

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

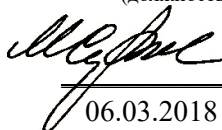
Кафедра №44

«УТВЕРЖДАЮ»

Руководитель направления

д.т.н., проф.

(должность, уч. степень, звание)


М.Б. Сергеев
06.03.2018

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования»
(Название дисциплины)


Код направления	09.03.01
Наименование на- правления	Информатика и вычислительная техника
Наименование на- правленности	Вычислительные машины, комплексы, системы и сети
Форма обучения	заочная

Санкт-Петербург 2018 г.

Лист согласования рабочей программы дисциплины

Программу составил(а)

доц., к.т.н., доц.



Л.Н. Бариков

06.03.2018

Программа одобрена на заседании кафедры № 44

06.03.2018, протокол № 6-17/18

Заведующий кафедрой № 44

д.т.н., проф.

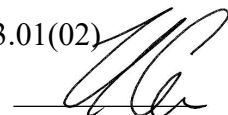


М.Б. Сергеев

06.03.2018

Ответственный за ОП 09.03.01(02)

доц., к.т.н., доц.

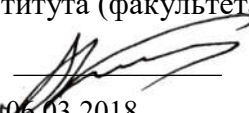


Н.В. Соловьев

06.03.2018

Заместитель директора института (факультета) № 4 по методической работе

доц., к.т.н., доц.



А.А. Ключарев

06.03.2018

Аннотация

Дисциплина «Основы программирования» входит в вариативную часть образовательной программы подготовки студентов по направлению «09.03.01 «Информатика и вычислительная техника» направленность «Вычислительные машины, комплексы, системы и сети». Дисциплина реализуется кафедрой №44

Дисциплина нацелена на формирование у выпускника общепрофессиональных компетенций:

ОПК-2 «способность осваивать методики использования программных средств для решения практических задач».

Содержание дисциплины охватывает круг вопросов, связанных с современным состоянием и тенденциями развития языков программирования и средств для разработки программ различного уровня сложности (принципы разработки, написания и отладки программ разной степени сложности, основы системного программирования, основы объектно-ориентированного подхода к программированию, принципы действия, основные функции и интерфейс современных инструментальных средств).

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 дисциплине

1.1. Цели преподавания дисциплины

Формирование у студентов представления о современном состоянии языков программирования и средствах для разработки программ различного уровня сложности, а также развитие практических навыков по разработке программ на языках программирования высокого уровня с использованием сред визуального объектно-ориентированного программирования.

1.2. Перечень планируемых результатов обучения дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины студент должен обладать следующими компетенциями:

ОПК-2 «способность осваивать методики использования программных средств для решения практических задач»:

знать - принципы разработки, написания и отладки программ разной степени сложности, основы системного программирования, основы объектно-ориентированного подхода к программированию, принципы действия, основные функции и интерфейс современных инструментальных средств;

уметь - для сформулированной задачи разработать алгоритм, написать и отладить программу на языке высокого уровня, работать с современными системами программирования, включая объектно-ориентированные;

владеть навыками - процедурного и объектно-ориентированного программирования, навыками разработки и отладки программ не менее, чем на одном алгоритмических процедурных языках программирования высокого уровня;

иметь опыт деятельности – в постановке и решении практических задач различной степени сложности.

2. Место дисциплины в структуре ОП

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

- Информатика
- Математическая логика и теория алгоритмов.

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

- Программирование на языках Ассемблера
- Технология программирования
- Человеко-машинный интерфейс
- Базы данных
- Системное программное обеспечение
- Операционные системы
- Открытые системы
- Цифровая обработка изображений
- Теория принятия решений
- Логическое программирование
- Технология разработки открытого программного обеспечения
- Мультимедиа производство,

а также используются при подготовке выпускной квалификационной работы бакалавра.

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
Общая трудоемкость дисциплины, ЗЕ/(час)	10/ 360	4/ 144	6/ 216
<i>Аудиторные занятия</i> , всего час., <i>В том числе</i>	32	20	12
лекции (Л), (час)	14	10	4
Практические/семинарские занятия (ПЗ), (час)	4		4
лабораторные работы (ЛР), (час)	14	10	4
курсовой проект (работа) (КП, КР), (час)	*		*
Экзамен, (час)	18	9	9
Самостоятельная работа , всего (час)	310	115	195
Вид промежуточной аттестации: зачет, экзамен, дифференцированный зачет (Зачет. Экз. Дифф. зач)	Экз., Экз.	Экз.	Экз.

* - часы , не входящие в аудиторную нагрузку

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. Методы разработки программных средств	2	-	-	-	20
Раздел 2. Основы языка программирования С++	2	-	-	-	30
Раздел 3. Структурное программирование	2	-	7	-	20
Раздел 4. Процедурное и модульное программирование	4	-	3	-	45
Итого в семестре:	10	-	10	-	115
Семестр 3					
Раздел 5. Динамические структуры данных	2	-	4	-	80
Раздел 6. Концепции объектно-ориентированного программирования	2	-	-	-	85

Выполнение курсовой работы	-	4	-	-	30
Итого в семестре:	4	4	4	-	195
Итого:	14	4	14	-	310

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	<p>Тема 1.1. Современные методы разработки программных средств. Развитие современных методов программирования. Структурное, процедурное, модульное и объектно-ориентированное программирование. Технологии нисходящего и восходящего проектирования программ. Сущность структурного программирования: разбиение на подзадачи, нисходящее проектирование, стандартные структуры управления. Достоинства и недостатки.</p> <p>Тема 1.2. Этапы разработки программных средств. Кодирование и документирование программ. Проектирование и тестирование программ. Основные этапы разработки программ. Этап постановки задачи. Этап разработки структур данных: определение входных и выходных данных, формы получения результатов. Использование технологии нисходящего проектирования на этапе проектирования программных средств. Этап структурного программирования. Этап тестирования и отладки.</p>
2	<p>Тема 2.1. Современные языки программирования. Понятие языка программирования. Этапы развития языков программирования. Современные тенденции в области языков программирования. Сравнение развития языков в представлении данных и способах реализации алгоритмов. Сравнительная характеристика языков программирования высокого уровня. Синтаксис и семантика. Способы описания синтаксиса: лингвистические формулы и синтаксические диаграммы.</p> <p>Тема 2.2. Базовые средства языка программирования C++. Структура языка программирования C++. Базовые элементы языка C++: алфавит, лексемы, выражения, комментарии. Основные (стандартные) типы данных языка C++. Типы арифметические целочисленные. Управляющие последовательности. Типы арифметические с плавающей точкой. Логический тип bool. Тип void. Идентификаторы языка. Область действия и область видимости идентификатора. Ключевые (зарезервированные) слова. Константы. Разделители. Спецификации классов памяти. Предложения языка: описания и операторы. Программа на языке C++: состав и структура.</p> <p>Тема 2.3. Понятие объекта. Объекты. Классификация объектов. Арифметические операции над объектами. Арифметические преобразования. Логические и поразрядные логические операции над объектами. Операции сдвига. Понятие об L-значении. Операции присваивания. Преобразования при присваивании. Операция запятая. Операции увеличения и уменьшения значения. Условная операция. Операция получения адреса объекта в основной памяти и операция получения содержимого объекта по адресу. Указатели. Указатели на объект, на функцию, на void. Операции с указателями. Ссылки. Операция приведения. Операция размер. Первичные операции. Понятие выражения. Порядок выполнения операций в выражении.</p>
3	Тема 3.1. Базовые конструкции структурного программирования.

	<p>Средства реализации линейных алгоритмов: оператор - выражение, составной оператор (блок), пустой оператор. Средства реализации разветвляющихся алгоритмов: условный оператор, оператор - переключатель, оператор перехода, оператор разрыва, оператор перехода на начало следующей итерации, оператор возврата в вызывающую функцию. Средства реализации циклических алгоритмов: операторы цикла с предусловием, с постусловием, с параметром. Реализация арифметических, итерационных и вложенных циклов. Реализация рекуррентных вычислений.</p> <p>Тема 3.2. Примеры разработки программ.</p> <p>Линейные программы. Вычисление результатов поразрядных логических операций над целыми числами. Разветвляющиеся программы – поиск экстремальных значений. Попадание точки в заданные области на плоскости. Циклические программы. Решение задач на заданное число повторений. Решение задач на получение результатов с заданной точностью. Получение таблиц значений алгебраических функций в заданном диапазоне изменения значения аргумента.</p> <p>Тема 3.3. Статические массивы.</p> <p>Статическое распределение памяти. Структурированные типы данных: одномерные и многомерные массивы. Решение типовых задач на массивы: поиск значений максимального отрицательного и минимального положительных элементов одномерного вещественного массива, перестановка элементов одномерного массива. Перестановка столбцов и строк матриц по заданным правилам.</p>
4	<p>Тема 4.1. Концепции процедурного программирования.</p> <p>Функции. Основные понятия. Принципы использования функций в программах. Объявление и определение функций. Список параметров и тип функции. Вызов функций на исполнение. Формальные и фактические параметры. Области действия описаний функций. Локальность и глобальность. Механизм передачи параметров. Глобальные переменные. Функция main(). Передача массивов в качестве параметров. Примеры. Реализация алгоритмов сортировки структур данных и поиска в этих структурах.</p> <p>Тема 4.2. Понятие рекурсии.</p> <p>Передача имен функций в качестве параметров. Понятие рекурсии. Рекурсивные определения и алгоритмы. Программирование рекурсивных алгоритмов: рекурсивные функции. Механизм рекурсивных вызовов. Перегрузка функций. Шаблоны функций.</p> <p>Тема 4.3. Концепции модульного программирования.</p> <p>Модули: назначение, структура, трансляция, тестирование. Особенности использования модулей. Модульные программы. Директива #include. Директива #define. Построение многомодульных программ средствами языка программирования C++.</p>
5	<p>Тема 5.1. Динамические структуры данных.</p> <p>Указатели и ссылки. Динамические массивы. Динамические структуры: списки. Виды списков: односвязные и двусвязные списки, линейные и циклические списки. Динамические структуры: стеки. Динамические структуры: очереди. Динамические структуры: деки.</p> <p>Тема 5.2. Деревья.</p> <p>Динамические структуры: деревья. Виды деревьев и способы их реализации. Бинарное дерево как рекурсивная структура данных. Рекурсивные процедуры обхода дерева: инфиксная форма, префиксная форма, постфиксная форма. Особенности использования рекурсии при построении дерева.</p>
6	<p>Тема 6.1. Классы.</p> <p>Основные принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм. Классы в языке C++. Разработка классов. Составляющие классов: поля, методы. Экземпляры класса (объекты). Время их жизни и видимость. Конструкторы. Свойства конструкторов. Статические элементы класса (статические поля и статические методы). Дружественные функции и дружест-</p>

<p>венные классы. Деструкторы. Тема 6.2 Перегрузка операций. Перегрузка операций с объектами конкретных классов. Перегрузка унарных операций. Перегрузка бинарных операций. Перегрузка операции присваивания. Перегрузка операции приведения типа. Перегрузка операции вызова функции. Перегрузка операции индексирования. Тема 6.3. Наследование Понятие наследования. Простое наследование. Виртуальные методы. Механизм позднего связывания. Множественное наследование. Шаблоны классов. Создание шаблонов классов. Использование шаблонов классов. Тема 6.4 Поточковые классы. Стандартные потоки. Методы обмена с потоками (операция извлечения из потока и операция включения в поток). Файловые потоки. Строковые потоки. Тема 6.5 Строки в стиле языка C++. Класс string. Конструкторы строк. Присваивание строк. Операции со строками. Функции добавления частей строк. Функции преобразования строк. Функции поиска подстрок. Функции сравнения частей строк. Функции получения характеристик строк. Тема 6.6. Основы доказательства правильности программ Теорема структуры и структурное программирование. Анализ программ. Утверждения о программах. Корректность программ. Правила вывода для основных структур программирования. Инвариантные утверждения. Утверждения о массивах. Способы верификации программ. Жизненный цикл программы.</p>

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
Семестр 3				
1	Выполнение курсовой работы	решение типовых задач	4	6
Всего:			4	

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 2			
1	Следование	2	1
2	Операторы ветвления	2	2
3	Операторы цикла: заданное число повторений	3	3
4	Статические одномерные массивы	3	3
Семестр 3			
5	Динамические массивы	2	5
6	Линейные списки	2	6
Всего:		14	

4.5. Курсовое проектирование (работа)

Цель курсовой работы: закрепление теоретических и практических знаний, полученных во время лекционных, лабораторных, практических и самостоятельных занятий.

За время курсового проектирования необходимо выполнить все этапы решения задачи с помощью ЭВМ от постановки задачи до выпуска документации на разработанное программное средство. Курсовой проект заканчивается оформлением пояснительной записки и устной защитой работы с показом работы программного средства на ЭВМ.

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

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 2, час	Семестр 3, час
Самостоятельная работа, всего	310	115	195
изучение теоретического материала дисциплины	223	80	143
курсовое проектирование	30	-	30
подготовка отчетов по лабораторным работам	49	31	18
контрольные работы заочников	8	4	4

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)
	Бариков Л.Н. Основы программирования [Электронный ресурс]: учебное пособие / Л.Н. Бариков, СПб. ГУАП - Электрон. текстовые дан. – СПб.: ГУАП, 2016. - 138с. Режим доступа: http://lib.aanet.ru	
004.4 П12	Павловская Т.А. С/С++. Программирование на языке высокого уровня [Текст]: учебник / Т.А. Павловская. - СПб.: ПИТЕР, 2003. - 459с.	49

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)
	Герман О. Программирование на Java и C# для студентов [Текст : Электронный ресурс]: монография / О. Герман, Ю. Герман. - СПб.: БХВ - Петербург, 2005. - 511с. Режим доступа: http://lib.aanet.ru	
	Дейл Н., Уимз Ч., Хедингтон М. Программирование на	

	C++. [Электронный ресурс] М. - ДМК Пресс, 2007. - 672с. Режим доступа: http://e.lanbook.com/view/book/1219	
	Санников, Е.В. Курс практического программирования в Delphi. Объектно-ориентированное программирование [Электронный ресурс]: - Электрон. дан. - М.: СОЛОН-Пресс, 2013. - 188с. - Режим доступа: http://e.lanbook.com/books/element.php?p11_id=64955	
	Игошин В.И. Теория алгоритмов: Учебное пособие / В.И. Игошин. - М.: ИНФРА-М, 2012. - 318с. Режим доступа: http://znanium.com/bookread.php?book=241722	

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
	Не предусмотрено

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	
1	Borland C++ 3.1	
2	Geany	
3	3	Visual S

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	Компьютерный класс	Г а.22-03, 22-08

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.
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-2 «способность осваивать методики использования программных средств для решения практических задач»	
1	Математика. Аналитическая геометрия и линейная алгебра
1	Математика. Математический анализ
1	Компьютерный практикум
1	Физика
2	Математика. Математический анализ
2	Физика
2	Основы программирования
2	Учебная практика
3	Электротехника
3	Основы программирования
5	Программирование на языках Ассемблера
5	Численные методы и вариационное исчисление
5	Экология
5	Теория принятия решений
5	Электроника
5	Теория автоматов
6	Схемотехника
6	Моделирование
6	Компьютерная графика
6	Операционные системы
7	Системы виртуальной реальности
7	Организация ЭВМ и вычислительных систем
7	Логическое программирование
7	Моделирование
7	Человеко-машинный интерфейс
7	Микропроцессорные системы
7	Интерактивная компьютерная графика

8	Открытые системы
8	Теория оптимального управления
8	Технология разработки открытого программного обеспечения
8	Системы искусственного интеллекта
8	Цифровые системы автоматизации и управления
8	Системное программное обеспечение
8	Микропроцессорные системы
9	Корпоративные сети со службой каталога
9	Системное программное обеспечение
9	Системы искусственного интеллекта
9	Проектирование систем обработки и передачи информации
9	Цифровая обработка изображений
9	Основы построения экспертных систем
9	Цифровые системы автоматизации и управления
9	Распределенные вычисления на сетях
9	Интерфейсы периферийных устройств
10	Разработка Интернет-приложений
10	Введение в ортогональные преобразования информации
10	Теория вычислительных процессов
10	Проектирование систем обработки и передачи информации
10	Теория надежности ВС и ПО
10	Интерфейсы периферийных устройств
10	Администрирование вычислительных сетей на базе UNIX

10.3. В качестве критериев оценки уровня сформированности (освоения) у обучающихся компетенций применяется шкала модульно–рейтинговой системы университета. В таблице 15 представлены 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;

		<ul style="list-style-type: none"> - 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
	Семестр 2
1	Базовые элементы языка C++: алфавит, лексемы, комментарии.
2	Концепция данных в языке C++.
3	Основные (стандартные) типы данных языка C++.
4	Типы арифметические целочисленные. Управляющие последовательности.
5	Типы арифметические с плавающей точкой.
6	Логический тип bool. Тип void.
7	Предложения языка C++: описания и операторы.
8	Программа на языке C++: состав и структура.
9	Идентификаторы языка. Область действия и область видимости идентификатора.
10	Ключевые (зарезервированные) слова.
11	Константы. Разделители.
12	Спецификации классов памяти.
13	Объекты. Классификация объектов.
14	Арифметические операции над объектами. Арифметические преобразования.
15	Логические и поразрядные логические операции над объектами.
16	Операции сдвига. Понятие об L-значении.
17	Операции присваивания. Преобразования при присваивании.
18	Операция запятой. Операции увеличения и уменьшения значения.
19	Условная операция. Операция получения адреса объекта в основной памяти и операция получения содержимого объекта по адресу.
20	Указатели. Указатели на объект, на функцию, на void. Операции с указателями.
21	Ссылки.
22	Операция приведения. Операция размер. Первичные операции.
23	Понятие выражения. Порядок выполнения операций в выражении.
24	Средства реализации линейных алгоритмов: оператор - выражение, составной оператор (блок), пустой оператор.
25	Средства реализации разветвляющихся алгоритмов: условный оператор, оператор

26	- переключатель, оператор перехода, оператор разрыва, оператор перехода на начало следующей итерации, оператор возврата в вызывающую функцию. Средства реализации циклических алгоритмов: операторы цикла с предусловием, с постусловием, с параметром.
27	Реализация арифметических, итерационных и вложенных циклов.
28	Реализация рекуррентных вычислений.
29	Директивы препроцессора: директива #include.
30	Директивы препроцессора: директива #define.
Семестр 3	
1	Структурированные типы данных: массивы одномерные и многомерные.
2	Строки.
3	Структурированные типы данных: структуры (struct).
4	Структурированные типы данных: перечисления (enum).
5	Структурированные типы данных: объединения (union).
6	Переименование типов (typedef).
7	Статическое распределение памяти.
8	Функции. Основные понятия. Принципы использования функций в программах.
9	Объявление и определение функций.
10	Список параметров и тип функции.
11	Вызов функций на исполнение. Формальные и фактические параметры.
12	Области действия описаний функций. Локальность и глобальность.
13	Механизм передачи параметров.
14	Глобальные переменные.
15	Функция main().
16	Передача массивов в качестве параметров. Примеры.
17	Реализация алгоритмов сортировки структур данных и поиска в этих структурах.
18	Передача имен функций в качестве параметров.
19	Понятие рекурсии. Рекурсивные определения и алгоритмы.
20	Программирование рекурсивных алгоритмов: рекурсивные функции. Механизм рекурсивных вызовов.
21	Перегрузка функций.
22	Шаблоны функций.
23	Динамические структуры: списки. Виды списков: односвязные и двусвязные списки, линейные и циклические списки.
24	Динамические структуры: стеки.
25	Динамические структуры: очереди.
26	Динамические структуры: деки.
27	Динамические структуры: деревья. Виды деревьев и способы их реализации.
28	Классы в языке C++.
29	Разработка классов. Составляющие классов: поля, методы.
30	Экземпляры класса (объекты). Время их жизни и видимость.
31	Конструкторы. Свойства конструкторов.
32	Статические элементы класса (статические поля и статические методы).
33	Наследование. Простое наследование.
34	Виртуальные методы. Механизм позднего связывания.
35	Критерии качества программы.
36	Жизненный цикл программы.

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
	Решение задач линейного программирования:
1	Транспортная задача.
2	Задача о назначениях.
3	Задача об оптимальном ассортименте.
4	Задача о диете.
5	Задача о максимальном потоке.
	Решение задач теории расписаний:
6	Задача одного исполнителя на минимум.
7	Задача одного исполнителя на минисумму.
8	Задача о нескольких исполнителях.
	Решение прикладных задач теории графов:
9	Задача о дорожной сети.
10	Задача о телефонной связи.
11	Задача строительной трассировки.
12	Задача электронной трассировки
13	Задача размещения.
	Реализация алгоритмов теории графов:
14	Поиск в глубину графа.
15	Поиск в ширину графа.
16	Кратчайшие пути в графе.
17	Эйлеровы пути.
18	Гамильтоновы циклы.
	Разработка программ создания и использования баз данных в различных предметных областях:
19	База данных телефонной сети.
20	База данных библиотеки.
21	База данных магазина грамзаписей.
	Разработка программ разного назначения:
22	Игра «Морской бой».
23	Игра «Sudoku».
24	Игра «Змейка»
25	Игра «Поле чудес».

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 программы 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 – получение студентами необходимых знаний, умений и навыков о современном состоянии языков программирования и средствах для разработки программ различного уровня сложности, а также развитие практических навыков по разработке программ на языках программирования высокого уровня с использованием сред визуального объектно-ориентированного 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 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;
- умение 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 материала по рассматриваемой теме;
- демонстрация примеров составления схем алгоритмов решения конкретных задач;
- ответы на возникающие вопросы по теме лекции;
- выдача раздаточного материала с примерами программ по теме лекции и обсуждение их особенностей.

Структура лекционного материала приведена в издании:

Бариков Л.Н. Основы программирования [Текст]: учебное пособие / Л.Н.Бариков. – СПб.: ГУАП, 2016. - 138с.

Методические указания для обучающихся по прохождению практических занятий

Практическое занятие является одной из основных форм организации учебного процесса. Оно заключается в выполнении обучающимися под руководством преподавателя комплекса учебных заданий с целью усвоения научно-теоретических основ учебной дисциплины, приобретения умений и навыков, опыта творческой деятельности.

Целью практического занятия является привитие обучающимся умений и навыков практической деятельности по изучаемой дисциплине.

Планируемые результаты при освоении обучающимися практических занятий:

- закрепление, углубление, расширение и детализация знаний при решении конкретных задач;
- развитие познавательных способностей, самостоятельности мышления, творческой активности;
- овладение новыми методами и методиками изучения конкретной учебной дисциплины;
- выработка способности логического осмысления полученных знаний для выполнения заданий;
- обеспечение рационального сочетания коллективной и индивидуальной форм обучения.

Функции практических занятий:

- познавательная;
- развивающая;
- воспитательная.

По характеру выполняемых обучающимся заданий по практическим занятиям подразделяются на:

- ознакомительные, проводимые с целью закрепления и конкретизации изученного теоретического материала;
- аналитические, ставящие своей целью получение новой информации на основе формализованных методов;
- творческие, связанные с получением новой информации путем самостоятельно выбранных подходов к решению задач.

Формы организации практических занятий:

- интерактивная форма (обсуждение вариантов схем алгоритмов для решения конкретных практических задач);
- не интерактивная форма (выполнение упражнений, решение типовых задач).

Методика проведения практического занятия может быть различной, при этом важно достижение общей цели дисциплины.

Требования к проведению практических занятий

Во время проведения практических занятий необходимо представить преподавателю все этапы решения задачи по курсовому проектированию от постановки задачи и выбора средств разработки до выпуска документации на разработанный программный продукт.

Методические указания для обучающихся при выполнении 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 аппаратно-программным обеспечением.

Задание и требования к проведению лабораторных работ

По каждой лабораторной работе обучающийся получает вариант индивидуального задания в соответствии с его номером в списке группы. Перед проведением лабораторной работы обучающемуся следует внимательно ознакомиться с методическими указаниями по её выполнению. В соответствии с заданием обучающийся должен выполнить постановку задачи: сформулировать условие, определить входные и выходные данные, разработать математическую модель. После этого он должен построить схему алгоритма решения задачи и защитить её у преподавателя. Это является допуском к реализации алгоритма на компьютере. После отладки программы обучающийся должен продемонстрировать преподавателю работу программы на полном наборе тестов. Лабораторная работа завершается оформлением и защитой отчета по лабораторной работе.

Структура и форма отчета о лабораторной работе

Отчет о лабораторной работе должен включать в себя: титульный лист, формулировку задания, математическая модель, схема алгоритма решения задачи, текст программы, контрольные (тестовые) примеры.

Требования к оформлению отчета о лабораторной работе

По каждой лабораторной работе выполняется отдельный отчет. Титульный лист оформляется в соответствии с шаблоном (образцом) приведенным на сайте ГУАП (www.guar.ru) в разделе «Сектор нормативной документации». Текстовые и графические материалы оформляются в соответствии с действующими ГОСТами и требованиями, приведенными на сайте ГУАП (www.guar.ru) в разделе «Сектор нормативной документации».

Методические указания к выполнению лабораторных работ включены в состав следующих изданий:

Бариков Л.Н. Основы программирования: практикум. [Электронный ресурс]: учебное пособие / Л.Н. Бариков. - СПб.: ГУАП, кафедральное издание, - 2016.

Методические указания для обучающихся по прохождению курсового проектирования

Курсовое проектирование является завершающим и одним из важнейших этапов подготовки специалистов по дисциплине “Основы программирования”. Одной из основных целей этого этапа является обучение студентов оформлению необходимой программной документации на разработанное программное средство в соответствии с требованиями ЕСПД.

Курсовой проект 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 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.

За время курсового проектирования необходимо выполнить все этапы решения задачи с помощью ЭВМ от постановки задачи до выпуска документации на разработанное программное средство. Курсовой проект заканчивается оформлением пояснительной записки и устной защитой работы с показом работы программного средства на ЭВМ.

Структура пояснительной записки курсовой работы / проекта

В состав пояснительной записки должны входить:

- титульный лист;
- содержание пояснительной записки;
- введение;
- четыре программных документа, оформленных по Стандартам ЕСПД:
 - Спецификация.
 - Текст программы.
 - Описание программы.
 - Описание применения.
- заключение;
- литература.

Руководитель курсового проектирования может в каждом конкретном случае потребовать дополнительного оформления еще ряда документов.

Требования к оформлению пояснительной записки курсовой работы / проекта

Документ **Пояснительная записка** содержит укрупненные схемы алгоритмов, общие описания алгоритмов и функционирования программы, а также обоснование принятых технических и технико-экономических решений. Требования к содержанию и оформлению **Пояснительной записки** устанавливает ГОСТ 19.404-79.

Требования к содержанию и оформлению документа **Спецификация** устанавливает ГОСТ 19.202-78. Этот ГОСТ определяет программный документ **Спецификация** как состав программы и документации на неё.

Требования к содержанию и оформлению документа **Текст программы** устанавливает ГОСТ 19.401-78. Этот ГОСТ определяет программный документ **Текст программы** как запись программы с необходимыми комментариями.

Требования к содержанию и оформлению документа **Описание программы** устанавливает ГОСТ 19.402-78. Этот ГОСТ определяет программный документ **Описание программы** как совокупность сведений о логической структуре и функционировании программы.

Документ **Описание применения** содержит сведения о назначении программы, области применения, применяемых методах, класса решаемых задач, ограничениях для применения, минимальной конфигурации технических средств, операционной среде. Требования к содержанию и оформлению этого программного документа устанавливает ГОСТ 19.502-78.

Методические указания к выполнению курсовой работы / проекта:

Бариков Л.Н. Документирование программного обеспечения. [Текст]: Методические указания к выполнению курсовой работы / Л.Н.Бариков. – СПб.: ГУАП, 2013. – 24 с.

Методические указания для обучающихся при выполнении самостоятельной работы

В ходе выполнения самостоятельной работы обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

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-методические материалы по дисциплине.

Для обучающихся по заочной форме обучения читаются установочные лекции. Полный лекционный курс они изучают самостоятельно.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

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».

Система оценок при проведении промежуточной аттестации осуществляется в соответствии с требованиями Положений «О текущем контроле успеваемости и промежуточной аттестации студентов ГУАП, обучающихся по программе 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