

Кафедра № 34

УТВЕРЖДАЮ
 Руководитель направления
 проф. д.т.н. доц.
 (подпись, и.т.с., дата)

С.В. Белицкий
 (подпись, фамилия)
 «24» марта 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ


«Язык программирования»
 (наименование дисциплины)

Код направления подготовки/ специальности	10.05.03
Наименование направления подготовки/ специальности	Информационная безопасность автоматизированных систем
Наименование специальности	Обеспечение информационной безопасности распределенных информационных систем
Форма обучения	очная

Программу составил (а)
 доц. д.т.н. доц.
 (подпись, и.т.с., дата)  24.03.22 В.А. Мыльников
 (подпись, фамилия)

Программа одобрена на заседании кафедры № 34
 «24» марта 2022 г. протокол № 8

Заведующий кафедрой № 34
 д.т.н. доц.
 (подпись, и.т.с., дата)  24.03.22 С.В. Белицкий
 (подпись, фамилия)

Ответственный за ОП ВО 10.05.03(05)
 доц. д.т.н. доц.
 (подпись, и.т.с., дата)  24.03.22 В.А. Мыльников
 (подпись, фамилия)

Заместитель директора института №3 по методической работе
 ст. преп.
 (подпись, и.т.с., дата)  24.03.22 Н.В. Решетникова
 (подпись, фамилия)

Аннотация

Дисциплина «Языки программирования» входит в базовую часть образовательной программы подготовки обучающихся по специальности «10.05.03 «Информационная безопасность автоматизированных систем» направленность «Обеспечение информационной безопасности распределенных информационных систем». Дисциплина реализуется кафедрой №54.

Дисциплина нацелена на формирование у выпускника

общефессиональных компетенций:

ОПК-1 «способность анализировать физические явления и процессы, применять соответствующий математический аппарат для формализации и решения профессиональных задач»,

ОПК-3 «способность применять языки, системы и инструментальные средства программирования в профессиональной деятельности»,

ОПК-4 «способность понимать значение информации в развитии современного общества, применять достижения современных информационных технологий для поиска информации в компьютерных системах, сетях, библиотечных фондах».

Содержание дисциплины охватывает круг вопросов, связанных с приемами программирования на нескольких распространенных языках программирования; методами разработки средних и больших программ на данных языках; принципами организации структур данных в языках со ссылочной и размерной объектной моделью; основами объектно-ориентированного программирования; синтаксисом и семантикой нескольких распространенных языков программирования

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа студента, консультации.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 4 зачетных единицы, 144 часа.

Язык обучения по дисциплине «русский».

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Сформировать у учащихся уверенные знания и навыки программирования на нескольких современных языках программирования, дать их сравнительную характеристику, заложить способность легко разбираться в новых языках программирования, дать последовательное изложение объектно-ориентированного программирования, особенностей его использования на нескольких языках.

Основными задачами освоения дисциплины являются:

- Знакомство студентов с несколькими распространенными современными языками программирования, их сравнительный анализ.
- Сравнение организации ссылочной и размерной объектной модели в различных языках.
- Изучение стандартных библиотек языков C++, Java.
- Знакомство с функциональной парадигмой программирования.

1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП

В результате освоения дисциплины обучающийся должен обладать следующими компетенциями:

ОПК-1 «способность анализировать физические явления и процессы, применять соответствующий математический аппарат для формализации и решения профессиональных задач»:

знать – постановку исследовательских задач и выбору путей их решения;

уметь – осмысливать, систематизировать, прогнозировать информацию;

владеть навыками - пользоваться средствами к выбору путей решения поставленных задач;

иметь опыт деятельности – к логическому мышлению, обобщению, анализу, критическому осмыслению информации.

ОПК-3 «способность применять языки, системы и инструментальные средства программирования в профессиональной деятельности»:

знать - приемы программирования на нескольких распространенных языках программирования;

уметь - выполнять и анализировать постановку задачи;

владеть навыками - несколькими распространенными языками программирования;

иметь опыт деятельности - использовать подходящие структуры данных и библиотечные функции/классы;

ОПК-4 «способность понимать значение информации в развитии современного общества, применять достижения современных информационных технологий для поиска информации в компьютерных системах, сетях, библиотечных фондах»:

знать - методы разработки средних и больших программ на данных языках;

уметь - формулировать математическую модель задачи; оценивать сложность и эффективность алгоритма;

владеть навыками - навыками работы в распространенных интегрированных средах программирования;

иметь опыт деятельности - выполнять отладку и тестирование;

2. Место дисциплины в структуре ОП

Дисциплина базируется на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- Математический анализ
- Математическая логика и теория алгоритмов
- Учебная (ознакомительная) практика

- Теория вероятностей и математическая статистика
- Вычислительная математика
- Технологии и методы программирования
- Учебная практика
- Математические основы обработки информации
- Производственная (эксплуатационная) практика
- Моделирование систем
- Системное программное обеспечение
- Информатика
- Основы программирования
- Информационные технологии
- Теория информации
- Теория информационной безопасности

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и используются при изучении других дисциплин:

- Защита информации в сенсорных сетях
- Научно-исследовательская работа
- Производственная преддипломная практика
- Технология построения защищенных распределенных приложений
- Информационная безопасность распределенных информационных систем

3. Объем дисциплины в ЗЕ/академ. час

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 1

Таблица 1 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам
		№8
1	2	3
Общая трудоемкость дисциплины, ЗЕ/(час)	4/ 144	4/ 144
<i>Аудиторные занятия</i> , всего час., <i>В том числе</i>	68	68
лекции (Л), (час)	34	34
Практические/семинарские занятия (ПЗ), (час)		
лабораторные работы (ЛР), (час)	34	34
курсовой проект (работа) (КП, КР), (час)		
Экзамен, (час)	36	36
<i>Самостоятельная работа</i> , всего	40	40
Вид промежуточного контроля: зачет, дифф. зачет, экзамен (Зачет ,	Экз.	Экз.

Дифф. зач, Экз.)		
------------------	--	--

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий

Разделы и темы дисциплины и их трудоемкость приведены в таблице 2.

Таблица 2. – Разделы, темы дисциплины и их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ)	ЛР (час)	КП (час)	СРС (час)
Семестр 8					
Раздел 1. Язык Java	12		12		16
Раздел 2. Язык C++	12		12		12
Раздел 3. Элементы функционального программирования	10		10		12
Итого в семестре:	34		34		40
Итого:	17	0	34	0	57

4.2. Содержание разделов и тем лекционных занятий

Содержание разделов и тем лекционных занятий приведено в таблице 3.

Таблица 3 - Содержание разделов и тем лекционных занятий

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	Раздел 1. Язык Java Основы языка Java. Массивы, функции, классы, наследование, полиморфизм. Стандартные пакеты контейнеров. Работа с файлами. Обобщенное программирование
2	Раздел 2. Язык C++ Размерная объектная модель. Перегрузка операций. Реализация конструкторов, деструкторов, операции присваивания. Особенности реализации полиморфизма. Множественное наследование
3	Раздел 3. Элементы функционального программирования Функциональная парадигма. Haskell – определение функций, каррирование. Охранные выражения и сравнение с образцом. Рекурсивные алгоритмы. Работа со списками. Бесконечные списки

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 4.

Таблица 4 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	№ раздела дисциплины
Учебным планом не предусмотрено				
Всего:				

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	№ раздела дисциплины
Семестр 8			
1	Знакомство со средой Eclipse, простейшая программа. Переменные и типы данных, консольный ввод/вывод. Создание и использование объектов стандартных классов (Scanner). Определение функций. Операции с целыми числами. Использование математических функций из класса Math. Генерация случайных чисел (класс Random).	2	1
2	Сравнения и логические операции, условная операция (?:), условный оператор if. Оператор выбора switch. Циклы for и while. Массивы. Цикл for(:).	1	1
3	Определение классов: поля и методы, конструкторы классов, перегрузка методов, ограничение доступа. Пакеты, структурирование программ, создание jar-файлов.	1	1
4	Работа со строками	1	1
5	Файловый ввод/вывод (текстовые файлы) и обработка исключений.	1	1
6	Обобщённые типы (generics), контейнерная библиотека Java, простейшие алгоритмы.	2	1
7	Программа Складской учёт	2	1
8	Наследование и полиморфизм, виртуальные функции, интерфейсы на Java	2	1
9	Знакомство со средой Visual Studio: проекты, простейшая программа. Переменные и типы данных, присваивание, арифметические выражения. Простейший ввод/вывод. Математические функции, случайные числа. Целочисленные операции (в т.ч. побитовые). Условный оператор, логические операции. Циклы for и while. Массивы. Функции.	2	2
10	Работа с массивами C++	2	2
11	Функции; заголовочные файлы и многофайловая компоновка	1	2
12	Классы стандартной библиотеки string и vector	1	2
13	Указатели и динамическая память, C-строки	1	2
14	Классы. Проект «Книжный магазин»	1	2
15	Проект «Компания»	1	2
16	Перегрузка операций	1	2
17	Наследование и полиморфизм, виртуальные функции, интерфейсы на C++	2	2
18	Знакомство с языком программирования Haskell	2	3
19	Определение функций, каррирование, рекурсивные алгоритмы	4	3
20	Операции и функции для работы со списками	2	3
21	Конструкторы списков	2	3

Всего:	34	
--------	----	--

4.5. Курсовое проектирование (работа)

Учебным планом не предусмотрено

4.6. Самостоятельная работа студентов

Виды самостоятельной работы и ее трудоемкость приведены в таблице 6.

Таблица 6 Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 8, час
1	2	3
Самостоятельная работа, всего	40	40
изучение теоретического материала дисциплины (ТО)	30	30
курсовое проектирование (КП, КР)		
расчетно-графические задания (РГЗ)		
выполнение реферата (Р)		
Подготовка к текущему контролю (ТК)	10	10

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы студентов указаны в п.п. 8-10.

6. Перечень основной и дополнительной литературы

6.1. Основная литература

Перечень основной литературы приведен в таблице 7.

Таблица 7 – Перечень основной литературы

Шифр	Библиографическая ссылка / URL адрес	Количество экземпляров в библиотеке (кроме электронных экземпляров)
004.4 И 21	Иванова, Г. С. Технология программирования [Текст] : учебник / Г. С. Иванова. - М. : КноРус, 2011. - 333 с.	22
004 В 52	Вирт, Н. Алгоритмы и структуры данных. Новая версия для Оберона + CD [Текст] / Н. Вирт ; пер. Д. Б. Подшивалов. - 2-е изд., испр. - М. : ДМК Пресс, 2012. - 272 с.	10
004.4 Г 95	Гуриков, С. Р. Введение в программирование на языке Visual C# [Текст] : учебное пособие / С. Р. Гуриков. - М. : ФОРУМ : ИНФРА-М, 2013. - 448 с.	10
004 Л 85	Лупин, С. А. Технологии параллельного программирования [Текст] : учебное пособие / С. А. Лупин, М. А. Посыпкин. - М. : ФОРУМ: ИНФРА-М, 2015. - 208 с.	20
004.4 К 53	Кнут, Д. Искусство программирования [Текст] = The art of computer programming : [в 3 т.]. Т. 1. Основные алгоритмы / Д. Кнут ; ред. Ю. В. Козаченко. - 3-е изд. - М. : Вильямс, 2014. - 720 с.	5
004.4 К 84	Крук, Евгений Аврамович (проф.). Методы программирования и прикладные алгоритмы [Текст] : учебное пособие в 3 ч. Ч. 1 / Е. А. Крук, А. А. Овчинников ; С.-Петербург. гос. ун-т аэрокосм.	45

	приборостроения. - СПб. : Изд-во ГУАП, 2014. - 178 с.	
004.4 К 36	Керниган, Б. В. Язык программирования С [Текст] = The C programming language : пер. с англ. / Б. В. Керниган, Д. Ритчи. - 2-е изд., перераб. и доп. - М. и др. : Вильямс, 2016. - 288 с.	10
004.4 П 21	Пахомов, Б. И. С/С++ и MS Visual C++ для начинающих 2012 [Текст] / Б. И. Пахомов. - 2-е изд. - СПб. : БХВ - Петербург, 2015. - 528 с.	10

6.2. Дополнительная литература

Перечень дополнительной литературы приведен в таблице 8.

Таблица 8 – Перечень дополнительной литературы

Шифр	Библиографическая ссылка/ URL адрес	Количество экземпляров в библиотеке (кроме электронных экземпляров)
004.4 Э 38	Эккель, Брюс. Философия Java [Текст] / Б. Эккель. - 4-е изд. - СПб. : ПИТЕР, 2011. - 637 с.	1
004.4 С 28	Седжвик, Р. Алгоритмы на С++ : анализ структуры данных, сортировка, поиск, алгоритмы на графах [Текст] / Р. Седжвик ; конс. К. Ван Вик. - М. : Вильямс, 2014. - 1056 с.	5
004.4 О-66	Орлов, С. А. Теория и практика языков программирования [Текст] : учебник для бакалавров и магистров / С. А. Орлов. - СПб. : ПИТЕР, 2014. - 688 с.	5
004.4 Т 70	Троелсен, Э. Язык программирования С# 5.0 и платформа .NET 4.5 [Текст] = Pro C# 5.0 and the .NET 4.5 framework / Э. Троелсен. - 6-е изд. - М. и др. : Вильямс, 2015. - 1312 с.	2

7. Перечень ресурсов информационно-телекоммуникационной сети ИНТЕРНЕТ, необходимых для освоения дисциплины

Перечень ресурсов информационно-телекоммуникационной сети ИНТЕРНЕТ, необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень ресурсов информационно-телекоммуникационной сети ИНТЕРНЕТ, необходимых для освоения дисциплины

URL адрес	Наименование
www.intuit.ru/	Национальный Открытый Университет "ИНТУИТ"

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

8.1. Перечень программного обеспечения

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10 – Перечень программного обеспечения

№ п/п	Наименование
	ОС windows 7 и выше
	Среда программирования

8.2. Перечень информационно-справочных систем

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11 – Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Состав материально-технической базы представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Лекционная аудитория	
2	Специализированный компьютерный класс	

10. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

10.1. Состав фонда оценочных средств приведен в таблице 13

Таблица 13 - Состав фонда оценочных средств для промежуточной аттестации

Вид промежуточной аттестации	Примерный перечень оценочных средств
Экзамен	Список вопросов; Тесты.

10.2. Перечень компетенций, относящихся к дисциплине, и этапы их формирования в процессе освоения образовательной программы приведены в таблице 14.

Таблица 14 – Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Номер семестра	Этапы формирования компетенций по дисциплинам/практикам в процессе освоения ОП
	ОПК-1 «способность анализировать физические явления и процессы, применять соответствующий математический аппарат для формализации и решения профессиональных задач»
1	Математический анализ
1	Математическая логика и теория алгоритмов
2	Физика
2	Математический анализ
2	Учебная (ознакомительная) практика
3	Теория вероятностей и математическая статистика
3	Электротехника
3	Физика
3	Инженерная графика
4	Основы радиотехники
4	Вычислительная математика
4	Технологии и методы программирования
4	Учебная практика

4	Электроника и схемотехника
5	Мультимедиа технологии
5	Технологии обработки аудио- и видеоданных
5	Устройства и системы беспроводной связи
5	Организация ЭВМ и вычислительных систем
5	Метрология
5	Микропроцессорная техника
5	Математические основы обработки информации
6	Производственная (эксплуатационная) практика
6	Моделирование систем
6	Системное программное обеспечение
6	Операционные системы
7	Распределенные информационные системы
7	Постквантовая криптография
7	Безопасность сетей ЭВМ
7	Распределенные сети хранения данных
7	Безопасность операционных систем
8	Языки программирования
8	Теория графов и ее приложения
8	Производственная (конструкторская) практика
8	Исследование операций и теории игр
9	Научно-исследовательская работа
9	Научно-исследовательская работа
9	Защита информации в сенсорных сетях
10	Научно-исследовательская работа
10	Научно-исследовательская работа
10	Производственная преддипломная практика
ОПК-3 «способность применять языки, системы и инструментальные средства программирования в профессиональной деятельности»	
1	Информатика
2	Основы программирования
3	Основы программирования
4	Технологии и методы программирования
5	Криптографические методы защиты информации
6	Программно-аппаратные средства обеспечения информационной безопасности
6	Криптографические методы защиты информации
6	Производственная (эксплуатационная) практика
7	Методы и средства проектирования информационных систем
8	Методы и средства проектирования информационных систем
8	Языки программирования
8	Производственная (конструкторская) практика
9	Научно-исследовательская работа

9	Научно-исследовательская работа
10	Научно-исследовательская работа
10	Научно-исследовательская работа
10	Производственная преддипломная практика
ОПК-4 «способность понимать значение информации в развитии современного общества, применять достижения современных информационных технологий для поиска информации в компьютерных системах, сетях, библиотечных фондах»	
1	Промышленная экология
1	Информатика
1	Экология
2	Основы программирования
2	Учебная (ознакомительная) практика
3	Основы программирования
3	Информационные технологии
4	Основы информационной безопасности
4	Учебная практика
4	Технологии и методы программирования
4	Безопасность жизнедеятельности
5	Теория информации
6	Теория информационной безопасности
6	Производственная (эксплуатационная) практика
6	Моделирование систем
7	Техническая защита информации
8	Производственная (конструкторская) практика
8	Языки программирования
8	Защита информации в распределенных информационных системах
9	Научно-исследовательская работа
9	Научно-исследовательская работа
10	Научно-исследовательская работа
10	Научно-исследовательская работа
10	Информационная безопасность распределенных информационных систем
10	Технология построения защищенных распределенных приложений
10	Производственная преддипломная практика

10.3. В качестве критериев оценки уровня сформированности (освоения) у обучающихся компетенций применяется шкала модульно–рейтинговой системы университета. В таблице 15 представлена 100–балльная и 4–балльная шкалы для оценки сформированности компетенций.

Таблица 15 –Критерии оценки уровня сформированности компетенций

Оценка компетенции		Характеристика сформированных компетенций
100-балльная шкала	4-балльная шкала	

$85 \leq K \leq 100$	«отлично» «зачтено»	<ul style="list-style-type: none"> - обучающийся глубоко и всесторонне усвоил программный материал; - уверенно, логично, последовательно и грамотно его излагает; - опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления; - умело обосновывает и аргументирует выдвигаемые им идеи; - делает выводы и обобщения; - свободно владеет системой специализированных понятий.
$70 \leq K \leq 84$	«хорошо» «зачтено»	<ul style="list-style-type: none"> - обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; - не допускает существенных неточностей; - увязывает усвоенные знания с практической деятельностью направления; - аргументирует научные положения; - делает выводы и обобщения; - владеет системой специализированных понятий.
$55 \leq K \leq 69$	«удовлетворительно» «зачтено»	<ul style="list-style-type: none"> - обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; - допускает несущественные ошибки и неточности; - испытывает затруднения в практическом применении знаний направления; - слабо аргументирует научные положения; - затрудняется в формулировании выводов и обобщений; - частично владеет системой специализированных понятий.
$K \leq 54$	«неудовлетворительно» «не зачтено»	<ul style="list-style-type: none"> - обучающийся не усвоил значительной части программного материала; - допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; - испытывает трудности в практическом применении знаний; - не может аргументировать научные положения; - не формулирует выводов и обобщений.

10.4. Типовые контрольные задания или иные материалы:

1. Вопросы (задачи) для экзамена (таблица 16)

Таблица 16 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена
	<p>Язык программирования Java</p> <ol style="list-style-type: none"> 1. Виртуальная Java-машина. Схема компиляции и выполнения программы. Преимущества двухступенчатой компиляции. 2. Общая структура программы. Статические и экземплярные переменные. 3. Типы. Преобразование числовых типов. 4. Операции. Операторы. 5. Ввод-вывод. Класс Scanner. 6. Перечислимый тип. 7. Строки. Операции над строками. Методы типа String. 8. Функции. Параметры. Передача параметра по ссылке. 9. Одномерные массивы. Передача массива как параметра. Цикл for(x: a) 10. Двумерные массивы. 11. Классы. Класс Person. Уровни доступа. Конструкторы. Конструктор по умолчанию. Явная инициализация полей. Вызов одного конструктора из другого. Ссылочная модель хранения объектов. Сборка мусора. 12. Статические методы, поля и конструкторы. 13. Моделирование ссылочных параметров классами.

14. Пакеты. Подключение пакетов. Использование имен из других пакетов. Правила видимости. Правила именования пакетов. Статический импорт.
 15. Библиотеки `java`.
 16. Наследование. Класс `Student`. Вызов конструктора базового класса. Запрет наследования.
 17. Полиморфизм. Виртуальные функции. Обрыв цепочки виртуальности. Полиморфные контейнеры.
 18. `UpCast` и `DownCast` для предков и потомков.
 19. Класс `Object`, его методы.
 20. Переопределение `equals` и `hashCode` в потомках.
 21. Определение типа во время выполнения. Операция `instanceof`. Метод `getClass`. Класс `Class`. Пример: родословная объекта.
 22. Объектные оболочки и автоупаковка. Распаковка.
 23. Классы-перечисления.
 24. Интерфейсы. Реализация интерфейса. Примеры реализации интерфейсов.
 25. Использование интерфейса `Comparable<T>` для сортировки массива `Person`.
 26. Исключения. Обработка и генерация исключений. Иерархия стандартных исключений. Спецификация исключений. Диагностические утверждения.
 27. Файловые потоки. Иерархия классов файловых потоков. Комбинирование потоковых классов. Текстовые потоки. Файлы произвольного доступа.
 28. Обобщенные классы и методы. Какие типы могут быть параметрами обобщенных классов. Действия, которые можно делать с объектами типа-параметра обобщенного класса. Задание условий на типы-параметры. ? в параметрах.
 29. Коллекции. Интерфейсы `java.util.Collection<E>` и `java.util.Iterator<E>`. Обход коллекции с помощью итератора.
 30. Иерархия интерфейсов коллекций. Иерархия классов коллекций.
 31. Интерфейсы `List<E>`, `ListIterator<E>`.
 32. Классы `ArrayList<E>`, `LinkedList<E>`.
 33. Интерфейсы `Queue<E>`, `Deque<E>`, `Set<E>`, `SortedSet<E>`, `NavigableSet<E>`. Классы `HashSet<E>`, `TreeSet<E>`.
 34. Интерфейс `Comparator<E>` и его использование.
 35. Ассоциативные массивы. Интерфейсы `Map<K,V>`, `SortedMap<K,V>`. Классы `HashMap<K,V>`, `TreeMap<K,V>`. Цикл по ассоциативному массиву.
- Язык программирования C++**
36. Простейшая программа. Ввод-вывод.
 37. Пространства имен. Коллизия имен из разных пространств и ее разрешение.
 38. Препроцессор, его роль. Основные директивы препроцессора.
 39. Основные типы. Неявные преобразования типов.
 40. Основные операции и операторы.
 41. Функции. Передача параметров по ссылке. Понятие ссылки.
 42. Массивы. Хранение в памяти. Передача массивов в функции.
 43. Строки как массивы символов. Ввод строк.
 44. Двумерные массивы и их хранение в памяти.
 45. Определение типов.
 46. Указатели. Передача параметров с помощью указателей.
 47. Бестиповые указатели. Правила преобразования типов для них. Ошибки, связанные с приведением типов указателей.
 48. Структуры. Указатели на структуры.
 49. Указатели и константность. Указатели на константы и константные указатели. Ссылки и константность.
 50. Операции с указателями. Связь одномерных массивов и указателей.
 51. Передача одномерных массивов в подпрограммы.

52. Идиома *r++, пример использования.
 53. Указатели и C-строки. Стандартные функции работы с C-строками.
 54. inline-функции.
 55. Указатели на функции. Callback-вызовы.
 56. Предварительное объявление.
 57. Многофайловая компоновка. Схема компиляции программы. Ее отличие от схемы компиляции на Паскале.
 58. Содержимое заголовочных файлов. Что нельзя помещать в заголовочные файлы.
 59. Стражи включения.
 60. Предкомпилируемые заголовки.
 61. Ошибки и особенности компоновки.
 62. Указатели и динамическая память.
 63. Двумерные динамические массивы. Передача двумерного массива как параметра функции.
 64. Использование указателей для создания динамических структур данных.
 65. Строки string, их операции и методы.
 66. Тип vector, его операции. Понятие емкости вектора.
 67. Классы. Защита доступа. Пример: класс Date.
 68. Операции над типами. Перегрузка бинарных операций.
 69. Операции над типами. Перегрузка унарных операций.
 70. Дружественные функции. Операции ввода-вывода.
 71. Класс динамического массива. Операция []. Деструкторы, момент вызова деструктора.
 72. Объекты классов в динамической памяти.
 73. Конструктор копии.
 74. Операция присваивания.
 75. Ситуации, в которых вызывается конструктор копии.
 76. Шаблоны классов и функций. Описание шаблонов. Процесс компиляции шаблонов. Отличие от обобщенных классов и методов в Java и .NET.
 77. Массив объектов класса.
 78. Класс matrix. Инициализация подобъекта. Порядок вызова конструкторов и деструкторов. Список инициализации. Операция ().
 79. Класс frac. Конструктор преобразования. Ключевое слово explicit. Операция приведения типа.
 80. Наследование. Порядок вызова конструкторов и деструкторов. Конструктор копии и операция присваивания для потомка.
 81. Принцип «Выделение ресурса есть инициализация»
 82. Преобразование типов в иерархии «Предок-Потомок» для объектов, указателей и ссылок. static_cast и его проблемы.
 83. Множественное наследование. Ромбовидное наследование и виртуальные базовые классы. Проблемы множественного наследования.
 84. Полиморфизм и виртуальные функции. Особенности реализации полиморфизма в C++.
 85. Накладные расходы на аппарат виртуальных функций.
 86. Полиморфные контейнеры. Виртуальные деструкторы.
 87. Операция dynamic_cast.
 88. Операция typeid и структура type_info.
 89. Исключения в C++.
- Элементы функционального программирования**
90. Основные особенности функциональной парадигмы программирования.
 91. Функции Haskell: определение, вызов. Виды определения функций: с

	<p>помощью if, охранных выражений и сопоставления с образцом.</p> <p>92. Определение и вызов бинарных операций.</p> <p>93. Секции where и let в функциях.</p> <p>94. Функции как параметры функций.</p> <p>95. Каррирование. Сечение бинарной функции в инфиксной форме.</p> <p>96. Рекурсия. Примеры.</p> <p>97. Списки. Основные операции над списками. Основные функции для работы со списками. Бесконечные списки.</p> <p>98. Примеры реализации основных функций со списками.</p> <p>99. Конструкторы списков. Быстрая сортировка.</p> <p>Конструкторы списков. Решето Эратосфена.</p>
--	---

2. Вопросы (задачи) для зачета / дифференцированного зачета (таблица 17)

Таблица 17 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифференцированного зачета
	Учебным планом не предусмотрено

3. Темы и задание для выполнения курсовой работы / выполнения курсового проекта (таблица 18)

Таблица 18 – Примерный перечень тем для выполнения курсовой работы / выполнения курсового проекта

№ п/п	Примерный перечень тем для выполнения курсовой работы / выполнения курсового проекта
	Учебным планом не предусмотрено

4. Вопросы для проведения промежуточной аттестации при тестировании (таблица 19)

Таблица 19 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов
	<ol style="list-style-type: none"> 1. Что такое агрегация <ol style="list-style-type: none"> a. Любое взаимодействие между объектами класса b. Отношение между классами, при котором один класс наследует интерфейс другого класса c. Отношение между классами, при котором один класс содержит в качестве поля объект другого класса 2. Дружественные функции <ol style="list-style-type: none"> a. имеют доступ к секции private предка b. имеют доступ к секции protected предка c. имеют доступ к секции public предка 3. Что является интерфейсом на C++ <ol style="list-style-type: none"> a. <code>class Intf { public: virtual void p(); };</code> b. <code>class Intf { public: void p(); ~A(){};};</code> c. <code>class Intf { public: virtual void p() = 0; virtual ~A(){};};</code> d. <code>class Intf { int n; public: virtual void p();virtual ~A(){};};</code> 4. Верны ли утверждения: <ol style="list-style-type: none"> a. Полиморфизм – это возможность создавать несколько объектов одного класса b. Полиморфизм – это возможность родственных классов выполнять одни и те же действия по-разному c. Полиморфизм – это зависимость между классами при множественном наследовании 5. В каких языках есть множественное наследование классов <ol style="list-style-type: none"> a. PascalABC.NET

	<ul style="list-style-type: none"> b. Java c. C# d. C++ <p>6. В каких языках есть ключевое слово interface</p> <ul style="list-style-type: none"> a. PascalABC.NET b. Java c. C# d. C++ <p>7. В каком порядке следует записывать обработчики исключений</p> <ul style="list-style-type: none"> a. в произвольном b. от общих к частным c. от частных к общим <p>8. В какую сторону можно осуществлять присваивание в иерархии Предок-Потомок</p> <ul style="list-style-type: none"> a. нельзя вообще – это несовместимые типы b. предку можно присваивать потомка, но не наоборот c. в обе стороны d. потомку можно присваивать предка но не наоборот <p>9. Как перегрузить постфиксную операцию ++ в C++</p> <ul style="list-style-type: none"> a. <code>class A{ operator++(); };</code> b. <code>class A{ A operator++(); };</code> c. <code>class A{ A& operator++(int); };</code> d. <code>class A{ A operator++(); };</code> e. <code>class A{ A operator++(int); };</code> <p>10. Когда используется dynamic_cast</p> <ul style="list-style-type: none"> a. для всех типов b. для классов, связанных иерархией наследования c. для всех классов <p>11. Что возвращает dynamic_cast для ссылки в случае невозможности преобразования</p> <ul style="list-style-type: none"> a. ничего b. исключение c. 0 <p>12. В каких языках все классы – потомки одного класса</p> <ul style="list-style-type: none"> a. PascalABC.NET b. Java c. C# d. C++
--	---

5. Контрольные и практические задачи / задания по дисциплине (таблица 20)

Таблица 20 – Примерный перечень контрольных и практических задач / заданий

№ п/п	Примерный перечень контрольных и практических задач / заданий

10.5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и / или опыта деятельности, характеризующих этапы формирования компетенций, содержатся в Положениях «О текущем контроле успеваемости и промежуточной аттестации студентов ГУАП, обучающихся по программам высшего образования» и «О модульно-рейтинговой системе оценки качества учебной работы студентов в ГУАП».

11. Методические указания для обучающихся по освоению дисциплины

Сформировать у учащихся уверенные знания и навыки программирования на нескольких современных языках программирования, дать их сравнительную характеристику, заложить способность легко разбираться в новых языках программирования, дать последовательное изложение объектно-ориентированного программирования, особенностей его использования на нескольких языках.

Основными задачами освоения дисциплины являются:

- Знакомство студентов с несколькими распространенными современными языками программирования, их сравнительный анализ.
- Сравнение организации ссылочной и размерной объектной модели в различных языках.
- Изучение стандартных библиотек языков C++, Java.
- Знакомство с функциональной парадигмой программирования.

Методические указания для обучающихся по освоению лекционного материала

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимся лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально–деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

- Изложение лекционного материала;
- Представление теоретического материала преподавателем в виде слайдов;
- Освоение теоретического материала по практическим вопросам;
- Список вопросов по теме для самостоятельной работы студента (Табл.21).

Методические указания для обучающихся по прохождению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач у обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;

- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ

Задание на лабораторные работы представлены по темам изучаемой дисциплины и представляют собой программную реализацию изучаемых задач:

- создание консольного java-приложения с использованием массивов, строк и файлов.
- создание консольного приложения, позволяющего манипулировать строкой, разбив ее на элементы с помощью регулярных выражений.
- управление наборами данных в приложениях java с помощью специальных классов-коллекций, позволяющих создавать типовые структуры, такие как список, очередь, стек и т.п..
- изучение принципов построения java-приложений с графическим пользовательским интерфейсом с помощью библиотек java.awt и javax.swing.
- разработка сетевых приложений, ориентированных на работу в сетях TCP/IP (как локальных, так и в сети Интернет).
- разработка веб-приложения (J2EE) и размещение его на локальном сервере приложений под управлением Tomcat.

Структура и форма отчета о лабораторной работе

Отчёт по лабораторной работе оформляется индивидуально каждым студентом, выполнившим необходимые (независимо от того, выполнялся ли эксперимент индивидуально или в составе группы студентов). Страницы отчёта следует пронумеровать (титульный лист не нумеруется, далее идет страница 2 и т.д.). Титульный лист отчёта должен содержать фразу: «Отчёт по лабораторной работе «Название работы», чуть ниже: Выполнил студент группы (номер группы) (Фамилия, инициалы)». Внизу листа следует указать текущий год. Например, Отчёт по лабораторной работе № (номер работы) «Введение в спектральный анализ», Выполнил студент группы 5221 Иванов И.И. Вторая страница текста, следующая за титульным листом, должна начинаться с пункта: Цель работы. Отчёт, как правило, должен содержать следующие основные разделы:

1. Цель работы;
2. Теоретическая часть;
3. Программное обеспечение, используемое в работе;
4. Результаты;
5. Выводы.

В случае необходимости в конце отчёта приводится перечень литературы.

Требования к оформлению отчета о лабораторной работе

Теоретическая часть должна содержать минимум необходимых теоретических сведений о предметной области. Не следует копировать целиком или частично методическое пособие (описание) лабораторной работы или разделы учебника.

В разделе Программное обеспечение необходимо описать, с помощью каких инструментальных средств и каким образом были разработаны модели и получены результаты. Рисунки, блок-схемы, описание модели и её особенностей, необходимость отладки – все это должно быть представлено в указанном разделе.

Раздел Результаты включает в себя скриншоты программного приложения, полученные при выполнении лабораторной работы. Рисунки, графики и таблицы нумеруются и подписываются заголовками.

Выводы не должны быть простым перечислением того, что сделано. Здесь важно отметить, какие новые знания о предмете исследования были получены при выполнении работы, к

чему привело обсуждение результатов, насколько выполнена заявленная цель работы. Выводы по работе каждый студент делает самостоятельно. В случае необходимости в конце отчёта приводится Список литературы, использованной при подготовке к работе. В тексте отчёта делаются краткие ссылки на литературу (учебники, справочники, иные источники...) номером в квадратных скобках, напр., [1]. Литературные источники нумеруются по мере их появления в тексте отчёта. В конце отчёта даётся их подробный список. На все источники списка литературы должны быть ссылки в тексте отчёта, там, где это необходимо. При сдаче отчёта преподаватель может сделать устные и письменные замечания, задать дополнительные вопросы. Все ответы на дополнительные вопросы, обсуждения выполняются студентом на отдельных листах, включаемых в отчёт (при этом в тексте основного отчёта делается сноска или другой значок, которому будет соответствовать новый материал). При этом письменные замечания преподавателя должны остаться в тексте для ясности динамики работы над отчётом. Объём отчёта должен быть оптимальным для понимания того, что и как сделал студент, выполняя работу. Обязательные требования к отчёту включают общую и специальную грамотность изложения, а также аккуратность оформления. После приёма преподавателем отчёт хранится на кафедре.

Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине.

Методические указания для обучающихся по прохождению промежуточной аттестации

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

– экзамен – это форма оценки знаний, полученных обучающимся при изучении дисциплины, при выполнении курсовых проектов, курсовых работ, научно-исследовательских работ и прохождении практик с аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Система оценок при проведении промежуточной аттестации осуществляется в соответствии с требованиями Положений «О текущем контроле успеваемости и промежуточной аттестации студентов ГУАП, обучающихся по программам высшего образования» и «О модульно-рейтинговой системе оценки качества учебной работы студентов в ГУАП».

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой