

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ряполов Петр Алексеевич

Должность: декан ЕНФ

Дата подписания: 24.10.2023 11:26:48

Уникальный программный ключ:

efd3ecd183f7649d0e3a73c230c6662946c7c99039b2b268921fde408c1fb6

Аннотация к рабочей программе дисциплины «Программирование на языках низкого уровня»

Цели и задачи дисциплины

Целью изучения дисциплины является формирование у обучаемых знаний, умений и навыков, необходимых для успешного овладения общекультурными и профессиональными компетенциями в области автоматизации и управления; ознакомление студентов с базовыми понятиями, основными принципами построения, алгоритмами и методиками анализа и синтеза систем компьютерного управления в мехатронике.

Задачи дисциплины:

- получение обучающимися представлений о различных средствах программирования и проектирования и их применения для задач управления в мехатронике;
- изучение обучающимися методов математического описания компьютерных систем управления, методов анализа и синтеза цифровых систем управления;
- освоение обучающимися современных подходов к формированию программного обеспечения систем управления движением в реальном времени.

Компетенции, формируемые в результате освоения дисциплины

- ОПК-3 владение современными информационными технологиями, готовностью применять современные средства автоматизированного проектирования и машинной графики при проектировании систем и их отдельных модулей, а также для подготовки конструкторско-технологической документации, соблюдать основные требования информационной безопасности.
- ОПК-6 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.
- ПК-2 способность разрабатывать программное обеспечение, необходимое для обработки информации и управления в мехатронных и робототехнических системах, а также для их проектирования.

Основные дидактические единицы (разделы)

Введение.

Способы представления и форматы данных.

Режимы работы процессора.

Пользовательские и системные регистры.

Регистры общего назначения.

Сегментные и управляющие регистры.

Организация памяти.

Способы адресации.

Синтаксис языка Ассемблер.
Структура программы.
Сегментация.
Директивы описания сегментов.
Модели памяти.
Псевдокоманды (директивы) определения данных.
Структура команды.
Виды операндов.
Команды пересылки для разных операндов.
Команды конвертирования форматов данных.
Команды двоичной и десятичной арифметики.
Команды сравнения и переходов.
Логические и сдвиговые операции.
Операции над битами и байтами.
Ввод с клавиатуры.
Работа с устройствами.
Работа с файлами.
Описание и инициализация массивов.
Макроопределения.
Организация и размещение процедур.
Связь Ассемблера с языками высокого уровня.
Разработка программ и использование отладчика.

МИНОБРНАУКИ РОССИИ

Юго-Западный государственный университет

«УТВЕРЖДАЮ»

Декан естественно-научного
факультета

П.А. Ряполов

« 31 » 08 2018 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Программирование на языках низкого уровня

(наименование дисциплины)

ОПОП ВО 15.03.06 Мехатроника и робототехника

шифр и наименование направления подготовки (специальности)

направленность (профиль) «Сервисная робототехника»

наименование направленности (профиля, специализации)

форма обучения очная

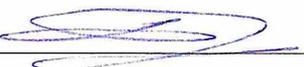
(очная, очно-заочная, заочная)

Курск-2018

Рабочая программа дисциплины составлена в соответствии с ФГОС ВО – бакалавриат по направлению подготовки (специальности) 15.03.06 Мехатроника и робототехника на основании учебного плана ОПОП ВО 15.03.06 Мехатроника и робототехника, направленность (профиль) «Сервисная робототехника», одобренного Ученым советом университета (протокол № 9 «26» марта 2018 г.).

Рабочая программа дисциплины обсуждена и рекомендована к реализации в образовательном процессе для обучения студентов по ОПОП ВО 15.03.06 Мехатроника и робототехника, направленность (профиль) «Сервисная робототехника» на заседании кафедры механики, мехатроники и робототехники № 1 «31» августа 2018.
(наименование кафедры, дата, номер протокола)

Зав. кафедрой _____  С.Ф. Яцун

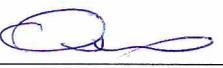
Разработчик программы
к.т.н., доцент _____  П.А. Безмен
(ученая степень и ученое звание, Ф.И.О.)

Согласовано: на заседании кафедры механики, мехатроники и робототехники
№ 1 «31» августа 2018.

Зав. кафедрой _____  С.Ф. Яцун

/Директор научной библиотеки _____  В.Г. Макаровская

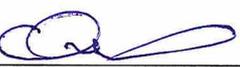
Рабочая программа дисциплины пересмотрена, обсуждена и рекомендована к реализации в образовательном процессе на основании учебного плана ОПОП ВО 15.03.06 Мехатроника и робототехника, направленность (профиль) «Сервисная робототехника», одобренного Ученым советом университета, на заседании кафедры механики, мехатроники и робототехники № 1 «29» августа 2019.
(наименование кафедры, дата, номер протокола)

Зав. кафедрой _____ 

Рабочая программа дисциплины пересмотрена, обсуждена и рекомендована к реализации в образовательном процессе на основании учебного плана ОПОП ВО 15.03.06 Мехатроника и робототехника, направленность (профиль) «Сервисная робототехника», одобренного Ученым советом университета, на заседании кафедры механики, мехатроники и робототехники № 1 «28» августа 2020.
(наименование кафедры, дата, номер протокола)

Зав. кафедрой _____ 

Рабочая программа дисциплины пересмотрена, обсуждена и рекомендована к реализации в образовательном процессе на основании учебного плана ОПОП ВО 15.03.06 Мехатроника и робототехника, направленность (профиль) «Сервисная робототехника», одобренного Ученым советом университета, на заседании кафедры Механика, мехатроники и робототехники № 1 «31» 08 2021.
(наименование кафедры, дата, номер протокола)

Зав. кафедрой _____ 

1. Планируемые результаты обучения, соотнесенные с планируемыми результатами освоения ОП

1.1 1 . Цель преподавания дисциплины

Целью изучения дисциплины «Программирование на языках низкого уровня» является формирование у обучаемых знаний, умений и навыков, необходимых для успешного овладения общекультурными и профессиональными компетенциями в области автоматизации и управления; ознакомление студентов с базовыми понятиями, основными принципами построения, алгоритмами и методиками анализа и синтеза систем компьютерного управления в мехатронике.

1.2 .Задачи преподавания дисциплины

- получение представлений о различных средствах программирования и проектирования и их применения для задач управления в мехатронике;
- изучение методов математического описания компьютерных систем управления, методов анализа и синтеза цифровых систем управления;
- освоение современных подходов к формированию программного обеспечения систем управления движением в реальном времени.

В результате изучения дисциплины студент должен:

- **знать:** различные парадигмы разработки программных продуктов в историческом контексте; методологию объектно-ориентированного программирования.
- **уметь:** разрабатывать компьютерные модели реальных и концептуальных систем на основе парадигмы компонентно-ориентированного программирования.
- **владеть:** навыками работы с современными аппаратными и программными средствами анализа, проектирования и разработки систем управления.

1.3 Компетенции, формируемые в результате освоения дисциплины

Согласно ФГОС и «Матрице распределения компетенций» изучение дисциплины направлено на формирование следующих компетенций:

ОПК-3 – владение современными информационными технологиями, готовностью применять современные средства автоматизированного проектирования и машинной графики при проектировании систем и их отдельных модулей, а также для подготовки конструкторско-технологической документации, соблюдать основные требования информационной безопасности;

ОПК-6 – способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с

применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;

ПК-2 – способность разрабатывать программное обеспечение, необходимое для обработки информации и управления в мехатронных и робототехнических системах, а также для их проектирования.

В результате изучения дисциплины обучающийся должен обладать следующими знаниями, умениями и навыками:

знать:

- современные информационные технологии;
- методы решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.

уметь:

- разрабатывать программное обеспечение, необходимое для обработки информации и управления в мехатронных и робототехнических системах, а также для их проектирования.

владеть:

- современными информационными технологиями;
- способностью осуществить подготовку документации на программный продукт.

приобрести опыт деятельности: в осуществлении подготовки документации на программные продукты (ОПК-3); в решении стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (ОПК-6); в разработке программного обеспечения, необходимого для обработки информации и управления в мехатронных и робототехнических системах, а также для их проектирования (ПК-2).

2 Место дисциплины в структуре образовательной программы

«Программирование на языках низкого уровня» представляет собой дисциплину с индексом Б1.В.ДВ.04.01 цикла «Дисциплины (модули)» вариативной части учебного плана направления подготовки 15.03.06 – Мехатроника и робототехника.

Изучается на 2 курсе в 3 семестре.

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

Общая трудоёмкость (объём) дисциплины составляет 5 зачетных единиц (з.е.), 180 часов.

Таблица 3.1 – Объем дисциплины по видам учебных занятий

Виды учебной работы	Всего, часов
Общая трудоёмкость дисциплины	180
Контактная работа обучающихся с преподавателем по видам учебных занятий (всего)	73,15
в том числе:	
лекции	36
лабораторные занятия	36
практические занятия	0
Самостоятельная работа обучающихся (всего)	70,85
Контроль (подготовка к экзамену)	36
Контактная работа по промежуточной аттестации (всего АттКР)	1,15
в том числе:	
зачет	не предусмотрен
зачет с оценкой	не предусмотрен
курсовая работа (проект)	не предусмотрена
экзамен (включая консультацию перед экзаменом)	1,15

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

4.1. Содержание дисциплины

Таблица 4.1.1 – Содержание дисциплины, структурированное по темам (разделам)

№ п/п	Раздел (тема) дисциплины	Содержание
1	2	3
1	Введение	Ведущая роль ЭВМ в реализации цифрового управления мехатронными системами. Программное обеспечение систем компьютерного управления. Влияние иерархической структуры мехатронной системы на структуру ее программного обеспечения.
2	Принципы объектно-ориентированного программирования (ООП)	Классификация задач и иерархическая структура программного обеспечения компьютерного управления. Основные принципы ООП. Объект, класс, методы, события.
3	Проектирование программ	Последовательность проектирования программ. Оценка эффективности программ, отладка и тестирование программ. Документирование программ.
4	Классы	Структуры, объединения и классы. Методы, свойства, указатели, ссылки. Функции с переменным числом параметров, указатели на функции, перегрузка функций, рекурсивные функции. Абстрактные, статические, динамические, виртуальные и перегружаемые методы. Классы как абстрактные типы. Конструкторы и деструкторы. Компонентные методы. Перегрузка операций.
5	Наследование	Наследование классов. Множественное наследование, виртуальные базовые классы. Классы и шаблоны.
6	События	Обработка событий. События мыши и клавиатуры. Обработка исключительных ситуаций.
7	Организация функций ввода-вывода данных	Разработка консольных приложений и приложений с графическим интерфейсом пользователя. Работа с файлами.

Таблица 4.1.2 - Содержание учебной дисциплины и его методическое обеспечение

№ п/п	Раздел, темы дисциплины	Виды деятельности			Учебно-методические материалы	Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)	Компетенции
		лек. час	№ лаб.	№ пр.			
1	2	3	4	5	6	7	8
1	Введение	4	1-2	—	У-1, У-3, МУ-1, МУ-2	КО, ЛР-1-2 (1-4 недели)	ОПК-3, ОПК-6, ПК-2
2	Принципы объектно - ориентированного программирования (ООП)	4	3-4	—	У-1, МУ-1, МУ-2	КО, ЛР-3-4 (5-7 недели)	ОПК-3, ОПК-6, ПК-2
3	Проектирование программ	4	5-6	—	У-1, У-4, МУ-1, МУ-2	КО, ЛР-5-6 (8-10 недели)	ОПК-3, ОПК-6, ПК-2
4	Классы	10	7	—	У-1, МУ-1, МУ-2	КО, ЛР-7 (11-12 недели)	ОПК-3, ОПК-6, ПК-2
5	Наследование	8	8	—	У-1, У-2, МУ-1, МУ-2	КО, ЛР-8 (13-14 недели)	ОПК-3, ОПК-6, ПК-2
6	События	4	9	—	У-1, МУ-1, МУ-2	КО, ЛР-9 (15-16 недели)	ОПК-3, ОПК-6, ПК-2
7	Организация функций ввода-вывода данных	2	10	—	У-1, У-2, МУ-1, МУ-2	КО, ЛР-10 (17-18 недели)	ОПК-3, ОПК-6, ПК-2

Примечание: КО – контрольный опрос, ЛР – лабораторная работа, ПР – практическая работа

4.2 Лабораторные и (или) практические занятия

Таблица 4.2.1 - Лабораторные работы

№	Наименование лабораторной работы	Объём, час.
1	Вещественные типы данных	2
2	Циклы	2
3	Одномерные массивы	4
4	Сортировка массива	4
5	Структуры	4
6	Строки и файлы	6
7	Приложение ОС Windows с выводом в окно текста и фигур	2
8	Средства ввода/вывода в ОС Windows	2
9	Преобразование растровых изображений	4
10	Построение графика функции	6
	Итого:	36

4.3 Самостоятельная работа студентов (СРС)

Выполняется в соответствии с рабочей программой дисциплины. Задания выдаются в ходе изучения дисциплины. Задачами работы являются: систематизация, закрепление и развитие знаний, полученных в ходе аудиторных занятий; стимулирование более глубокого и систематического изучения дисциплины в течение семестра; развитие умения самостоятельно работать с учебной и специальной литературой.

Таблица 4.3 - Самостоятельная работа студентов

№	Наименование раздела дисциплины	Срок выполнения	Время, затрачиваемое на выполнение СРС, час.
1	Введение	1 неделя	10
2	Принципы объектно-ориентированного программирования (ООП)	2-4 недели	10
3	Проектирование программ	5-7 недели	10
4	Классы	8-10 недели	10
5	Наследование	11-13 недели	10
6	События	14-15 недели	10
7	Организация функций ввода-вывода данных	16-17 недели	10,85
Итого:			70,85

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

Студенты могут при самостоятельном изучении отдельных тем и вопросов дисциплин пользоваться учебно-наглядными пособиями, учебным оборудованием и методическими разработками кафедры в рабочее время, установленное Правилами внутреннего распорядка работников.

Учебно-методическое обеспечение для самостоятельной работы обучающихся по данной дисциплине организуется:

библиотекой университета:

- библиотечный фонд укомплектован учебной, методической, научной, периодической, справочной и художественной литературой в соответствии с УП и данной РПД;

- имеется доступ к основным информационным образовательным ресурсам, информационной базе данных, в том числе библиографической, возможностью выхода в Интернет

кафедрой:

- путем обеспечения доступности всего необходимого учебно-методического и справочного материала;
- путем предоставления сведений о наличии учебно-методической литературы, современных программных средств;
- путем разработки:
 - методических рекомендаций, пособий по организации самостоятельной работы студентов;
 - заданий для самостоятельной работы студентов;
 - тем рефератов и докладов;
 - тем курсовых работ и проектов и методические рекомендации по их выполнению;
 - вопросов к экзаменам и зачетам;
 - методических указаний к выполнению лабораторных и практических работ и т.д.

типографией университета:

- помощь авторам в подготовке и издании научной, учебной и методической литературы;
- удовлетворение потребности в тиражировании научной, учебной и методической литературы.

6. Образовательные технологии. Технологии использования воспитательного потенциала дисциплины

Реализация компетентного подхода предусматривает широкое использование в образовательном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования общепрофессиональных компетенций обучающихся.

В соответствии с требованиями ФГОС по направлению подготовки 15.03.06 Мехатроника и робототехника и Приказом Министерства образования и науки РФ от 19 декабря 2013 г. № 1367 реализация компетентного подхода предусматривает широкое использование в образовательном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков студентов. В рамках изучения дисциплины предусмотрено проведение лекционных и практических занятий в интерактивной форме - разборов конкретных ситуаций, компьютерных симуляций, а также предусмотрены встречи с представителями российских и зарубежных компаний, мастер-классы экспертов и специалистов в области мехатроники и робототехники (ОАО «Авиавтоматика им. В.В. Тарасова», НИЦ (г. Курск) ФГУП «18 ЦНИИ» МО РФ, Барс Плюс, ОАО «Курскхелп.ру» и др).

Удельный вес занятий, проводимых в интерактивных формах, составляет 33% аудиторных занятий согласно УП.

Таблица 6.1 – Интерактивные образовательные технологии, используемые при проведении аудиторных занятий

№	Наименование раздела (лекции, практического или лабораторного занятия)	Используемые интерактивные образовательные технологии	Объём, час.
1	2	3	4
1	Программное обеспечение систем компьютерного управления (лекция)	Открытая лекция (мастер-класс)	2
2	Вещественные типы данных (лабораторная работа)	Компьютерная симуляция (разбор конкретных ситуаций)	1
3	Циклы (лабораторная работа)	Компьютерная симуляция (разбор конкретных ситуаций)	1
4	Одномерные массивы (лабораторная работа)	Компьютерная симуляция (разбор конкретных ситуаций)	2
5	Сортировка массива (лабораторная работа)	Компьютерная симуляция (разбор конкретных ситуаций)	2
6	Структуры (лабораторная работа)	Компьютерная симуляция (разбор конкретных ситуаций)	2
7	Строки и файлы (лабораторная работа)	Компьютерная симуляция (разбор конкретных ситуаций)	2
8	Приложение ОС Windows с выводом в окно текста и фигур (лабораторная работа)	Компьютерная симуляция (разбор конкретных ситуаций)	2

Таблица 6.1 – Интерактивные образовательные технологии, используемые при проведении аудиторных занятий

1	2	3	4
9	Средства ввода/вывода в ОС Windows (лабораторная работа)	Компьютерная симуляция (разбор конкретных ситуаций)	2
10	Преобразование растровых изображений (лабораторная работа)	Компьютерная симуляция (разбор конкретных ситуаций)	2
11	Построение графика функции (лабораторная работа)	Компьютерная симуляция (разбор конкретных ситуаций)	2
Итого:			20

Содержание дисциплины обладает значительным воспитательным потенциалом, поскольку в нем аккумулирован исторический и современный научный опыт человечества. Реализация воспитательного потенциала дисциплины осуществляется в рамках единого образовательного и воспитательного процесса и способствует непрерывному развитию личности каждого обучающегося. Дисциплина вносит значимый вклад в формирование профессиональной культуры обучающихся. Содержание дисциплины способствует профессионально-трудовому воспитанию обучающихся.

Реализация воспитательного потенциала дисциплины подразумевает:

- целенаправленный отбор преподавателем и включение в лекционный материал, материал для практических занятий содержания, демонстрирующего обучающимся образцы настоящего научного подвижничества создателей и представителей данной отрасли науки и производства, высокого профессионализма ученых и представителей производства, их ответственности за результаты и последствия деятельности для человека и общества; примеры подлинной нравственности людей, причастных к развитию науки и производства;

- применение технологий, форм и методов преподавания дисциплины, имеющих высокий воспитательный эффект за счет создания условий для взаимодействия обучающихся с преподавателем, другими обучающимися, представителями работодателей (командная работа, проектное обучение, разбор конкретных ситуаций, решение кейсов, мастер-классы и др.);

- личный пример преподавателя, демонстрацию им в образовательной деятельности и общении с обучающимися за рамками образовательного процесса высокой общей и профессиональной культуры.

Реализация воспитательного потенциала дисциплины на учебных занятиях направлена на поддержание в университете единой развивающей образовательной и воспитательной среды. Реализация воспитательного потенциала дисциплины в ходе самостоятельной работы обучающихся способствует развитию в них целеустремленности, инициативности, креативности, ответственности за результаты своей работы – качеств, необходимых для успешной социализации и профессионального становления.

7 Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Таблица 7.1 Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Код и содержание компетенции	Этапы формирования компетенций и дисциплины (модули), при изучении которых формируется данная компетенция		
	начальный	базовый	завершающий
1	2	3	4
ОПК-3 – владение современными информационными технологиями, готовностью применять современные средства автоматизированного проектирования и машинной графики при проектировании систем и их отдельных модулей, а также для подготовки конструкторско-технологической документации, соблюдать основные требования информационной безопасности	Информатика Инженерная и компьютерная графика	Программирование на языках низкого уровня Объектно-ориентированное программирование в мехатронике	Системы автоматизированного проектирования электронных компонентов Системы автоматизированного проектирования элементов конструкций Научно-исследовательская работа
ОПК-6 – способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Информатика Технологии конструкторских материалов. Материаловедение	Математика Программирование на языках низкого уровня Объектно-ориентированное программирование в мехатронике	-
ПК-2 – способность разрабатывать программное обеспечение, необходимое для обработки информации и управления в мехатронных и робототехнических системах, а также для их проектирования	-	Программирование на языках низкого уровня Объектно-ориентированное программирование в мехатронике	Микропроцессорная техника в мехатронике и робототехнике Программное обеспечение мехатронных и робототехнических систем Компьютерное управление бытовыми мехатронными системами Научно-исследовательская работа

**Этапы для РПД всех форм обучения определяются по учебному плану очной формы обучения следующим образом:*

Этап	Учебный план очной формы обучения/ семестр изучения дисциплины		
	Бакалавриат	Специалитет	Магистратура
<i>Начальный</i>	1-3 семестры	1-3 семестры	1 семестр
<i>Основной</i>	4-6 семестры	4-6 семестры	2 семестр
<i>Завершающий</i>	7-8 семестры	7-10 семестры	3-4 семестр

7.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Таблица 7.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

№ п/п	Код компетенции (или её части)	Показатели оценивания компетенций	Критерии и шкала оценивания компетенций		
			Пороговый уровень («удовлетворительно»)	Продвинутый уровень (хорошо)	Высокий уровень («отлично»)
1	2	3	4	5	6
1	ОПК-3 / базовый	<p>1. Доля освоенных обучающимся знаний, умений, навыков от общего объема ЗУН, установленных в п.1.3 РПД</p> <p>2. Качество освоенных обучающимся знаний, умений, навыков</p> <p>3. Умение применять знания, умения, навыки в типовых и нестандартных ситуациях</p>	<p>знать: способы обработки данных</p>	<p>знать: программные пакеты для анализа данных</p>	<p>знать: способы написания новых программных продуктов</p>
			<p>уметь: использовать готовые пакетные решения для обработки данных</p>	<p>уметь: формировать ТЗ для новых программных продуктов</p>	<p>уметь: разрабатывать новые программные продукты для обработки данных</p>
			<p>владеть: ПК на уровне продвинутый пользователь</p>	<p>владеть: одним программным пакетом для обработки данных</p>	<p>владеть: одним языком программирования для написания программных продуктов по обработке данных</p>
2	ОПК-6 / базовый	<p>1. Доля освоенных обучающимся знаний, умений, навыков от общего объема ЗУН, установленных в п.1.3 РПД</p> <p>2. Качество освоенных обучающимся знаний, умений, навыков</p> <p>3. Умение применять знания, умения, навыки в типовых и нестандартных ситуациях</p>	<p>знать: текущие проблемы мехатроники и робототехники</p>	<p>знать: методы формирования диагностических решений проблемы мехатроники и робототехники</p>	<p>знать: методы конструктивных решений проблем мехатроники и робототехники</p>
			<p>уметь: свободно использовать в профессиональной сфере современные информационные технологии, компьютерные сети, программные продукты и ресурсы Интернет для решения задач профессиональной деятельности</p>	<p>уметь: глубоко осмысливать и формировать диагностические решения проблем мехатроники и робототехники</p>	<p>уметь: глубоко осознавать основы конструкторских решений проблем мехатроники и робототехники</p>
			<p>владеть: способностью свободно владеть и использовать в профессиональной сфере современные информационные технологии</p>	<p>владеть: способностью свободно владеть и использовать в профессиональной сфере современные информационные технологии; способностью использовать современные</p>	<p>владеть: способностью свободно владеть и использовать в профессиональной сфере современные информационные технологии; способностью использовать</p>

1	2	3	4	5	6
				компьютерные сети для решения задач профессиональной деятельности	современные компьютерные сети, программные продукты и ресурсы Интернет для решения задач профессиональной деятельности
3	ПК-2 / базовый	<p>1. Доля освоенных обучающимся знаниями, умениями, навыками от общего объема ЗУН, установленных в п.1.3 РПД</p> <p>2. Качество освоенных обучающимся знаниями, умениями, навыками</p> <p>3. Умение применять знания, умения, навыки в типовых и нестандартных ситуациях</p>	<p>знать: области применения современных микроконтроллеров и микроЭВМ</p>	<p>знать: особенности управляющих вычислительных комплексов на базе микроконтроллеров и микроЭВМ</p>	<p>знать: алгоритмы управления и особенности управляющих вычислительных комплексов на базе микроконтроллеров и микроЭВМ</p>
			<p>уметь: составлять алгоритмы программ для управляющих вычислительных комплексов на базе микроконтроллеров и микроЭВМ</p>	<p>уметь: реализовывать алгоритмы управления в виде программ для управляющих вычислительных комплексов на базе микроконтроллеров и микроЭВМ</p>	<p>уметь: реализовывать алгоритмы управления в виде программ для управляющих вычислительных комплексов на базе микроконтроллеров и микроЭВМ; читать и анализировать программы для управляющих вычислительных комплексов на базе микроконтроллеров и микроЭВМ</p>
			<p>владеть: навыками составления алгоритмов программ</p>	<p>владеть: языками программирования высокого уровня</p>	<p>владеть: языками программирования высокого уровня; языком ассемблер для определенного типа микроконтроллеров</p>

7.3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Таблица 7.3 - Паспорт комплекта оценочных средств для текущего контроля

№ п/п	Раздел (тема) дисциплины	Код контролируемой компетенции и (или её части)	Технология формирования	Оценочные средства		Описание шкал оценивания
				наименование	№№ заданий	
1	2	3	4	5	6	7
1	Введение	ОПК-3, ОПК-6,	Л № 1, ЛР-1-2	КО, ЛР	защита ЛР, вопросы 1-34 (п. 7.3.1)	В соответстви и с п. 7.4
2	Принципы объектно - ориентированного программирования (ООП)	ОПК-3, ОПК-6, ПК-2,	Л № 2, ЛР-3-4	КО, ЛР	защита ЛР, вопросы 35-105 (п. 7.3.1)	В соответстви и с п. 7.4
3	Проектирование программ	ОПК-3, ОПК-6, ПК-2,	Л № 3, ЛР-5-6	КО, ЛР	защита ЛР, вопросы 106-134 (п. 7.3.1)	В соответстви и с п. 7.4
4	Классы	ОПК-3, ОПК-6, ПК-2,	Л № 4, ЛР-7	КО, ЛР	защита ЛР, вопросы 135-154 (п. 7.3.1)	В соответстви и с п. 7.4
5	Наследование	ОПК-3, ОПК-6, ПК-2,	Л № 5, ЛР-8	КО, ЛР	защита ЛР, вопросы 155-166 (п. 7.3.1)	В соответстви и с п. 7.4
6	События	ОПК-3, ОПК-6, ПК-2,	Л № 6, ЛР-9	КО, ЛР	защита ЛР, вопросы 167-169 (п. 7.3.1)	В соответстви и с п. 7.4
7	Организация функций ввода-вывода данных	ОПК-3, ОПК-6, ПК-2,	Л № 7, ЛР-10	КО, ЛР	защита ЛР, вопросы 170-190 (п. 7.3.1)	В соответстви и с п. 7.4

Примечание: КО – контрольный опрос, ЛР – лабораторная работа, ПР – практическая работа.

Форма промежуточного контроля по дисциплине – экзамен.
Перечень экзаменационных вопросов приведен в приложении А.

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине включает в себя:

- перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы;
- описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;
- типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы;
- методические материалы, определяющие процессы оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Фонд оценочных средств для проведения промежуточной аттестации приводится в соответствии с инструкцией И 02.018-2015 «Инструкция по заполнению фонда оценочных средств»

Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций:

- Положение П 02.016–2015 «О балльно-рейтинговой системе оценки качества освоения образовательных программ»;
- Список методических указаний, используемых в образовательном процессе, представлен в п. 8.2.

Полностью оценочные средства представлены в учебно-методическом комплексе дисциплины.

Для тестирования используются контрольно-измерительные материалы (КИМ) – задания в тестовой форме, составляющие банк тестовых заданий (БТЗ) по дисциплине, утвержденный в установленном в университете порядке.

Проверяемыми на промежуточной аттестации элементами содержания являются темы дисциплины, указанные в разделе 4 настоящей программы. Все темы дисциплины отражены в КИМ в равных долях (%). БТЗ включает в себя не менее 100 заданий и постоянно пополняется.

Для проверки знаний используются вопросы и задания в различных формах:

- закрытой (с выбором одного или нескольких правильных ответов), -
- открытой (необходимо вписать правильный ответ),
- на установление правильной последовательности,
- на установление соответствия.

Умения, навыки и компетенции проверяются с помощью задач (ситуационных, производственных или кейсового характера) и различного вида конструкторов. Все задачи являются многоходовыми. Некоторые задачи,

проверяющие уровень сформированности компетенций, являются многовариантными. Часть умений, навыков и компетенций прямо не отражена в формулировках задач, но они могут быть проявлены обучающимися при их решении.

В каждый вариант КИМ включаются задания по каждому проверяемому элементу содержания во всех перечисленных выше формах и разного уровня сложности. Такой формат КИМ позволяет объективно определить качество освоения обучающимися основных элементов содержания дисциплины и уровень сформированности компетенций.

Максимальное количество баллов за тестирование - 36 баллов.

7.3.1 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Вопросы по разделу (теме) 1 «Введение»:

1. Что представляет собой алфавит языка?
2. Что понимают под словами «синтаксис» и «семантика»?
3. Дать определение компилятора.
4. Для чего нужны идентификаторы?
5. Перечислить особенности ключевых слов.
6. Что такое препроцессор?
7. Что содержится в заголовочных файлах и как можно подключить их?
8. Как ввести и как вывести данные в C++-программе?
9. Как объявляют переменные целого и вещественного типа?
10. Дать определение переменной.
11. Перечислить допустимые в языке C++ типы данных.
12. Почему беззнаковые типы допускают увеличение диапазона представляемых ими положительных значений?
13. Что происходит при инициализации переменной?
14. Можно ли определить константу без ее инициализации?
15. В чем заключается процесс оптимизации по памяти во время трансляции программы? Можно ли его предотвратить, и если можно, то как именно?
Перечислить операции, относящиеся к каждому из следующих типов:
 - арифметические;
 - отношений;
 - логические и поразрядные;
 - инкремента и декремента;
 - присваивания.
16. Что такое линейные и разветвленные программы?
17. В чем заключается понятие форматного вывода данных?
18. Как осуществляется ввод данных?
19. Каковы формы условного оператора?

20. Каким образом строится условное выражение?
21. Какую структуру имеет оператор-переключатель?
22. С какой целью используется оператор разрыва break?
23. Что происходит, если забыть поставить break?
24. Дайте определение цикла.
25. Каким образом цикл while может имитировать цикл for?
26. Каким образом цикл while может имитировать цикл do-while?
27. Почему тело оператора с последующим условием нужно заключать в фигурные скобки, даже если в нем всего один оператор?
28. Почему открытый цикл всегда оказывается бесконечным?
29. В каких случаях используются операторы break, continue, exit?
30. Почему в языке C++ нет необходимости применять оператор goto?
31. Дать определение одномерного массива и матрицы.
32. Почему при объявлении статического массива его размер должен быть объявлен как константа?
33. Что происходит при обращении к элементу массива вне допустимого диапазона индексов?
34. Как выполнить инициализацию одномерного массива? Есть ли принципиальные отличия при инициализации матрицы?

Вопросы по разделу (теме) 2 «Принципы объектно - ориентированного программирования (ООП)»:

35. Что такое объект?
36. Дайте определение класса.
37. Каким образом можно описать элемент данных типа структуры?
38. Как описать переменную структурного типа?
39. Как обратиться к члену структуры, если используются вложенные структуры?
40. Верно ли, что все поля структуры должны быть различных типов?
41. Что такое структура с битовыми полями? В чем преимущества и недостатки использования структур с битовыми полями?
42. Каков максимальный уровень вложенности структур?
43. Дать определение объединения.
44. Перечислить общие и отличные черты структур и объединений.
45. Может ли быть вложено объединение в структуру? А структура в объединение?
46. Что такое перечисление? Для чего оно используется?
47. Дать определение функции. Для чего функции используются в программах?
48. Что такое описание и определение функции?
49. Для чего нужен прототип? Всегда ли его присутствие необходимо в программе, содержащей функции?
50. Признаком чего служит слово void, стоящее перед именем функции?
51. Признаком чего служит слово void, стоящее на месте параметров? Всегда ли оно необходимо?
52. Как осуществляется выход из функции?
53. В чем разница между функциями, не возвращающими никаких значений и возвращающими значения объявленного типа?

54. Что называют формальными параметрами функции и что фактическими? Как эти два понятия соотносятся между собой?
55. Какие переменные называются локальными и какие глобальными?
56. Почему локальные переменные называют автоматическими?
57. Где хранятся локальные переменные? Может ли их суммарный объем превышать доступное стековое пространство?
58. Чем характеризуются регистровые переменные?
59. В чем заключается особенность передачи массивов в качестве параметров функций?
60. Как запретить изменять значения элементов массива, переданного в функцию в качестве параметра?
61. Есть ли принципиальные отличия при передаче массивов и символьных строк как параметров функций?
62. Можно ли возвращать массив из функции, пользуясь оператором return?
63. Что происходит при передаче переменной структурного типа в функцию в качестве ее параметра?
64. Описать механизм возвращения из функции переменной структурного типа.
65. В чем заключаются преимущества и недостатки использования рекурсивных функций?
66. Что представляют собой встраиваемые функции? Перечислить преимущества и недостатки их использования.
67. Что такое указатель?
68. Привести примеры объявления указателей.
69. Почему указатели необходимо инициализировать?
70. Для чего используется оператор взятия адреса?
71. Что такое нулевой указатель и указатель на тип void? В чем заключается их принципиальное отличие?
72. Куда указывает нулевой указатель и куда неинициализированный?
73. Каким образом указатель на тип void может быть использован для обращения к типизированным данным?
74. Определить статические и динамические переменные.
75. Что такое «куча»? Какие элементы она содержит?
76. Что общего у функций malloc() и calloc() и в чем состоят их различия?
77. Как освободить память?
78. Всегда ли возможна работа с «кучей»?
79. Можно ли использовать содержимое динамически выделяемой памяти после того, как она была освобождена?
80. Почему функции free() нет необходимости сообщать, сколько байтов нужно освободить?
81. Чем может быть вызвана «утечка памяти»?
82. Можно ли захватить память функциями malloc() или calloc() и освободить ее операцией delete?
83. Как избежать случайного использования незащищенных областей памяти?
84. Как при резервировании памяти сразу присвоить значение созданному объекту?
85. Что могут адресовать указатели?

86. К чему приводит увеличение указателя на некоторую величину?
87. Почему можно разыменовать имя массива, но нельзя определить его адрес?
88. Как осуществляется работа с динамически выделенным массивом?
89. Как выделить память под матрицу?
90. Как происходит обращение к полям структуры, если она адресуется указателем?
91. На что указывает оператор доступа к полю структуры \rightarrow , стоящий между указателем структуры и именем поля?
92. Что происходит, если не освободить память, занятую под массивы и структуры после завершения работы с ними?
93. Дать определение ссылки.
94. В чем заключаются особенности передачи параметров функций по значению, по указателю и по ссылке?
95. Как защитить данные от несанкционированного изменения их функцией?
96. Почему следует особенно внимательно относиться к параметрам, передаваемым по ссылке?
97. Как реализовать возврат из функции более чем одного значения?
98. Что такое динамические структуры?
99. Почему динамические структуры называют самоссылочными?
100. На что указывает указатель на структуру?
101. Приведите примеры динамических структур.
102. Перечислите действия, необходимые для создания списка.
103. Сколько информационных полей и какого типа может иметь самоссылочная структура?
104. По какому принципу организован стек и по какому очередь?
105. В чем заключаются особенности организации двусвязных списков?

Вопросы по разделу (теме) 3 «Проектирование программ»:

106. Составить программу, которая при вводе оценки в виде цифры выводит оценку в буквенном виде: 5 – отлично, 4 – хорошо, 3 – удовлетворительно, 2 – неудовлетворительно.
107. Составить программу, которая по введенному номеру месяца выводит его название и время года.
108. Составить программу, которая при вводе символа определяет, скобка ли это, и указывает, какая именно, например фигурная открывающая ($\{$), квадратная закрывающая ($\}$).
109. Составить программу, которая при вводе символа выводит либо текст «цифра», если введена цифра, либо текст «латинская буква», если введена латинская буква, либо текст «не цифра и не латинская буква» в остальных случаях.
110. Ввести с клавиатуры координаты точек (x_1, y_1) и (x_2, y_2) и определить расстояние между этими точками.
111. Поменять местами значения переменных x и y .
112. Значения переменных a, b, c поменять местами так, чтобы оказалось $a \geq b \geq c$.
113. Определить: $y = \max(\min(a,b), \min(c,d))$.
114. Определить: $y = \min(a,b,c)$.

115. Переменной k присвоить номер четверти координатной плоскости, в которой находится точка с координатами, введенными с клавиатуры. Отдельно учесть случаи, когда точка попадает на одну из координатных осей или в начало координат.
116. Написать программу преобразования прописных латинских букв в строчные. При написании программы использовать условное выражение.
117. Написать программу решения квадратного уравнения с произвольными коэффициентами, которые вводятся с клавиатуры.
118. Определить $d = \max(a, b, c)$, если значения переменных a , b и c введены с клавиатуры.
119. Перечислите известные методы сортировки массива.
120. В чем заключается и чем обусловлена необходимость оптимизации пузырькового метода сортировки?
121. Что общего при работе с матрицей в целом и обработке ее отдельно взятых строк? В чем различия?
122. Как располагаются в оперативной памяти элементы массивов?
123. Дан массив, содержащий 20 действительных чисел. Определить, сколько из них отличается от последнего элемента массива.
124. Найти максимальный и минимальный элементы 15-элементного действительного массива и их индексы.
125. Сдвинуть циклически элементы 20-элементного целочисленного массива на две позиции вправо.
126. Определить сумму и количество элементов массива, объявленного как `int mas[20]`, попадающего в заданный с клавиатуры диапазон.
127. Не пользуясь дополнительным массивом, переписать массив `float arr[10]` с обратной стороны.
128. В действительной матрице 3×5 заменить нулями элементы, меньшие среднего арифметического значения этой матрицы, и единицами – большие.
129. Дана матрица 5×7 , элементами которой являются значения символьного типа. Составить одномерный массив, содержащий количество символов m в каждом из ее столбцов. Подсчитать количество символов a над третьей строкой матрицы и общее количество символов b и z под третьей строкой.
130. Дана действительная матрица 7×7 . Найти минимальное значение среди элементов, стоящих над главной диагональю, и максимальное значение среди элементов, находящихся ниже главной диагонали, а также их местоположение.
131. В квадратной действительной матрице 9-го порядка определить количество отрицательных элементов над побочной диагональю и сумму положительных элементов под главной диагональю.
132. Дана действительная матрица 6×8 . Найти сумму элементов каждой строки верхней половины матрицы и произведение элементов каждой строки ее нижней половины. Определить значение и местоположение максимального элемента верхней половины матрицы и минимального элемента ее нижней половины.
133. Дана целочисленная матрица 4×7 . Определить минимальное и максимальное значение матрицы и их местоположение. Найти среднее арифметическое

значение положительных элементов и модулей отрицательных элементов в каждом столбце матрицы. Результаты вычислений записать в одномерные массивы.

134. Дана вещественная квадратная матрица порядка 5. Получить новую матрицу путем прибавления к элементам каждой строки матрицы наименьшего значения элементов этой строки.

Вопросы по разделу (теме) 4 «Классы»:

135. Что такое объект?
136. Дайте определение класса.
137. Что называют данными-элементами класса и функциями-членами класса?
138. Как называется механизм, объединяющий данные и программный код, реализующий обработку этих данных?
139. В чем заключаются особенности открытых и закрытых членов класса?
140. Что такое статические данные-элементы? Как они объявляются и где используются?
141. Какие спецификации доступа используются при описании класса?
142. Что является элементами класса?
143. Как осуществляется доступ к элементам класса?
144. Для чего используется указатель `this`?
145. Что такое конструктор?
146. Что такое деструктор?
147. Как может осуществляться передача объектов в функции?
148. Что такое указатель `this`? Для чего он используется?
149. Как организовать возвращение объекта из функции?
150. Что такое дружественные функции, дружественные классы?
151. Перечислите основные свойства дружественных функций и дружественных классов.
152. Как осуществляется перегрузка операций?
153. В чем состоят различия между перегрузкой операций с использованием дружественных функций и без них?
154. Какую роль играет указатель `this` при перегрузке операторов?

Вопросы по разделу (теме) 5 «Наследование»:

155. В чем заключаются различия между наследованием со спецификаторами `public`, `protected`, `private`?
156. Что такое простое наследование?
157. Что такое множественное наследование?
158. Как реализовать иерархию классов?
159. Что называется виртуальной функцией? Где и каким образом она объявляется?
160. Для чего необходимы виртуальные функции?
161. Что представляет собой динамический полиморфизм (`run-time polymorphism`), статический полиморфизм? С помощью каких механизмов они реализуются?
162. Как создаются указатели на базовый и производные классы? Перечислите их основные свойства.

163. Какую роль играют виртуальные функции при наследовании?
164. Дайте определение абстрактных классов и чисто виртуальных функций.
165. Как объявляется чисто виртуальная функция?
166. Когда необходимы виртуальные базовые классы?

Вопросы по разделу (теме) 6 «События»:

167. Что такое исключения (исключительные ситуации)? Когда они возникают?
168. Перечислите этапы обработки исключительных ситуаций.
169. Почему проблема поиска ошибок более актуальна при использовании классов?

Вопросы по разделу (теме) 7 «Организация функций ввода-вывода данных»:

170. Что такое поток?
171. Что представляет собой файловый указатель?
172. Перечислите режимы доступа к файлу.
173. Как открыть и как закрыть файл?
174. Особенности работы с двоичными файлами.
175. Что представляет собой файловый указатель?
176. Как организовать доступ к произвольному месту двоичного файла?
177. Что включает в себя понятие файла? Как оно связано со стандартными потоками ввода-вывода?
178. Перечислите стандартные потоки ввода-вывода, открываемые при запуске программы на языках C и C++.
179. Как связаны между собой понятия логического и физического файлов?
180. Как работает функция `open()`?
181. Что такое режим доступа? Перечислите возможные режимы доступа при работе с текстовыми файлами.
182. Почему при работе с файлами не рекомендуется использовать оператор цикла с последующим условием?
183. Каков механизм действий, связанных с закрытием файла? Почему необходимо закрывать файлы по окончании работы с ними?
184. Как открыть двоичный файл?
185. Какие функции осуществляют чтение и запись при работе с двоичными файлами?
186. В чем состоят особенности чтения массивов из двоичных файлов и их записи в двоичные файлы?
187. Что такое последовательный и произвольный доступ к компонентам файла?
188. В чем заключается опасность использования функции `fseek()`?
189. Как «подойти» к предпоследнему элементу файла? Привести все возможные варианты.
190. Для чего используется функция `ftell()`?

7.4 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, регулируются следующими нормативными актами университета:

- Положение П 02.016–2015 «О балльно-рейтинговой системе оценки качества освоения образовательных программ»;

- методические указания, используемые в образовательном процессе, указанные в списке литературы.

Для *текущего контроля* по дисциплине в рамках действующей в университете балльно-рейтинговой системы применяется следующий порядок начисления баллов:

Таблица 7.4 – Порядок начисления баллов в рамках БРС

Формы текущего контроля	Минимальный балл		Максимальный балл	
	балл	примечание	балл	примечание
1	2	3	4	5
Вещественные типы данных (лабораторная работа №1)	2	Задания выполнены, но не защищены	4	Задания выполнены верно на 80% и более
Циклы (лабораторная работа №2)	2	Задания выполнены, но не защищены	4	Задания выполнены верно на 80% и более
Одномерные массивы (лабораторная работа №3)	2	Задания выполнены, но не защищены	5	Задания выполнены верно на 80% и более
Сортировка массива (лабораторная работа №4)	2	Задания выполнены, но не защищены	5	Задания выполнены верно на 80% и более
Структуры (лабораторная работа №5)	2	Задания выполнены, но не защищены	5	Задания выполнены верно на 80% и более
Строки и файлы (лабораторная работа №6)	2	Задания выполнены, но не защищены	5	Задания выполнены верно на 80% и более
Приложение ОС Windows с выводом в окно текста и фигур (лабораторная работа №7)	3	Задания выполнены, но не защищены	5	Задания выполнены верно на 80% и более
Средства ввода/вывода в ОС Windows (лабораторная работа №8)	3	Задания выполнены, но не защищены	5	Задания выполнены верно на 80% и более
Преобразование растровых изображений (лабораторная работа №9)	3	Задания выполнены, но не защищены	5	Задания выполнены верно на 80% и более

1	2	3	4	5
Построение графика функции (лабораторная работа №10)	3	Задания выполнены, но не защищены	5	Задания выполнены верно на 80% и более
Итого - успеваемость:	24		48	
Посещаемость 1 к.т.	0	присутствовал менее чем на 20% занятий	4	присутствовал более чем на 80% занятий
Посещаемость 2 к.т.	0	присутствовал менее чем на 20% занятий	4	присутствовал более чем на 80% занятий
Посещаемость 3 к.т.	0	присутствовал менее чем на 20% занятий	4	присутствовал более чем на 80% занятий
Посещаемость 4 к.т.	0	присутствовал менее чем на 20% занятий	4	присутствовал более чем на 80% занятий
Итого - посещаемость:	0		16	
Экзамен	0		36	
Итого баллов за семестр	24		100	

Для промежуточной аттестации, проводимой в форме тестирования, используется следующая методика оценивания знаний, умений, навыков и (или) опыта деятельности. В каждом варианте КИМ - 18 заданий (18 вопросов в закрытой форме).

Каждый верный ответ оценивается следующим образом: задание в закрытой форме – 2 балла.

Максимальное количество баллов за тестирование - 36 баллов

8 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

8.1 Основная учебная литература

1. Лапина, Т. И. Методы и технологии объектно-ориентированного программирования [Текст] : учебное пособие / Т. И. Лапина, Е. А. Петрик ; Юго-Зап. гос. ун-т. – Курск : ЮЗГУ, 2011. - 131 с.

2. Лапина, Т. И. Методы и технологии объектно-ориентированного программирования [Электронный ресурс] : учебное пособие / Т. И. Лапина, Е. А. Петрик ; Юго-Зап. гос. ун-т. - Курск : ЮЗГУ, 2011. - 131 с.

8.2 Дополнительная литература

3. Лафоре, Р. Объектно-ориентированное программирование в С++ [Текст] / Р. Лафоре. - 4-е изд. - СПб. [и др.]: Питер, 2012. - 928 с.

4. Информатика. Базовый курс [Текст] : учебное пособие / под ред. С. В. Симоновича. - 2-е изд. - СПб. : Питер, 2010. - 640 с.

5. Иванова, Г. С. Технология программирования [Текст] : учебник / Галина Сергеевна Иванова. - М. : Кнорус, 2011. - 336 с.

8.3 Перечень методических указаний

1. Объектно-ориентированное программирование в мехатронике [Электронный ресурс] : методические указания по выполнению лабораторных работ по дисциплине «Объектно-ориентированное программирование в мехатронике» по направлению 15.03.06 - «Мехатроника и робототехника» / Юго-Зап. гос. ун-т ; сост. П. А. Безмен. - Курск : ЮЗГУ, 2016. - 80 с.

2. Методические указания по организации самостоятельной работы студентов направления 15.03.06 «Мехатроника и робототехника» / Юго-Зап. гос. ун-т ; сост.: Е. Н. Политов, Л. Ю. Ворочаева, А. В. Мальчиков. - Курск : ЮЗГУ, 2017. - 31 с.

8.4 Другие учебно-методические материалы

Иллюстрационные материалы (мультимедийные презентации).

Журнал "Мехатроника, автоматизация, управление".

Журнал "Проблемы управления / CONTROLSCIENCES".

9 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Электронная библиотека ЮЗГУ <http://www.lib.swsu.ru/>
2. Информационная система «Единое окно доступа к образовательным ресурсам» <http://window.edu.ru/library>
3. Электронно-библиотечная система «Университетская библиотека online» <http://www.biblioclub.ru>

10 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины необходимо начинать с изучения теоретических положений и законов, воспользовавшись учебником, учебным пособием, либо конспектом лекций. В рабочей программе представлены список литературы, методических пособий и указаний, которые необходимо использовать при выполнении заданий лабораторных занятий. Конспект лекций студенты обязаны вести на занятиях.

Лабораторные занятия включают в себя:

- а) теоретическую подготовку студентов к занятию, в ходе которой студент обязан осмыслить теоретический материал, выносимый на занятие;
- б) выполнение индивидуальных заданий на самом лабораторном занятии.

11 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

1. Интегрированная среда разработки Microsoft Visual Studio Professional 2015.
2. Интегрированная среда разработки Eclipse.
3. Пакет программ LibreOffice.
4. Операционная система Microsoft Windows.

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

Работа студентов организована в аудитории в соответствии с расписанием.

Персональные компьютеры с доступом в Интернет для преподавателей и студентов; аудитории, оснащенные мультимедийными средствами обучения для чтения лекций, проведения семинарских занятий, проверки самостоятельных работ.

Наглядность и эффективность докладов (презентаций, лекционного материала) достигается с помощью Мультимедиа-центра: ноутбука Lenovo G710, проектора BenQ MX505 и интерактивной системой с короткофокусным проектором ActivBoard.

Компьютерный класс (ауд. Г-217а, Г-217б, главный учебный корпус ЮЗГУ).

13 Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

При обучении лиц с ограниченными возможностями здоровья учитываются их индивидуальные психофизические особенности. Обучение инвалидов осуществляется также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).

Для лиц с нарушением слуха возможно предоставление учебной информации в визуальной форме (краткий конспект лекций; тексты заданий, напечатанные увеличенным шрифтом), на аудиторных занятиях допускается присутствие ассистента, а также сурдопереводчиков и тифлосурдопереводчиков. Текущий контроль успеваемости осуществляется в письменной форме: обучающийся письменно отвечает на вопросы, письменно выполняет практические задания. Доклад (реферат) также может быть представлен в письменной форме, при этом требования к содержанию остаются теми же, а требования к качеству изложения материала (понятность, качество речи, взаимодействие с аудиторией и т. д.) заменяются на соответствующие требования, предъявляемые к письменным работам (качество оформления текста и списка литературы, грамотность, наличие иллюстрационных материалов и т.д.). Промежуточная аттестация для лиц с нарушениями слуха проводится в письменной форме, при этом используются общие критерии оценивания. При необходимости время подготовки к ответу может быть увеличено.

Для лиц с нарушением зрения допускается аудиальное предоставление информации, а также использование на аудиторных занятиях звукозаписывающих устройств (диктофонов и т.д.). Допускается присутствие на занятиях ассистента (помощника), оказывающего обучающимся необходимую техническую помощь. Текущий контроль успеваемости осуществляется в устной форме. При проведении промежуточной аттестации для лиц с нарушением зрения тестирование может быть заменено на устное собеседование по вопросам.

Для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, на аудиторных занятиях, а также при проведении процедур текущего контроля успеваемости и промежуточной аттестации могут быть предоставлены необходимые технические средства (персональный компьютер, ноутбук или другой гаджет); допускается присутствие ассистента (ассистентов), оказывающего обучающимся необходимую техническую помощь (занять рабочее место, передвигаться по аудитории, прочитать задание, оформить ответ, общаться с преподавателем).

**14 Лист дополнений и изменений, внесенных в рабочую программу
дисциплины**

Номер изме- нения	Номера страниц				Всего страниц	Дата	Основание для изменения и подпись лица, проводившего изменения
	изме- ненных	замене- нных	аннулиро- ванных	новых			

Приложение А

Вопросы для подготовки к экзамену

1. Ключевые слова языка C++.
2. Функция main().
3. Операторы сравнения в языке C++.
4. Основные типы данных языка C++.
5. Логические переменные в языке C++.
6. Типы переменных в языке C++ для чисел с плавающей запятой.
7. Приведение типов в языке C.
8. Объявление массивов в языке C++.
9. Описание типа «строка» как массива в языке C++.
10. Объявление структуры в языке C++.
11. Логические операторы в языке C++.
12. Объявление массива структур в языке C++.
13. Условный оператор в языке C++.
14. Цикл со счетчиком for в языке C++.
15. Оператор return в языке C++.
16. Цикл while в языке C++.
17. Оператор перехода goto в языке C++.
18. Описание функции на языке C++, возвращающей величину простого типа.
19. Типы переменных в языке C++ для целых чисел.
20. Описание функции на языке C++, не возвращающей величину, используя ключевое слово void.
21. Цикл с постусловием do-while в языке C++.
22. Операторы инкремента ++ и декремента -- в языке C++.
23. Оператор break в языке C++.
24. Ввод и вывод символов используя функции getchar() и putchar().
25. Понятие глобальной переменной.
26. Ввод и вывод строк используя функции gets() и puts().
27. Понятие рекурсии.
28. Использование стандартных потоков ввода cin и вывода cout.
29. Объявление констант в языке C++.
30. Комментарии в текстах программ на языке C++.
31. Классы. Методы, свойства, указатели.
32. Создание объекта класса.
33. Классы как абстрактные типы.
34. Абстрактные методы.

35. Конструкторы и деструкторы.
36. Наследование классов.
37. Множественное наследование.
38. Виртуальные базовые классы.
39. Обработка исключительных ситуаций.
40. Члены класса и управление доступом.
41. Реализация полиморфизма используя перегрузку функций.
42. Реализация динамического полиморфизма используя вызов виртуальных функций через указатель на объект.
43. Дружественные функции и классы.
44. Чтение и запись текстовых файлов средствами языка С.
45. Чтение и запись бинарных файлов средствами языка С.