

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра №34

«УТВЕРЖДАЮ»
Руководитель направления
проф. д.т.н., доц.
(должность, уч. степень, звание)
 С.В. Безруков
(подпись)
«24» июня 2021 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования»
(Название дисциплины)

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

Лист согласования рабочей программы дисциплины

Программу составил(а)

доц., к.т.н., доц.

(должность, уч. степень, звание)

 24.06.21
подпись, дата

Т.Н. Елина
инициалы, фамилия

Программа одобрена на заседании кафедры № 34

«24» июня 2021 г., протокол № 11

Заведующий кафедрой № 34

проф. д.т.н., доц.

(должность, уч. степень, звание)

«24» июня 2021 г.

(подпись, дата)



С.В. Безруков
инициалы, фамилия

Ответственный за ОП 10.05.03(07)

доц., к.т.н., доц.

(должность, уч. степень, звание)

 24.06.21
подпись, дата

В.А. Мыльников
инициалы, фамилия

Заместитель директора института (декана факультета) № 3 по методической работе

доц., к.т.н., доц.

(должность, уч. степень, звание)

 24.06.21
подпись, дата

Г.С. Аршавина-Тельник
инициалы, фамилия

Аннотация

Дисциплина «Основы 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-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»;

профессиональных компетенций:

ПК-1 «способность осуществлять поиск, изучение, обобщение и систематизацию научно-технической информации, нормативных и методических материалов в сфере профессиональной деятельности, в том числе на иностранном языке».

Содержание дисциплины охватывает круг вопросов, связанных с изучением основных особенностей разработки программ на одном из языков программирования высокого уровня, разработка и использование составных типов данных, разработка и создание динамических переменных, динамических структур данных, разработка программ с использованием современных инструментальных средств с использованием технологии быстрой разработки приложений (RAD). Формирование у будущих специалистов твердых теоретических знаний и практических навыков по разработке программ с использованием языка высокого уровня C++.

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

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

Общая трудоемкость освоения дисциплины составляет 10 зачетных единиц, 360 часов.

Язык обучения по дисциплине «русский».

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Целью дисциплины является изучение основных особенностей разработки программ на одном из языков программирования высокого уровня, разработка и использование составных типов данных, разработка и создание динамических переменных, динамических структур данных, разработка программ с использованием современных инструментальных средств с использованием технологии быстрой разработки приложений (RAD). Формирование у будущих специалистов твердых теоретических знаний и практических навыков по разработке программ с использованием языка высокого уровня C++. Основной задачей изучения дисциплины является реализация требований, установленных в Государственном стандарте высшего профессионального образования к подготовке специалистов. Изучение методов и способов разработки алгоритмов и программ с использованием языка высокого уровня C++. В ходе изучения дисциплины ставятся следующие задачи: приобретение системы знаний о разработке программ как одной из функций деятельности информатика, позволяющей принимать соответствующие решения при решении задач в экономической отрасли; организация разработки информационной системы для широкого круга внутренних и внешних пользователей предприятия; формирование и представление информации, использования её при разработке приложений, удовлетворяющей требованиям различных пользователей программного обеспечения.

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

В результате освоения дисциплины обучающийся должен обладать следующими компетенциями:

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

знать – способы формулировки задач для решения на ЭВМ; современные методы и средства разработки приложений на языках высокого уровня

уметь – самостоятельно разрабатывать, отлаживать, тестировать и документировать программы на языке СИ/СИ++

владеть навыками – применения основных методов и средств проектирования и разработки приложений с использованием языков программирования высокого уровня

иметь опыт деятельности – использования программных комплексов для решения поставленных задач;

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 – основные приемы структурного и модульного программирования используемые при разработке программ; способы записи и документирования алгоритмов и программ, составления программной документации; способы отладки и испытания программ;

уметь – использовать стандартные модули при построении приложений

владеть навыками – использования языков программирования, построения логически правильных и эффективных программ

иметь опыт деятельности – разработки прикладных приложений конкретной предметной области;

ПК-1 «способность осуществлять поиск, изучение, обобщение и систематизацию научно-технической информации, нормативных и методических материалов в сфере профессиональной деятельности, в том числе на иностранном языке»:

знать – основные приемы структурного и модульного программирования используемые при разработке программ; способы записи и документирования алгоритмов и программ, составления программной документации; способы отладки и испытания программ;

уметь – использовать стандартные модули при построении приложений

владеть навыками – использования языков программирования, построения логически правильных и эффективных программ

иметь опыт деятельности – разработки прикладных приложений конкретной предметной области.

2. Место дисциплины в структуре ОП

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

- Информатика
- Учебная (ознакомительная) практика

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

- Технологии и методы программирования
- Программно-аппаратные средства обеспечения информационной безопасности
- Языки программирования
- Научно-исследовательская работа

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	
		№2	№3
1	2	3	4
Общая трудоемкость дисциплины, ЗЕ/(час)	10/ 360	5/ 180	5/ 180
Из них часов практической подготовки	32	16	16
Аудиторные занятия, всего час., В том числе	153	85	68
лекции (Л), (час)	51	34	17
Практические/семинарские занятия (ПЗ), (час)	17	17	
лабораторные работы (ЛР), (час)	68	34	34
курсовой проект (работа) (КП, КР), (час)	17		17
Экзамен, (час)	81	27	54
Самостоятельная работа, всего	126	68	58

Вид промежуточного контроля: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.)	Экз., Экз.	Экз.	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) (час)	ЛР (час)	КП (час)	СРС (час)
Семестр 2					
Раздел 1. Технология создания алгоритмов программ	6	6			10
Раздел 2. Программирование обработки текстовой информации	2	6	6		10
Раздел 3. Сложные типы данных	2	5	6		10
Раздел 4. Основы конструирования программ методом функциональной декомпозиции. Модульное программирование	6		6		10
Раздел 5. Разработка приложений работы с файлами и файловой структурой	2		4		10
Раздел 6. Разработка приложений с использованием динамических структур данных	8		6		10
Раздел 7. Рекурсивные структуры данных и рекурсивные алгоритмы обработки	8		6		8
Итого в семестре:	34	17	34		68
Семестр 3					
Раздел 8. Разработка консольных приложений с использованием стандартного модуля <code>conio.h</code>	5		12		20
Раздел 9. Разработка графических приложений с использованием стандартного графического модуля	6		12		20
Раздел 10. Основы разработки приложений с использованием инструментальной среды разработки и отладки программ C++ Builder	6		10		18
Выполнение курсовой работы				17	
Итого в семестре:	17		34	17	58
Итого:	51	17	68	17	126

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
Раздел 1. Технология создания алгоритмов программ	Введение. Основы теории алгоритмов. Общие принципы построения алгоритмов. Основные алгоритмические конструкции. Линейная алгоритмическая структура. Разветвляющаяся алгоритмическая структура. Виды ветвлений. Циклическая алгоритмическая структура. Виды циклов
Раздел 2. Программирование обработки текстовой информации	Множество значений, операции, функции работы со строками. Определение переменных. Ввод и вывод значений переменных типа сорокового типа. Способы представления текстов в программах: наборы символов, строки символов, наборы строк. Математические модели строки символов: представление с длиной, с маркером конца. Понятие "подстрока". Математические модели операций над строками и подстроками. Примеры. Представление строки знаков массивами языка СИ/СИ++ и реализация действий со строками. Типовые задачи и приемы обработки текстов. Действия с текстами, разделенными на слова. Рациональные и эффективные приемы программирования обработки текстов.
Раздел 3. Сложные типы данных	Структуры. Назначение Структур. Структуры как тип данных. Номенклатура структур в языке СИ/СИ++. Структуры с фиксированным числом компонент. Правила построения описания структур. Описание переменных. Действия со значениями структуры и ее компонент. Применение структур для представления текстовой информации. Примеры: реализация типовых приемов обработки строк. Иерархические структуры. Описание и семантика типа данных. Описание переменных. Операции над иерархическими структурами.
Раздел 4. Основы конструирования программ методом функциональной декомпозиции. Модульное программирование	Основные этапы конструирования программ: проектирование, кодирование, отладка и тестирование. Метод функциональной декомпозиции. Определение функций частей программы. Построение иерархии функций. Сопоставление функциям частей (модулей) программы. Модуль. Конструирование модуля. Спецификация модуля. Проектирование логики модуля. Реализация на языке программирования.
Раздел 5. Разработка приложений работы с файлами и файловой структурой	Концепция файла. Файл как тип данных. Определение файлового типа и переменных в языке СИ/СИ++. Внутренние и внешние файловые переменные. Буферная переменная. 5 Стандартные процедуры для работы с файлами: состав набора, назначение и семантика процедур. Взаимосвязь процедур. Примеры применения. Типовые задачи обработки файлов: формирование файла, просмотр, слияние файлов, сортировка файла. Текстовые файлы. Назначение, структура, описание в языке СИ/СИ++. Стандартные процедуры для работы с текстовыми файлами. Примеры. Применение текстовых файлов для представления и обработки текстовой информации.
Раздел 6. Разработка приложений с использованием динамических структур данных	Статические и динамические объекты программы. Концепция ссылки (указателя). Ссылочный тип данных. Определение типа в языке СИ/СИ++: синтаксис и семантика. Основные операции. Переменные ссылочного типа (переменные- указатели). Примеры организации динамических объектов. Действия с динамическими объектами: создание, уничтожение, присваивание значения. Технология программирования обработки динамических объектов. Линейный список. Типовые структуры. Основные операции: формирование списка, включение элемента, удаление элемента, просмотр списка, поиск элемента, удаление списка. Представление в программе на языке СИ/СИ++ и примеры применения. Технология надежного программирования обработки. Двухнаправленный список. Логические структуры. Представление в программе. Действия с элементом двухнаправленного списка и со списком как целым. Примеры применения. Кольцевой список. Логические структуры. Представление в программе. Действия со списком как целым и с его элементами. Примеры применения. Мультисписок. Логическая структура. Представление в программе. Основные действия по обработке. Примеры применения: реализация

	вычислений с "редкими" матрицами.
Раздел 7. Рекурсивные структуры данных и рекурсивные алгоритмы обработки	Рекурсивные процедуры в языке СИ/СИ++. Структура и правила построения описаний рекурсивных процедур. Вызов и семантика исполнения рекурсивных процедур. Примеры применения. Рекомендации по надежному и эффективному программированию и использованию.
Раздел 8. Разработка консольных приложений с использованием стандартного модуля conio.h	Расширенные возможности работы с монитором в текстовом режиме. Процедуры и функции для работы с клавиатурой.
Раздел 9. Разработка графических приложений с использованием стандартного графического модуля	Режимы работы монитора. Инициализация графического режима. Характеристики элемента экрана. Графические координаты экрана. Примитивы, используемые при построении приложений.
Раздел 10. Основы разработки приложений с использованием инструментальной среды разработки и отладки программ С++ Builder	Интегрированная среда разработки программ. Структура проекта. Использование компонентов. Обработка события создания формы. Обработка событий «мышь». Обработка событий нажатия клавиш клавиатуры. Разработка приложений с использованием записей и файлов. Использование средств обработки графической информации

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)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 2				
1	Разработка приложений, использующих строковые типы данных	3	1	2
2	Обработка строк	3	1	2
3	Создание структур	3	1	3
4	Разработка приложений с использованием структур	3	1	3
5	Разработка приложений с использованием динамических переменных	2	1	4
6	Использование указателей для работы с динамическими переменными	2	1	4

7	Использование динамической структуры данных: стек	2	1	4
8	Использование динамической структуры данных: список	2	1	5
9	Использование динамической структуры данных: очередь	2	1	5
10	Создание динамических массивов	3	2	6
11	Разработка приложений с использованием динамических массивов	3	2	6
12	Разработка приложений с использованием файлового ввода	3	2	7
13	Разработка приложений с использованием файлового вывода	3	2	7
Семестр 3				
14	Разработка приложений в среде Builder	2		8
15	Разработка программ с использованием инструментария Builder	3		8
16	Структура проекта Builder	2		8
17	Использование компонент Builder	2		8
18	Программирование с использованием компонент работы со строками – Типы данных для работы со строками; – Используемые компоненты для работы со строками; – Обработка событий	2		9
19	Разработка приложений с использованием диалогов для выбора файлов и главного меню	2	2	9
20	Компоненты диалога с пользователем	2	2	9
21	Приёмы программирования со структурами	2	1	10
22	Приёмы программирования с файлами	2		10
23	Разработка приложений с использованием средств для отображения графической информации	2		10
24	Разработка приложения для работы с таблицей данных	2		10
25	Обработка информации однотабличной базы данных	2		10
26	Печать данных	2		10
Всего:		51	21	

4.5. Курсовое проектирование (работа)

Цель курсовой работы: изучить методику разработки приложений с использованием современных инструментальных средств.

Часов практической подготовки: 11

Примерные темы заданий на курсовую работу приведены в разделе 10 РПД.

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 6.

Таблица 6 Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 2, час	Семестр 3, час
1	2	3	4
Самостоятельная работа, всего	110	60	50
изучение теоретического материала дисциплины (ТО)	8	4	4
курсовое проектирование (КП, КР)			
расчетно-графические задания (РГЗ)			
выполнение реферата (Р)			
Подготовка к текущему контролю (ТК)	8	4	4
домашнее задание (ДЗ)			
контрольные работы заочников (КРЗ)			

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

Учебно-методические материалы для самостоятельной работы обучающихся 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 Т 33	Теория языков программирования и методы трансляции [Текст] : методические указания к выполнению курсовой работы / С.-Петербург. гос. ун-т аэрокосм. приборостроения ; сост. Т. М. Максимова. - СПб. : Изд-во ГУАП, 2011. - 39 с. : табл. - Библиогр.: с. 26 (5 назв.). - Б. ц.	130
004.4 Б 46	Бенкен, Е. PHP, MySQL, XML программирование для Интернета [Текст] / Е. Бенкен. - 3-е изд. - СПб. : БХВ - Петербург, 2011. - 304 с. : рис. + 1 эл. опт. диск (CD-ROM). - Предм. указ.: с. 285 - 288. - ISBN 978-5-9775-0724-0 : 328.00 р.	6
004.4 И 21	Иванова, Г. С. Технология программирования [Текст] : учебник / Г. С. Иванова. - М. : КноРус, 2011. - 333 с. : рис., табл. - Библиогр.: с. 329 - 331.	22

	- Предм. указ.: с. 332 - 333. - ISBN 978-5-406-00519-4 : 297.00 р. Издание имеет гриф УМО по университетскому политехническому образованию.	
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. : табл. + 1 эл. опт. диск (CD-ROM). - (Классика программирования). - Библиогр. в конце глав. - ISBN 978-5-94074-734-5 (рус.). - ISBN0-13-022005-9 (англ.) : 289.00 р. Приложение: Алгоритмы и структуры данных. - М. : ДМК Пресс, 2012. - 1. Шифр - 00742	10
004 С 30	Семакин, И. Г. Основы алгоритмизации и программирования [Текст] : учебник для СПО / И. Г. Семакин, А. П. Шестаков. - 3-е изд. стер. - М. : Академия, 2012. - 391 с. : рис. - (Среднее профессиональное образование. Информатика и вычислительная техника). - ISBN 978-5-7695-8957-7 : 624.80 р. Имеет гриф Минобрнауки РФ	7

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 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. : рис. - (Высшее образование: Бакалавриат). - Библиогр.: с. 446 (4 назв.). - ISBN 978-5-91134-738-3 (ФОРУМ). - ISBN978-5-16-006674-5 (ИНФРА-М) : 636.90 р. На с. 272 -445: Приложения: Примеры заполнения лабораторных работ. Создание заставок к проектами	10
004.4 Э 38	Эккель, Брюс. Философия Java [Текст] / Б. Эккель. - 4-е изд. - СПб. : ПИТЕР, 2011. - 637 с. - (Библиотека программиста). - Алф. указ.: с. 631 - 637. - ISBN 978-5-388-00003-3 : 650.00 р.	1

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
	Не предусмотрено

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	Компьютерная лаборатория	

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 к экзамену; Задачи; Тесты.
Выполнение курсовой работы	Экспертная оценка на основе требований к содержанию курсовой работы по дисциплине.

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-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
ПК-1 «способность осуществлять поиск, изучение, обобщение и систематизацию научно-технической информации, нормативных и методических материалов в сфере профессиональной деятельности, в том числе на иностранном языке»	
2	Основы программирования
3	Основы программирования
8	Теория графов и ее приложения
9	Научно-технический семинар
10	Научно-технический семинар

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 ≤ K ≤ 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 ≤ 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
	<ol style="list-style-type: none"> 1 Понятие указателя. Синтаксис описания указателя. 2 Инициализация указателей. 3 Операции с указателями. Адресная арифметика. 4 Понятие ссылки. Синтаксис описания. 5 Структура как тип данных. Синтаксис описания типа. 6 Инициализация структур. 7 Односвязный список. Добавление элемента в начало списка. 8 Односвязный список. Добавление элемента в середину списка. 9 Односвязный список. Удаление элемента из начала списка. 10 Односвязный список. Удаление элемента из середины списка. 11 Понятие стека. Занесение и выборка элементов из вершины стека. 12 Изменение и чтение элемента без его выборки из стека. 13 Понятие очереди. Организация очередей. 14 Инициализация очереди. 15 Выборка элемента из очереди и проверка очереди на пустоту. 16 Добавление элемента в конец очереди. 17 Понятие динамического массива. 18 Инициализация динамических массивов, размещение их в динамической области памяти. 19 Использование массивов в качестве параметров функции. 20 Указатели на функции и их использование. 21 Понятие файлов в C++. 22 Поточковый ввод-вывод, основные понятия. 23 Открытие, закрытие потокового ввода-вывода в файл. 24 Дополнительные функции для работы с потоковым вводом-выводом в файл. 25 Режимы файлов. 26 Понятие двоичных файлов. 27 Работа с двоичными файлами. 28 Произвольный доступ к файлам. 29 Интегрированная Среда Разработки (Integrated Development Environment (IDE) C++Builder 6. Основные элементы. 30 Состав инструментальных панелей быстрых кнопок C++Builder. 31 Понятие компонентов C++Builder . Разновидности компонентов. 32 Свойства компонентов. Понятие и их назначение. 33 События, связанные с компонентами. Понятие и их назначение. 34 Последовательность создания консольного приложения в C++Builder.

	35 Разработка простейшего приложения Windows в C++Builder. 36 Форма (объект тип TForm). Основные свойства формы. 37 Свойства компонента Label (поле вывода текста). 38 Свойства компонента Edit (поле редактирования). 39 Свойства компонента Button (командная кнопка). 40 Свойства компонента Memo. 41 Свойства компонента RadioButton. 42 Свойства компонента CheckBox. 43 Свойства компонента ListBox. 44 Свойства компонента ComboBox. 45 Свойства компонента StringGrid. 46 Свойства компонента image. 47 Свойства компонента Timer. 48 Разновидности событий.
--	---

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
	1. Разработка приложений с использованием методов численного интегрирования. 2. Разработка приложений с использованием методов численного дифференцирования. 3. Разработка приложений с решением задач Коши методом Эйлера. 4. Разработка приложений с решением задач Коши методом Рунге- Кутта. 5. Разработка приложений с решением задач Коши методом прогноза и коррекции. 6. Разработка приложений с решением системы линейных уравнений методом Гаусса. 7. Разработка приложений с решение нелинейных и трансцендентных уравнений методами простых итераций, Ньютона (касательных), дихотомии (метод деления отрезка пополам). 8. Разработка приложения одномерной минимизации функции методом золотого сечения. 9. Разработка приложения минимизации функции методом градиентного спуска. 10. Разработка графических приложений. 11. Разработка простейших баз данных. 12. Разработка файловых менеджеров. 13. Разработка приложений с использованием графики. 14. Разработка компьютерных игр.

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

Таблица 19 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов
	<p>Какие среды программирования (IDE) предназначены для разработки программных средств?</p> <ul style="list-style-type: none"> ○ MVS, NetBeans, QT Creator, RAD Studio, Dev-C++ ○ MVS, Code::Blocks, QT Creator, AutoCAD, Eclipse ○ MVS, Code::Blocks, QT Creator, RAD Studio, MathCAD <p>Структура объявления переменных в C++</p> <ul style="list-style-type: none"> ○ [=], <идент. 2>,...; ○ [=];<идент. 2>,...; ○ [:=], <идент. 2>,...; ○ [==]; <идент. 2>,...; <p>4. Какой из следующих операторов - оператор сравнения двух переменных?</p> <ul style="list-style-type: none"> ○ == ○ equal ○ := ○ = <p>5. Какой из ниже перечисленных операторов, не является циклом в C++?</p> <ul style="list-style-type: none"> ○ for ○ while ○ repeat until ○ do while <p>6. Какой из перечисленных типов данных не является типом данных в C++?</p> <ul style="list-style-type: none"> ○ float ○ real ○ int ○ double <p>7. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на результирующем языке, называется:</p> <ul style="list-style-type: none"> ○ интерпретатор ○ транслятор ○ компилятор ○ сканер <p>8. Какой служебный знак ставится после оператора case ?</p> <ul style="list-style-type: none"> ○ ; ○ . ○ - ○ : <p>9. Чему будет равна переменная a, после выполнения этого кода <code>int a; for(a = 0; a < 10; a++) {}</code>?</p> <ul style="list-style-type: none"> ○ 1 ○ 9

	<ul style="list-style-type: none"> ○ 10 <p>10. Название C++ предложил</p> <ul style="list-style-type: none"> ○ Рик Масситти ○ Дональд Кнут ○ Кэн Томпсон ○ Бьерн Страуструп
--	--

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
	<p>Назначение инструментального комплекса. Использование стандартной библиотеки классов. Компиляция программ и опции. Особенности отладки программ. Изучение методов разработки приложений Процедурное программирование Объектно-ориентированное программирование Визуальное программирование (RAD технология) Создание динамических переменных. Разновидности динамических структур данных. Организация работы с динамическими структурами. Разновидности файлов. Стандартные процедуры для работы с файлами. Типовые задачи обработки файлов. Разновидности рекурсии. Способы разработки рекурсивных алгоритмов. Глубина рекурсии. Подключение библиотек. Инициализация функций. Способы передачи параметров. Разметка экрана в графическом режиме. Инициализация графического режима. Свойства графических приметив. Структура проекта. Разновидности компонентов. Использование компонентов. Обработка событий.</p>

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

Целью дисциплины является изучение основных особенностей разработки программ на одном из языков программирования высокого уровня, разработка и использование

составных типов данных, разработка и создание динамических переменных, динамических структур данных, разработка программ с использованием современных инструментальных средств с использованием технологии быстрой разработки приложений (RAD). Формирование у будущих специалистов твердых теоретических знаний и практических навыков по разработке программ с использованием языка высокого уровня C++. Основной задачей изучения дисциплины является реализация требований, установленных в Государственном стандарте высшего профессионального образования к подготовке специалистов. Изучение методов и способов разработки алгоритмов и программ с использованием языка высокого уровня C++. В ходе изучения дисциплины ставятся следующие задачи: приобретение системы знаний о разработке программ как одной из функций деятельности информатика, позволяющей принять соответствующие решения при решении задач в экономической отрасли; организация разработки информационной системы для широкого круга внутренних и внешних пользователей предприятия; формирование и представление информации, использования её при разработке приложений, удовлетворяющей требованиям различных пользователей программного обеспечения.

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 обучающимися 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, приходиться 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:

- Раздел 1. Технология создания алгоритмов программ
- Раздел 2. Программирование обработки текстовой информации
- Раздел 3. Сложные типы данных
- Раздел 4. Основы конструирования программ методом функциональной декомпозиции.

Модульное программирование

- Раздел 5. Разработка приложений работы с файлами и файловой структурой
- Раздел 6. Разработка приложений с использованием динамических структур данных
- Раздел 7. Рекурсивные структуры данных и рекурсивные алгоритмы обработки
- Раздел 8. Разработка консольных приложений с использованием стандартного модуля

conio.h

- Раздел 9. Разработка графических приложений с использованием стандартного графического модуля
- Раздел 10. Основы разработки приложений с использованием инструментальной среды разработки и отладки программ C++ Builder.

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

Цель проведения лабораторных занятий заключается в закреплении полученных теоретических знаний на лекциях и в процессе самостоятельного изучения студентами специальной литературы. Основной формой проведения лабораторных занятий является обсуждение наиболее проблемных и сложных ситуаций по отдельным темам, а также выполнение различных заданий с использованием компьютерных технологий. В обязанности преподавателя входят оказание методической помощи и консультирование студентов по соответствующим темам курса.

Примерные задания к лабораторным работам:

1. Разработать приложение, использующее строковые типы данных
2. Разработать приложение с использованием структур
3. Разработать приложение с использованием динамических переменных – использование указателей для работы с динамическими переменными; – использование динамической структуры данных – стек; – использование динамической структуры данных – список; использование динамической структуры данных – очередь
4. Разработать приложение с использованием динамических массивов
5. Разработать приложение с использованием файлового ввода-вывода
6. Разработать приложение в среде Builder: – разработать программ с использованием инструментария; – структура проекта; – использование компонент
7. Программирование с использованием компонент работы со строками – типы данных для работы со строками; – используемые компоненты для работы со строками; обработка событий.
8. Разработать приложение с использованием диалогов для выбора файлов и главного меню – компоненты диалога с пользователем; – приёмы программирования со структурами; – приёмы программирования с файлами.
9. Разработать приложение с использованием средств для отображения графической информации

10. Разработать приложение с использованием доступа к таблицам баз данных – Основы реляционных баз данных; – Создание таблиц при помощи DataBase Desktop; – Разработать приложения для работы с таблицей данных; – Обработка информации однотабличной базы данных; – Печать данных.

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 курсового проектирования/ работы

Курсовой проект/ работа проводится с целью формирования у обучающихся опыта комплексного решения конкретных задач профессиональной деятельности.

Курсовой проект/ работа позволяет обучающемуся:

- систематизировать и закрепить полученные теоретические знания и практические умения по профессиональным учебным дисциплинам и модулям в соответствии с требованиями к уровню подготовки, установленными программой учебной дисциплины, программой подготовки специалиста соответствующего уровня, квалификации;
- применить полученные знания, умения и практический опыт при решении комплексных задач, в соответствии с основными видами профессиональной деятельности по направлению/ специальности/ программе;
- углубить теоретические знания в соответствии с заданной темой;
- сформировать умения применять теоретические знания при решении нестандартных задач;
- приобрести опыт аналитической, расчётной, конструкторской работы и сформировать соответствующие умения;
- сформировать умения работы со специальной литературой, справочной, нормативной и правовой документацией и иными информационными источниками;
- сформировать умения формулировать логически обоснованные выводы, предложения и рекомендации по результатам выполнения работы;
- развить профессиональную письменную и устную речь обучающегося;
- развить системное мышление, творческую инициативу, самостоятельность, организованность и ответственность за принимаемые решения;
- сформировать навыки планомерной регулярной работы над решением поставленных задач.

Структура пояснительной записки курсовой работы / проекта

В соответствии с учебным планом студенты готовят курсовую работу. Цель выполнения курсовой работы: изучить методику разработки приложений с использованием современных инструментальных средств. Состав курсовой работы: Первая часть курсовой работы (электронная). В ней студент должен раскрыть практические знания о современных подходах разработки программ с использованием языка программирования C++. Представить исходный код самостоятельно разработанной программы, компилированный и отлаженный разработанный программный продукт на электронном носителе информации. Вторая часть (теоретическая). Приводится обзор нормативных актов по изучаемой теме, краткая теория и критический обзор литературных источников по проблеме исследования. Описать алгоритм, исходный код, методы программирования и библиотечные компоненты, использованные при разработке приложения. Привести описание работы приложения, результаты работы. При выполнении курсовой работы студент должен использовать язык программирования высокого уровня интегрированные среды разработки и отладки программ Visual C++ , Builder C++.

Основные разделы курсовой работы:

1. Титульный лист.
2. Утвержденное индивидуальное задание на курсовую работу.
3. Схема функциональной структуры программного приложения.
4. Описание автоматизируемых функций
5. Структурная схема иерархии форм приложения
6. Описание программного обеспечения
7. Программа и методика испытаний приложения
8. Основные результаты работы приложения.

Требования к оформлению пояснительной записки курсовой работы / проекта

В виду принадлежности курсового проекта к дисциплинам связанным с информационными технологиями и электронно-вычислительными машинами пояснительная записка должна быть оформлена при помощи любого программного инструмента и распечатана на листах формата А4 (210×297 мм), листы должны быть пронумерованы и сшиты. Поля листа должны составлять левое 25 мм, верхнее и нижнее 20 мм, правое 15 мм. Текст записки должен быть набран удобочитаемым шрифтом по размеру и начертанию соответствующий «Times New Roman» в 14 пт. Межстрочный интервал должен соответствовать полуторному. В записке также должен быть предусмотрен карман для помещения в него диска с работоспособным приложением и всеми исходными текстами программ. Допускается помещать на дискету архив в формате zip или rar.

Полный листинг программы должен включать в себя распечатку всех файлов программ, из которых состоит проект. Формы проекта должны быть распечатаны в двух видах: в виде формы и в виде тестового файла. Все файлы форм должны быть сгруппированы в следующей последовательности: сначала форма в процессе разработки, затем форма в текстовом виде и в завершении текст модуля связанный с формой. В записке фрагменты текстов программы, а также тексты распечаток модуля и формы должны быть выполнены шрифтом «Courier New» размером 10 пт., через одинарный интервал.

Титульный лист записки должен быть оформлен в соответствии с образцом, приведенным в приложении №1.

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 по выполнению контрольных работ (для обучающихся по заочной форме обучения).

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 – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

– зачет – это форма оценки знаний, полученных обучающимся в ходе изучения учебной дисциплины в целом или промежуточная (по окончании семестра) оценка знаний обучающимся по отдельным разделам дисциплины с аттестационной оценкой «зачтено» или «не зачтено».

– дифференцированный зачет – 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