППО.

2. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, обеспечивающие достижение планируемых результатов освоения ОПОП
ПК-1 Способен проектировать и разрабатывать программное обеспечение	ПК-1.2 Осуществляет выбор типового решения и/или шаблона проектирования программного обеспечения, структуры данных, базы данных или программных интерфейсов в соответствии с поставленной задачей	знает основные парадигмы кроссплатформенного и объектно-ориентированного программирования; основные концептуальные положения объектно-ориентированного кроссплатформенного направления программирования, методов, способов и средств разработки переносимых программ; особенности программирования для различных операционных систем; основные аспекты концепции кроссплатформенного программирования умеет применять концепции объектно-ориентированного кроссплатформенного программирования, методы и способы разработки программ на языке Java; составлять алгоритмы и реализовывать программы с использованием технологий разработки кроссплатформенных программ; разрабатывать простейшие программы для работы в различных операционных системах; создавать кроссплатформенные программы на уровне выполнения. владеет навыкам и применения программных пакетов визуального программирования для создания прикладных кроссплатформенных программ.

3. Указание места дисциплины (модуля) в структуре образовательной программы

Данная дисциплина (модуль) включена в Блок «Дисциплины, модули» Б1.В.08 основной профессиональной образовательной программы 09.03.03 Прикладная информатика и относится к части, формируемой участниками образовательных отношений учебного плана.

№ п/п	Предшествующие дисциплины	Код и наименование индикатора достижения компетенции
1	Объектно-ориентированное программирование	ОПК-4.1, ОПК-4.2, ПК-1.1, ПК-1.3

2	Операционные системы и сети	ОПК-5.1, ОПК-5.2
---	-----------------------------	------------------

Объектно-ориентированное программирование

знать:

- основные парадигмы объектно-ориентированного программирования;
- основные структуры данных;

уметь:

- разрабатывать программное обеспечение в объектно-ориентированном стиле;
- создавать свои собственные классы;

Операционные системы и сети

знать:

- сетевые операционные системы;
- сетевые протоколы и сетевые сервисы;
- назначения MAC-адреса, IP- адреса, маски сети;

уметь:

- настраивать параметры сети и подключение к Интернет;

владеть:

- навыками установки программного обеспечения.

№ п/п	Последующие дисциплины	Код и наименование индикатора достижения компетенции
1	Выполнение и защита выпускной квалификационной работы	УК-1.1, УК-1.2, УК-1.3, УК-1.4, УК-1.5, УК-1.6, УК-2.1, УК-2.2, УК-2.3, УК-2.4, УК-3.1, УК-3.2, УК-3.3, УК-3.4, УК-3.5, УК-3.6, УК-4.1, УК-4.2, УК-4.3, УК-4.4, УК-6.1, УК-6.2, УК-6.3, УК-6.4, УК-6.5, УК-7.1, УК-7.2, УК-7.3, УК-7.4, УК-8.1, УК-8.2, УК-8.3, УК-8.4, УК-9.1, УК-9.2, УК-9.3, УК-9.4, УК-9.5, УК-10.1, УК-10.2, УК-10.3, ОПК-1.1, ОПК-1.2, ОПК-1.3, ОПК-1.4, ОПК-1.5, ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-2.4, ОПК-2.5, ОПК-3.1, ОПК-3.2, ОПК-4.1, ОПК-4.2, ОПК-5.1, ОПК-5.2, ОПК-6.1, ОПК-6.2, ОПК-7.1, ОПК-7.2, ОПК-8.1, ОПК-8.2, ОПК-9.1, ОПК-9.2, ПК-1.1, ПК-1.2, ПК-1.3, ПК- 1.4, ПК-2.1, ПК-2.2, ПК-3.1, ПК- 3.2, ПК-4.1, ПК-4.2, ПК-4.3, ПК- 4.4, ПК(Ц)-1.1, ПК(Ц)-1.2, ПК(Ц)- 1.3, ПК(Ц)-1.4, ПК(Ц)-1.5, УК-5.1, УК-5.2, УК-5.3, УК-5.4, УК-5.5, УК-5.6

4. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Вид учебной работы	Всего часов	Из них часы на практическую подготовку	Семестр
			7
Контактная работа	32		32
Практические занятия (Пр)	32	0	32

Иная контактная работа, в том числе:	1,25		1,25
консультации по курсовой работе (проекту), контрольным работам (РГР)	1		1
контактная работа на аттестацию (сдача зачета, зачета с оценкой; защита курсовой работы (проекта); сдача контрольных работ (РГР))	0,25		0,25
контактная работа на аттестацию в сессию (консультация перед экзаменом и сдача экзамена)			
Часы на контроль	4		4
Самостоятельная работа (СР)	70,75		70,75
Общая трудоемкость дисциплины (модуля)			
часы:	108		108
зачетные единицы:	3		3

5. Содержание дисциплины (модуля), структурированное по разделам (темам) с указанием отведенного на них количества академических часов и видов учебных занятий

5.1. Тематический план дисциплины (модуля)

№	Разделы дисциплины	Семестр	Контактная работа (по учебным занятиям), час.						СР	Всего, час.	Код индикатора достижения компетенции
			лекции		ПЗ		ЛР				
			всего	из них на практическую подготовку	всего	из них на практическую подготовку	всего	из них на практическую подготовку			
1.	1 раздел. Введение в кроссплатформенное программирование. Основные конструкции языка.										
1.1.	Введение в кроссплатформенное программирование	7			4				4	8	ПК-1.2
1.2.	Основные конструкции языка	7			6				6	12	ПК-1.2
1.3.	Объектно-ориентированный подход	7			6				6	12	ПК-1.2
2.	2 раздел. Расширенные возможности разработки кроссплатформенных приложений.										
2.1.	Организация сетевого взаимодействия	7			4				4	8	ПК-1.2
2.2.	Организация взаимодействия с помощью графического интерфейса	7			6				6	12	ПК-1.2
2.3.	Серверные приложения	7			6				6	12	ПК-1.2
2.4.	Курсовое проектирование	7						38,75		38,75	ПК-1.2
3.	3 раздел. Иная контактная работа										
3.1.	Иная контактная работа	7								1,25	ПК-1.2

4.	4 раздел. Контроль									
4.1.	Зачет	7							4	ПК-1.2

5.1. Практические занятия

№ разд	Наименование раздела и темы практических занятий	Наименование и содержание практических занятий
1	Введение в кроссплатформенное программирование	Настройка среды выполнения. Отладка программ. Кроссплатформенное ПО. Платформа. Основные подходы к разработке кроссплатформенных приложений. Кроссплатформенные языки программирования. Стандартизированные библиотеки среды выполнения. Кроссплатформенные среды исполнения. Кроссплатформенный пользовательский интерфейс.
2	Основные конструкции языка	Разработка программы с использованием основных конструкций языка Типы данных. Условные конструкции. Циклы. Ссылочные типы данных. Строки. Массивы. Контейнеры. Обработка исключений. Потоки ввода-вывода. Файловый поток. CLI. Разбор аргументов командной строки.
3	Объектно-ориентированный подход	Разработка программы на основе объектно-ориентированного подхода Классы. Наследование классов. Интерфейсы. Генерация исключений в классах. Модели данных. Организация взаимодействия с базой данных.
4	Организация сетевого взаимодействия	Разработка сетевых программ Организация сетевого взаимодействия. Клиент-серверные приложения. Многопоточность. Асинхронность.
5	Организация взаимодействия с помощью графического интерфейса	Разработка программ с графическим интерфейсом пользователя Принципы построения графического интерфейса. Компоненты и контейнеры. Графические примитивы. Обработка событий.
6	Серверные приложения	Разработка серверных приложений. Приложения, выполняющиеся на стороне сервера. Работа по протоколу Http.

5.2. Самостоятельная работа обучающихся

№ разд	Наименование раздела дисциплины и темы	Содержание самостоятельной работы
1	Введение в кроссплатформенное программирование	Инструментальные средства разработки кроссплатформенного ПО. Изучение теоретического материала, подготовка к практическим занятиям.
2	Основные конструкции языка	Применение основных конструкций языка. Изучение теоретического материала, подготовка к практическим занятиям.
3	Объектно-ориентированный	Абстрактные классы. Шаблоны классов. Изучение теоретического материала, подготовка к практическим

	подход	занятиям.
4	Организация сетевого взаимодействия	Организация сетевого взаимодействия по протоколам TCP, UDP. Изучение теоретического материала, подготовка к практическим занятиям.
5	Организация взаимодействия с помощью графического интерфейса	Проектирование GUI. Библиотеки и фреймворки. Изучение теоретического материала, подготовка к практическим занятиям.
6	Серверные приложения	Сервлеты. Серверные страницы. Изучение теоретического материала, подготовка к практическим занятиям.
7	Курсовое проектирование	Курсовая работа Постановка задачи. Спецификация требований. Программная реализация. Описание программы. Оформление пояснительной записки.

6. Методические материалы для самостоятельной работы обучающихся по дисциплине (модулю)

Программой дисциплины предусмотрено проведение практических занятий, на которых дается основной систематизированный материал и предполагающих закрепление изученного материала и формирование у обучающихся необходимых знаний, умений и навыков. Кроме того, важнейшим этапом изучения дисциплины является самостоятельная работа обучающихся с использованием всех средств и возможностей современных образовательных технологий.

В объем самостоятельной работы по дисциплине включается следующее:

- изучение теоретических вопросов по всем темам дисциплины;
- подготовка к практическим занятиям;
- подготовка к текущему контролю успеваемости студентов;
- подготовка к зачету.

Залогом успешного освоения дисциплины является обязательное посещение практических занятий, так как пропуск одного (тем более, нескольких) занятий может осложнить освоение разделов курса.

Приступая к изучению дисциплины, необходимо в первую очередь ознакомиться с содержанием РПД.

При подготовке к практическим занятиям и в рамках самостоятельной работы по изучению дисциплины обучающимся необходимо:

- ознакомиться с соответствующей темой занятия;
- осмыслить круг изучаемых вопросов и логику их рассмотрения;
- изучить рекомендуемую рабочей программой литературу по данной теме;
- при самостоятельном изучении теоретических вопросов сделать конспект, используя рекомендованные в РПД источники;
- выполнить практические задания в рамках изучаемой темы;
- ответить на контрольные вопросы по теме, используя материалы ФОС;
- подготовиться к проверочной работе, предусмотренной в контрольных точках;
- подготовиться к промежуточной аттестации.

Итогом изучения дисциплины является зачет. Зачет проводится по расписанию. Форма проведения занятия может быть устная, письменная и в электронном виде. Студенты, не прошедшие аттестацию, должны ликвидировать задолженность в установленном порядке.

7. Оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине (модулю)

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

№ п/п	Контролируемые разделы дисциплины (модуля)	Код и наименование индикатора контролируемой компетенции	Вид оценочного средства
1	Введение в кроссплатформенное программирование	ПК-1.2	Устный опрос, практические задания.
2	Основные конструкции языка	ПК-1.2	Устный опрос, практические задания.
3	Объектно-ориентированный подход	ПК-1.2	Устный опрос, практические задания.
4	Организация сетевого взаимодействия	ПК-1.2	Устный опрос, практические задания.
5	Организация взаимодействия с помощью графического интерфейса	ПК-1.2	Устный опрос, практические задания.
6	Серверные приложения	ПК-1.2	Устный опрос, практические задания.
7	Курсовое проектирование	ПК-1.2	Защита курсовой работы
8	Иная контактная работа	ПК-1.2	Устный опрос

9	Зачет	ПК-1.2	Промежуточная аттестация обучающихся
---	-------	--------	--------------------------------------

7.2. Типовые контрольные задания или иные материалы текущего контроля успеваемости, необходимые для оценки знаний, умений и навыков и (или) опыта профессиональной деятельности, характеризующих этапы формирования компетенций в процессе освоения дисциплины

для проверки сформированности индикатора достижения компетенции ПК-1.2

Теоретические вопросы текущего контроля:

1. Дайте определение понятия «портирование».
2. Дайте определение понятия «портируемость».
3. Назовите основные варианты организации разработки ПО при смене платформы.
4. Перечислите недостатки варианта организации разработки ПО при переходе на целевую ОС.
5. Назовите особенности организации разработки ПО при использовании привычной рабочей среды.
6. Назовите платформенно-зависимые элементы, используемые при разработке программного продукта. Приведите примеры.
7. Чем отличается компилятор от транслятора?
8. Дайте определение понятия «условная компиляция».
9. Назовите особенности использования эмулятора.
10. Перечислите кроссплатформенные языки программирования.
11. Перечислите кроссплатформенные среды исполнения.
12. Перечислите варианты кроссплатформенного интерфейса.
13. Дайте определение «Виртуальная машина».
14. Дайте определение «Виртуальная Машина Java».
15. Дайте определение «Just-In-Time компилятора».
16. Дайте определение «Ahead-of-time компилятора».
17. Назначение файла с расширением .java.
18. Назначение файла с расширением .class.
19. Назначение файла с расширением .jar.
20. Дайте определение «Байт-код Java».
21. Раскройте динамическую объектно-ориентированную модель представления байт-кода Java.
22. Раскройте динамическое связывание внешних методов при выполнении байт-кода Java.
23. Как и каким образом виртуальная машина Java осуществляет контроль за всеми объектами?
24. Каким образом происходит организация выполнения многопоточных приложений в виртуальной машине Java?
25. Раскройте этапы интерпретации java-программы.
26. Каким образом можно повысить скорость интерпретации Java-программ?
27. Дайте определение алфавита языка Java.
28. Какие символы допускается использовать в качестве пробелов?
29. Назовите виды комментариев, приведите примеры.
30. Дайте определение «Зарезервированные ключевые слова».
31. Правила идентификаторов.
32. Напишите примитивные типы данных для работы с целыми числами. Приведите примеры литералов.
33. Напишите примитивные типы данных для работы с вещественными числами. Приведите примеры литералов.
34. Напишите тип данных для работы с символами. Приведите примеры литералов.
35. Напишите тип данных для работы с логическими переменными. Особенности использования в языке Java.
36. Что такое класс-оболочка для примитивных типов данных?
37. Напишите правила работы со строками. Приведите примеры литералов.
38. Правила описания переменных и констант. Концепция границ.

39. Перечислите и поясните правила использования символов разделителей.
40. Чем отличаются префиксные и суффиксные формы операций? Приведите примеры.
41. Перечислите операции отношения и логические операции.
42. Правила приоритетов операций.
43. Создание одномерных массивов. Примеры.
44. Создание многомерных массивов. Примеры.
45. Что такое класс?
46. Зачем нужен конструктор?
47. Что делает оператор new?
48. В чем отличие спецификаторов доступа public и private?
49. Что такое класс-оболочка?
50. Из каких элементов состоит описание класса?
51. Что такое конструктор?
52. Какие спецификаторы доступа Вы знаете?
53. Для чего используются перегруженные конструкторы?
54. Какой доступ к элементам класса при отсутствии спецификатора?
55. Что такое интерфейс класса?
56. Что такое инкапсуляция?
57. Приведите пример полиморфизма.
58. Что такое перегрузка функций?
59. Какие виды отношений между классами вы знаете?
60. Приведите пример агрегирования.
61. В чем отличие агрегирования от композиции?
62. Раскройте понятие отношения обобщения между классами.
63. Приведите пример отношения зависимости между классами.
64. Что такое интерфейс класса?
65. Что такое методы класса?
66. Для чего используется ключевое слово this?
67. Зачем кроме примитивных типов в язык Java введены еще соответствующие классы-оболочки?
68. Можно ли использовать объекты числовых классов-оболочек в арифметических выражениях?
69. Какое наибольшее целое значение можно занести в объект класса BigInteger?
70. Какое наибольшее вещественное значение можно занести в объект класса BigDecimal?
71. Можно ли использовать в одном выражении значения примитивных типов и распакованные значения числовых классов-оболочек?
72. Для чего в язык Java введены настраиваемые типы?
73. Можно ли создавать настраиваемые интерфейсы или настраиваемыми могут быть только классы?
74. Должны ли методы настраиваемого класса быть настраиваемыми?
75. Можно ли создавать настраиваемые методы в обычных, не настраиваемых классах?
76. Зачем в язык Java введено несколько классов, обрабатывающих строки символов?
77. Какова разница между классами String и StringBuilder?
78. Какова разница между классами StringBuffer и StringBuilder?
79. Что лучше использовать для сцепления строк: операцию сцепления или метод append() класса StringBuilder?
80. Что лучше использовать для разбора строки: метод split() или класс StringTokenizer?
81. Что называется коллекцией?
82. В чем отличие вектора от массива?
83. Что дает задание конкретного класса в шаблоне при определении коллекции?
84. В чем различие интерфейсов List и Set?
85. В чем различие интерфейсов List и Queue?
86. Что дополняет интерфейс Deque к интерфейсу Queue?
87. Зачем в Java введены интерфейсы NavigableSet и NavigableMap?
88. Что такое стек?

89. Что такое ассоциативный массив?
90. Что такое линейный список?
91. Что такое двунаправленный список?
92. Какие способы обхода коллекции вы знаете?
93. Каким классом-коллекцией лучше всего организовать очередь?
94. Когда удобнее использовать класс Vector, а когда — ArrayList?
95. Можно ли совсем отказаться от объекта Iterator в пользу цикла "for-each"?
96. Какие классы-коллекции реализуют структуру данных "дерево"?
97. Что такое графический интерфейс пользователя?
98. Что такое графическая библиотека классов?
99. Что называется графическим компонентом?
100. Назовите известные вам графические компоненты.
101. Что такое контейнер в графическом интерфейсе?
102. Будет ли основное окно приложения контейнером?
103. Можно ли использовать библиотеку Swing без библиотеки AWT?
104. Какая разница между компонентами AWT и компонентами Swing?
105. Что такое цвет в библиотеке AWT?
106. Что такое шрифт в библиотеке AWT?
107. Что такое графический контекст?
108. Как нарисовать треугольник?
109. Как нарисовать окружность?
110. Как преобразовать чертеж: повернуть его, уменьшить или увеличить?
111. Можно ли писать текст сверху вниз или справа налево?
112. Почему класс Container наследует от класса Component, а не наоборот?
113. Каковы общие свойства всех компонентов?
114. Почему класс Component сделан абстрактным?
115. Почему надпись в контейнере — это целый компонент класса Label, а не просто строка символов?
116. Почему для группы радиокнопок не создан отдельный компонент?
117. В чем разница между текстовыми компонентами TextField и TextArea?
118. Почему мы всегда используем окно типа Frame, а не типа Window?
119. Чем отличается создание "тяжелого" компонента от создания "легкого" компонента?
120. Чем отличаются компоненты Swing от компонентов AWT?
121. Какая конструктивная схема использована в компонентах Swing?
122. В каких случаях приходится изменять Модель компонента?
123. В каких случаях приходится изменять Вид компонента?
124. Какими компонентами Swing можно создать кнопку с двумя состояниями?
125. Какие компоненты Swing создают радиокнопки?
126. Как в Swing показать каталоги и имена файлов в виде дерева
127. Как используется модель MVC в текстовых компонентах?
128. В чем отличие текстовых компонентов Swing от аналогичных компонентов AWT?
129. Можно ли в текстовых компонентах Swing менять шрифт?
130. Можно ли в текстовых компонентах Swing использовать разные шрифты в одной строке?
131. В каких случаях удобно использовать готовые текстовые редакторы Swing?
132. Можно ли средствами Swing написать свой текстовый редактор Swing?
133. Можно ли в текстовых редакторах Swing использовать разные шрифты в одном документе?
134. Как конструктивная схема MVC использована для создания классов таблиц?
135. Можно ли хранить в таблицах класса JTable образцы цвета?
136. Можно ли хранить в ячейках таблицы раскрывающиеся списки?
137. Можно ли сделать таблицу с ячейками-кнопками?
138. Можно ли сделать отдельные ячейки таблицы не редактируемыми?
139. Можно ли делать вычисления в таблице класса JTable, как это делается в электронных таблицах?
140. Можно ли сортировать строки таблиц класса JTable?

141. Можно ли сделать электронную таблицу средствами класса JTable?
142. Что такое менеджер размещения?
143. Почему менеджеры размещения удобнее абсолютной расстановки компонентов?
144. В каких контейнерах можно установить менеджер размещения?
145. В каких компонентах можно установить менеджер размещения?
146. Почему менеджер размещения BorderLayout столь популярен?
147. Какой менеджер размещения установлен по умолчанию в окне класса JFrame?
148. Какой менеджер размещения установлен по умолчанию в классе JPanel?
149. Какой менеджер размещения установлен по умолчанию в классе JScrollPane?
150. Какой менеджер размещения установлен по умолчанию в классе JWindow?
151. Какой менеджер размещения установлен по умолчанию в классе JDialog?
152. Можно ли написать свой собственный менеджер размещения?
153. Какая модель обработки событий выбрана в AWT?
154. Что нового добавила в обработку событий библиотека Swing?
155. Какие компоненты отслеживают события мыши?
156. Какие компоненты отслеживают события клавиатуры?
157. Может ли одно и то же событие возникнуть сразу в нескольких компонентах?
158. Может ли одно действие вызвать сразу несколько событий?
159. Можно ли сделать обработку нескольких событий одним методом?
160. Можно ли обработать одно событие сразу несколькими методами?
161. Как в AWT осуществляется диспетчеризация событий?
162. Что такое апплет?
163. Чем апплет отличается от приложения?
164. Можно ли написать программу, которую смогут выполнять и интерпретатор, и браузер, т. е. одновременно апплет и приложение?
165. Можно ли написать конструктор в апплете?
166. Будет ли выполняться конструктор в апплете?
167. Как передать параметры в апплет?
168. Может ли апплет читать файлы на машине браузера?
169. Может ли апплет читать файлы на машине, с которой он загружен?
170. Может ли апплет передавать данные по сети?
171. Какой класс в Java обеспечивает работу со строками?
172. Как создать строку с помощью конструктора класса String?
173. Какие способы инициализации строк Вы знаете?
174. Каковы основные операции со строками?
175. Как выполнить сравнение строк?
176. Почему в объектно-ориентированных языках принята модель обработки исключительных ситуаций, отличная от модели, принятой в процедурных языках?
177. Можно ли вкладывать друг в друга блоки обработки исключений?
178. Можно ли в блоках обработки исключений применить оператор return?
179. Можно ли в блоках обработки исключений снова вызвать исключение?
180. Можно ли в одном блоке обработки исключений написать несколько блоков catch(){}?
181. Можно ли в одном блоке обработки исключений написать несколько блоков finally{}?
182. В каких случаях в заголовке метода, выбрасывающего исключения, может отсутствовать часть throws?
183. С какой целью создана разветвленная иерархия классов исключений?
184. Можно ли создать свои собственные классы-исключения?
185. Что такое процесс и подпроцесс в современных операционных системах?
186. Почему языки программирования, как правило, не содержат средств управления подпроцессами?
187. Зачем в язык Java введены средства создания и управления подпроцессами?
188. Какими способами можно создать и запустить подпроцесс?
189. Когда подпроцесс заканчивает свою работу?
190. Как можно остановить подпроцесс?

191. Что такое "монитор" в теории операционных систем?
192. Каким образом монитор реализуется в Java?
193. Что называется потоком (stream) данных?
194. Какие потоки ввода/вывода создаются исполняемой системой Java для каждой запущенной программы?
195. Как можно преобразовать поток ввода/вывода?
196. Как изменить кодировку символов в потоке?
197. Что такое буферизация ввода/вывода и для чего она нужна?
198. Что такое сериализация объекта?
199. Почему вопросы, относящиеся к печати, разобраны в этой главе, посвященной вводу/выводу?
200. Что такое сетевая архитектура "клиент-сервер"?
201. Чем отличается клиентская часть приложения от серверной части?
202. Может ли клиентская часть приложения взаимодействовать сразу с несколькими серверами?
203. Как установить связь через прокси-сервер?
204. Что такое JDBC?
205. Какая разница между HTTP-сервером и Web-сервером?
206. Что такое сервер приложений?
207. Может ли сервлет отправить клиенту не страницу HTML, а другой документ?
208. Может ли сервлет обрабатывать параллельно несколько запросов?
209. Могут ли сервлеты, установленные в один контейнер, обмениваться информацией?
210. Может ли сервлет установить сеанс связи с клиентом?
211. Можно ли записывать код Java на страницах JSP?
212. Можно ли включать в страницу JSP другие файлы?
213. Можно ли передавать управление из страницы JSP другим ресурсам?
214. Можно ли написать несколько классов Java, по-разному обрабатывающих один и тот же пользовательский тег?
215. Можно ли обработать пользовательский тег не классом Java, а страницей JSP?
216. Как подключить библиотеку пользовательских тегов к странице JSP?
217. Какой основной цели служат элементы XML?
218. Как выполняется синтаксический анализ документа XML?
219. В каких случаях удобнее проводить синтаксический анализ документа XML, основываясь на событиях, а в каких — построением дерева?
220. Каким образом можно преобразовать один документ XML в другой документ?

7.3. Система оценивания результатов обучения по дисциплине (модулю) при проведении текущего контроля успеваемости

<p>Оценка «отлично» (зачтено)</p>	<p>знания: - систематизированные, глубокие и полные знания по всем разделам дисциплины, а также по основным вопросам, выходящим за пределы учебной программы; - точное использование научной терминологии, систематически грамотное и логически правильное изложение ответа на вопросы; - полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой по дисциплине (модулю)</p> <p>умения: - умеет ориентироваться в теориях, концепциях и направлениях дисциплины и давать им критическую оценку, используя научные достижения других дисциплин</p> <p>навыки: - высокий уровень сформированности заявленных в рабочей программе компетенций; - владеет навыками самостоятельно и творчески решать сложные проблемы и нестандартные ситуации; - применяет теоретические знания для выбора методики выполнения заданий; - грамотно обосновывает ход решения задач; - безупречно владеет инструментарием учебной дисциплины, умение его эффективно использовать в постановке научных и практических задач; - творческая самостоятельная работа на практических/семинарских/лабораторных занятиях, активно участвует в групповых обсуждениях, высокий уровень культуры исполнения заданий</p>
<p>Оценка «хорошо» (зачтено)</p>	<p>знания: - достаточно полные и систематизированные знания по дисциплине; - усвоение основной и дополнительной литературы, рекомендованной рабочей программой по дисциплине (модулю)</p> <p>умения: - умеет ориентироваться в основных теориях, концепциях и направлениях дисциплины и давать им критическую оценку; - использует научную терминологию, лингвистически и логически правильно излагает ответы на вопросы, умеет делать обоснованные выводы; - владеет инструментарием по дисциплине, умение его использовать в постановке и решении научных и профессиональных задач</p> <p>навыки: - самостоятельная работа на практических занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий; - средний уровень сформированности заявленных в рабочей программе компетенций; - без затруднений выбирает стандартную методику выполнения заданий; - обосновывает ход решения задач без затруднений</p>

<p>Оценка «удовлетворительно» (зачтено)</p>	<p>знания: - достаточный минимальный объем знаний по дисциплине; - усвоение основной литературы, рекомендованной рабочей программой; - использование научной терминологии, стилистическое и логическое изложение ответа на вопросы, умение делать выводы без существенных ошибок умения: - умеет ориентироваться в основных теориях, концепциях и направлениях по дисциплине и давать им оценку; - владеет инструментарием учебной дисциплины, умение его использовать в решении типовых задач; - умеет под руководством преподавателя решать стандартные задачи навыки: - работа под руководством преподавателя на практических занятиях, допустимый уровень культуры исполнения заданий; - достаточный минимальный уровень сформированности заявленных в рабочей программе компетенций; - испытывает затруднения при обосновании алгоритма выполнения заданий</p>
<p>Оценка «неудовлетворительно» (не зачтено)</p>	<p>знания: - фрагментарные знания по дисциплине; - отказ от ответа (выполнения письменной работы); - знание отдельных источников, рекомендованных рабочей программой по дисциплине; умения: - не умеет использовать научную терминологию; - наличие грубых ошибок навыки: - низкий уровень культуры исполнения заданий; - низкий уровень сформированности заявленных в рабочей программе компетенций; - отсутствие навыков самостоятельной работы; - не может обосновать алгоритм выполнения заданий</p>

7.4. Теоретические вопросы и практические задания для проведения промежуточной аттестации обучающихся, необходимые для оценки знаний, умений и навыков и (или) опыта профессиональной деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

7.4.1. Теоретические вопросы для проведения промежуточной аттестации обучающихся

1. Определение и примеры кроссплатформенности.
2. Уровни кроссплатформенности (аппаратный / программный, компиляции / выполнения).
3. Обзор языков кроссплатформенного программирования.
4. Языки реализующие кроссплатформенность на уровне компиляции: C, C+
5. Языки реализующие кроссплатформенность на уровне выполнения: Java, JavaScript, PHP, C#
6. Обзор сред разработки кроссплатформенного ПО.
7. Обзор языков реализация кроссплатформенности на уровне выполнения
8. Характеристика языка Java.
9. Организация клиент-серверных передач данных.
10. Организация серверной обработки клиентских запросов.

7.4.2. Практические задания для проведения промежуточной аттестации обучающихся

1. Разработать программу с графическим интерфейсом пользователя, позволяющую просмотреть содержимое базы данных.
2. Разработать программу с графическим интерфейсом пользователя, позволяющую просмотреть содержимое XML-файла.
3. Разработать программу с интерфейсом CLI, позволяющую просмотреть содержимое базы

данных.

4. Разработать программу обмен сообщениями на основе протокола TCP.

7.4.3. Примерные темы курсовой работы (проекта) (при наличии)

1. Разработка кроссплатформенного приложения.
2. Разработка кроссплатформенного приложения с графическим интерфейсом пользователя.
3. Разработка кроссплатформенного приложения с сетевым взаимодействием.

7.5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта профессиональной деятельности, характеризующие этапы формирования компетенций

Процедура проведения промежуточной аттестации и текущего контроля успеваемости регламентируется локальным нормативным актом, определяющим порядок организации и проведения текущего контроля успеваемости и промежуточной аттестации обучающихся.

Процедура оценивания формирования компетенций при проведении текущего контроля приведена в п. 7.2.

Типовые контрольные задания или иные материалы текущего контроля приведены в п. 7.3.

Промежуточная аттестация по дисциплине проводится в форме зачета, защиты курсовой работы.

7.6. Критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Критерии оценивания	Уровень освоения и оценка			
	Оценка «неудовлетворительно»	Оценка «удовлетворительно»	Оценка «хорошо»	Оценка «отлично»
	«не зачтено»	«зачтено»		
	Уровень освоения компетенции «недостаточный». Компетенции не сформированы. Знания отсутствуют, умения и навыки не сформированы	Уровень освоения компетенции «пороговый». Компетенции сформированы. Сформированы базовые структуры знаний. Умения фрагментарны и носят репродуктивный характер. Демонстрируется низкий уровень самостоятельности практического навыка.	Уровень освоения компетенции «продвинутый». Компетенции сформированы. Знания обширные, системные. Умения носят репродуктивный характер, применяются к решению типовых заданий. Демонстрируется достаточный уровень самостоятельности устойчивого практического навыка.	Уровень освоения компетенции «высокий». Компетенции сформированы. Знания аргументированные, всесторонние. Умения успешно применяются к решению как типовых, так и нестандартных творческих заданий. Демонстрируется высокий уровень самостоятельности, высокая адаптивность практического навыка

знания	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> -существенные пробелы в знаниях учебного материала; -допускаются принципиальные ошибки при ответе на основные вопросы билета, отсутствует знание и понимание основных понятий и категорий; -непонимание сущности дополнительных вопросов в рамках заданий билета. 	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> -знания теоретического материала; -неполные ответы на основные вопросы, ошибки в ответе, недостаточное понимание сущности излагаемых вопросов; -неуверенные и неточные ответы на дополнительные вопросы. 	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> -знание и понимание основных вопросов контролируемого объема программного материала; - знания теоретического материала -способность устанавливать и объяснять связь практики и теории, выявлять противоречия, проблемы и тенденции развития; -правильные и конкретные, без грубых ошибок, ответы на поставленные вопросы. 	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> -глубокие, всесторонние и аргументированные знания программного материала; -полное понимание сущности и взаимосвязи рассматриваемых процессов и явлений, точное знание основных понятий, в рамках обсуждаемых заданий; -способность устанавливать и объяснять связь практики и теории, логически последовательные, содержательные, конкретные и исчерпывающие ответы на все задания билета, а также дополнительные вопросы экзаменатора.
умения	<p>При выполнении практического задания билета обучающийся продемонстрировал недостаточный уровень умений. Практические задания не выполнены. Обучающийся не отвечает на вопросы билета при дополнительных наводящих вопросах преподавателя.</p>	<p>Обучающийся выполнил практическое задание билета с существенными неточностями. Допускаются ошибки в содержании ответа и решении практических заданий. При ответах на дополнительные вопросы было допущено много неточностей.</p>	<p>Обучающийся выполнил практическое задание билета с небольшими неточностями. Показал хорошие умения в рамках освоенного учебного материала. Предложенные практические задания решены с небольшими неточностями. Ответил на большинство дополнительных вопросов.</p>	<p>Обучающийся правильно выполнил практическое задание билета. Показал отличные умения в рамках освоенного учебного материала. Решает предложенные практические задания без ошибок. Ответил на все дополнительные вопросы.</p>

владение навыками	<p>Не может выбрать методику выполнения заданий. Допускает грубые ошибки при выполнении заданий, нарушающие логику решения задач. Делает некорректные выводы. Не может обосновать алгоритм выполнения заданий.</p>	<p>Испытывает затруднения по выбору методики выполнения заданий. Допускает ошибки при выполнении заданий, нарушения логики решения задач. Испытывает затруднения с формулированием корректных выводов. Испытывает затруднения при обосновании алгоритма выполнения заданий.</p>	<p>Без затруднений выбирает стандартную методику выполнения заданий. Допускает ошибки при выполнении заданий, не нарушающие логику решения задач. Делает корректные выводы по результатам решения задачи. Обосновывает ход решения задач без затруднений.</p>	<p>Применяет теоретические знания для выбора методики выполнения заданий. Не допускает ошибок при выполнении заданий. Самостоятельно анализирует результаты выполнения заданий. Грамотно обосновывает ход решения задач.</p>
-------------------	--	---	---	--

Оценка по дисциплине зависит от уровня сформированности компетенций, закрепленных за дисциплиной, и представляет собой среднее арифметическое от выставленных оценок по отдельным результатам обучения (знания, умения, владение навыками).

Оценка «отлично»/«зачтено» выставляется, если среднее арифметическое находится в интервале от 4,5 до 5,0.

Оценка «хорошо»/«зачтено» выставляется, если среднее арифметическое находится в интервале от 3,5 до 4,4.

Оценка «удовлетворительно»/«зачтено» выставляется, если среднее арифметическое находится в интервале от 2,5 до 3,4.

Оценка «неудовлетворительно»/«не зачтено» выставляется, если среднее арифметическое находится в интервале от 0 до 2,4.

8. Учебно-методическое и материально-техническое обеспечение дисциплины (модуля)

8.1. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

№ п/п	Автор, название, место издания, издательство, год издания учебной и учебно-методической литературы	Количество экземпляров/электронный адрес ЭБС
<u>Основная литература</u>		
1	Леднева С. Ю., Ефимушкина Н. В., Цыганов А. А., Программирование на Java. Ч.1, Самара: Самарский государственный технический университет, ЭБС АСВ, 2015	http://www.iprbookshop.ru/90875.html
2	Лобзенко П. В., Щербань И. В., Проектирование клиент-серверных приложений, Ростов-на-Дону: Северо-Кавказский филиал Московского технического университета связи и информатики, 2018	http://www.iprbookshop.ru/89512.html
3	Гуськова О. И., Объектно ориентированное программирование в Java, Москва: Московский педагогический государственный университет, 2018	http://www.iprbookshop.ru/97750.html
4	Монахов В. В., Язык программирования Java и среда NetBeans, Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021	https://www.iprbookshop.ru/102078.html
<u>Дополнительная литература</u>		

1	Лобзенко П. В., Щербань И. В., Учебное пособие по дисциплине Разработка кроссплатформенных приложений С++, Ростов-на-Дону: Северо-Кавказский филиал Московского технического университета связи и информатики, 2018	http://www.iprbookshop.ru/89513.html
1	Кокорева Е. В., Технология разработки телекоммуникационных сервисов. Распределённые приложения, Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2019	https://www.iprbookshop.ru/102142.html

Обучающиеся из числа инвалидов и лиц с ОВЗ обеспечиваются печатными и (или) электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

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

Наименование ресурса сети «Интернет»	Электронный адрес ресурса
Степанов, Е.О. Кросс-платформенные и многозвенные технологии [Электронный ресурс] : учебное пособие / Е.О. Степанов. — Электрон. дан. — Москва, 2016. — 361 с.	https://intuit.ru/studies/courses/571/427/info
Центр справки Java	www.java.com/ru/download/help/index.html
Java Documentation	https://docs.oracle.com/en/java/

8.3. Перечень современных профессиональных баз данных и информационных справочных систем

Наименование	Электронный адрес ресурса
Электронно-библиотечная система издательства "IPRsmart"	http://www.iprbookshop.ru/
Электронно-библиотечная система издательства "ЮРАЙТ"	https://www.biblio-online.ru/
Электронно-библиотечная система издательства "Лань"	https://e.lanbook.com/
Электронная библиотека Ирбис 64	http://ntb.spbgasu.ru/irbis64r_plus/
Система дистанционного обучения СПбГАСУ Moodle	https://moodle.spbgasu.ru/

8.4. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного и свободно распространяемого программного обеспечения

Наименование	Способ распространения (лицензионное или свободно распространяемое)
Microsoft Windows 10 Pro	Договор № Д32009689201 от 18.12.2020г
Microsoft Office 2016	Договор № Д32009689201 от 18.12.2020г Программные продукты Майкрософт, договор № Д32009689201 от 18.12.2020 с АО "СофтЛайн Трейд": Windows 10, Project Professional 2016, Visio Professional 2016, Office 2016.

Microsoft Visual Studio 2017	Договор № Д32009689201 от 18.12.2020г Программные продукты Майкрософт, договор № Д32009689201 от 18.12.2020 с АО "СофтЛайн Трейд": Windows 10, Project Professional 2016, Visio Professional 2016, Office 2016.
Notepad++ версия 7.7.1	Свободно распространяемое

8.5. Материально-техническое обеспечение дисциплины

Сведения об оснащённости учебных аудиторий и помещений для самостоятельной работы

Наименование учебных аудиторий и помещений для самостоятельной работы	Оснащённость оборудованием и техническими средствами обучения
47. Компьютерный класс	Рабочие места с ПК (стол компьютерный, системный блок, монитор, клавиатура, мышь), стол рабочий, подключение к компьютерной сети СПбГАСУ, выход в Internet.
47. Учебные аудитории для проведения практических занятий, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Учебная аудитория для проведения практических занятий, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации – комплект мультимедийного оборудования (персональный компьютер, мультимедийный проектор, экран, аудиосистема), доска маркерная белая эмалевая, комплект учебной мебели, подключение к компьютерной сети СПбГАСУ. выход в Интернет.
47. Помещения для самостоятельной работы	Помещение для самостоятельной работы (читальный зал библиотеки, ауд. 217): ПК-23 шт., в т.ч. 1 шт.- ПК для лиц с ОВЗ (системный блок, монитор, клавиатура, мышь) с подключением к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду СПбГАСУ. ПО Microsoft Windows 10, Microsoft Office 2016

Для инвалидов и лиц с ОВЗ обеспечиваются специальные условия для получения образования в соответствии с требованиями нормативно-правовых документов.