Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 27.10.2023 11:23:19
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ
Декан факультета машиностроения
_____ /Сафонов Е.В./
«12» сентября 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования и алгоритмизации в машиностроении»

Направление подготовки
15.03.01 Машиностроение

Образовательная программа (профиль подготовки)
«Высокоэффективные технологические процессы и оборудование»

Квалификация (степень) выпускника
Бакалавр

Форма обучения
Заочная

Москва 2022 г.

Разработчики:

К.т.н., доцент  /А.В. Кузнецов/

Согласовано:

Заведующий кафедрой
«Автоматика и управление»

доцент, к.т.н.  /А.В. Кузнецов/

Заведующий кафедрой «ТиОМ»,

к.т.н., доцент  /А.Н. Васильев/

1. Цели освоения дисциплины

К основным целям освоения дисциплины «Основы программирования и алгоритмизации в машиностроении» следует отнести:

- формирование у студентов знаний в области разработки и проектирования программного обеспечения систем автоматизации и управления;
- приобретение студентами знаний технологии программирования, умений и навыков разработки прикладных программ;
- развитие способности студентов к самостоятельному изучению и освоению новых перспективных технологий программирования;
- подготовка студентов к деятельности в соответствии с квалификационной характеристикой бакалавра по направлению, в том числе формирование умений использовать освоенные технологии программирования в профессиональной деятельности.

К основным задачам освоения дисциплины «Основы программирования и алгоритмизации в машиностроении» следует отнести:

- овладение научной терминологией в области проектирования и использования программного обеспечения в системах автоматизации и управления техническими объектами;
- овладение методологией проектирования и нормативной документацией для приобретения навыков разработки прикладных программ;
- практическое освоение технологии программирования;
- изучение способов подготовки и принятия решений по оценке эффективности технологий программирования как на начальном этапе проектирования, так и конечном этапе прекращения сопровождения программ, находящихся в эксплуатации.

2. Место дисциплины в структуре ООП бакалавриата

Дисциплина «Основы программирования и алгоритмизации в машиностроении» относится к числу учебных дисциплин базовой части блока 1 (Б.1.1.19) основной образовательной программы бакалавриата.

Дисциплина «Основы программирования и алгоритмизации в машиностроении» взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ООП:

в базовой части (Б.1.1):

- Цифровая грамотность;
- Автоматизация и роботизация процессов производства изделий.

3.Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины (модуля) у обучающихся формируются следующие компетенции и должны быть достигнуты следующие результаты обучения как этап формирования соответствующих компетенций:

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать	Перечень планируемых результатов обучения по дисциплине
ОПК-2	Способен применять основные методы, способы и средства получения, хранения, переработки информации при решении задач профессиональной деятельности;	**знать:** • структуры данных, используемые для представления типовых информационных объектов; • программные компоненты системы программирования. **уметь:** • разрабатывать алгоритмы решения задач применительно к процессам и системам управления техническими (технологическими) объектами; • осуществлять инсталляцию и настройку инструментальных средств для разработки программ. **владеть:** • основными технологиями программирования; • способами оценки эффективности инструментальных средств и технологий программирования с целью принятия решений по их применению.
ОПК-4	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.	**знать:** • типовые алгоритмы обработки данных; • методы защиты программных продуктов. **уметь:** • использовать язык программирования для создания программы; • выполнять отладку и тестирование программы **владеть:** • навыками чтения и составления технической документации на программный продукт; • навыками использования инструментальных программных средств в процессе разработки и сопровождения программных продуктов.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет **4** зачетных единиц, т.е. **144** академических часа (из них 120 часа – самостоятельная работа студентов).

Разделы дисциплины «Основы программирования и алгоритмизации в машиностроении» изучаются на первом курсе во втором семестре и втором курсе на третьем семестре.

Второй семестр: лекции – 6 часов, лабораторные работы – 10 часов, форма контроля – зачет.

Третий семестр: лекции – 2 часов, лабораторные работы – 6 часов, форма контроля – экзамен.

Структура и содержание дисциплины «Основы программирования и алгоритмизации в машиностроении» по срокам и видам работы отражены в приложении.

Содержание разделов дисциплины

Второй семестр

Программирование как составляющая процесса разработки программного обеспечения

Общая характеристика процесса разработки программного обеспечения. Ключевые этапы разработки программного обеспечения. Модели процесса разработки программного обеспечения. Методологии разработки программного обеспечения. Защита программного обеспечения.

Алгоритмические основы программирования

Понятие алгоритма. Исполнители алгоритмов. Свойства алгоритмов. Типы алгоритмов. Формы представления алгоритмов. Методы разработки и анализа алгоритмов.

Системы программирования

Понятие системы программирования. Программные компоненты системы программирования. Редактор текста. Трансляторы. Компоновщик. Отладчик. Библиотеки подпрограмм. Языки программирования. Уровни языков. Поколения языков. Интегрированные среды разработки.

Язык программирования C/C++

Элементы языка С. Базовые типы данных. Операции языка С. Управляющие операторы. Операторы форматного ввода и вывода. Структура программы. Математические функции. Форматы вывода данных. Особенности арифметических операций. Программирование ветвящихся и циклических алгоритмов. Массивы. Обработка массивов. Указатели. Функции. Файлы. Символьные переменные и строки. Преобразование строк. Форматирование строк. Структуры. Объектно-ориентированное программирование. Классы. Дружественные функции и классы. Статические элементы и функции. Константные функции. Перегрузка операций. Шаблоны.

Третий семестр

Структуры и алгоритмы обработки данных

Понятие о структурах данных. Простые структуры и типы данных. Составные линейные типы данных. Массив. Очередь. Стек. Дек. Линейные списки. Составные нелинейные типы данных. Древовидные структуры данных. Элементы теории графов. Внешние структуры данных. Файлы. Базы данных. Алгоритмы обработки данных. Поиск. Сортировка. Параллельные алгоритмы.

Параллельный поиск. Параллельная сортировка. Параллельные алгоритмы на графах. Современные алгоритмы обработки данных.

Технологии программирования

Понятие технологии программирования. Понятие жизненного цикла ПС. Цели и структура современных моделей жизненного цикла ПС. Содержание отдельных этапов разработки ПС. Стандартизация жизненного цикла ПС. Планирование, управление и тестирование программного обеспечения. Отладка программ. Документирование, сопровождение, реинжиниринг и управление качеством. Задачи стандартизации программных средств. Стандарты ISO, SW-CMM. CASE-технологии

Современные технологии программирования. Технология визуального программирования. Технологии для создания и работы интернет приложений.

Оценка сложности алгоритмов и программ

Сравнение алгоритмов. Критерии оценки сложности. Виды сложностей. Пространственная сложность. Временная сложность. Асимптотическая оценка сложности.

5. Образовательные технологии

Методика преподавания дисциплины «Основы программирования и алгоритмизации в машиностроении» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся:

– проведение активных и интерактивных лекций;

– подготовка к выполнению лабораторных работ в специализированных аудиториях вуза;

– защита выполненных лабораторных работ;

– подготовка, представление и обсуждение докладов на семинарских занятиях;

– организация и проведение текущего контроля знаний студентов в форме устного опроса;

– организация и проведение текущего контроля знаний студентов в форме бланкового/компьютерного тестирования.

Удельный вес занятий, проводимых в интерактивных формах, определен главной целью образовательной программы, особенностью контингента обучающихся и содержанием дисциплины «Основы программирования и алгоритмизации в машиностроении» и в целом по дисциплине составляет 50% аудиторных занятий.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов.

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточной аттестации.

В первом семестре:

- устный опрос;
- подготовка к выполнению лабораторных работ и их защита;
- бланковое/компьютерное тестирование;
- зачет по материалам первого семестра.

Во втором семестре:

- устный опрос;
- подготовка к выполнению лабораторных работ и их защита;
- бланковое/компьютерное тестирование;
- экзамен по материалам первого и второго семестрах.

Оценочные средства текущего контроля успеваемости включают контрольные вопросы к лабораторным работам, темы докладов по отдельным разделам дисциплины, задания в форме тестирования для контроля освоения обучающимися разделов дисциплины. Оценочные средства для текущей промежуточной аттестации по итогам освоения дисциплины включают вопросы и задания к экзамену.

Образцы тестовых контрольных вопросов и заданий для проведения текущего контроля, перечень вопросов к экзамену приведены в Приложении 1.

6.1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю).

6.1.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

В результате освоения дисциплины (модуля) формируются следующие компетенции:

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать
ОПК-2	Способен применять основные методы, способы и средства получения, хранения, переработки информации при решении задач профессиональной деятельности.
ОПК-4	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе

освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

6.1.2. Описание показателей и критериев оценивания компетенций, формируемых по итогам освоения дисциплины (модуля), описание шкал оценивания

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю).

ОПК-2 - Способен применять основные методы, способы и средства получения, хранения, переработки информации при решении задач профессиональной деятельности

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
знать: • структуры данных, используемые для представления типовых информационных объектов	Обучающийся демонстрирует полное отсутствие знаний структур данных, используемых для представления типовых информационных объектов.	Обучающийся демонстрирует неполное соответствие знаний структур данных, используемых для представления типовых информационных объектов. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие знаний структур данных, используемых для представления типовых информационных объектов. Допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие знаний структур данных, используемых для представления типовых информационных объектов. Свободно оперирует приобретенными знаниями.
уметь: • разрабатывать алгоритмы решения задач применительно к процессам и системам управления техническими (технологическ	Обучающийся не умеет разрабатывать алгоритмы решения задач применительно к процессам и системам управления техническими	Обучающийся демонстрирует неполное соответствие следующих умений: пользоваться разрабатывать алгоритмы решения задач	Обучающийся демонстрирует частичное соответствие следующих умений: разрабатывать алгоритмы решения задач применительно	Обучающийся демонстрирует полное соответствие следующих умений: разрабатывать алгоритмы решения задач применительно к

ими) объектами	(технологическими) объектами.	применительно к процессам и системам управления техническими (технологическими) объектами. Допускаются значительные ошибки, проявляется недостаточность умений, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	к процессам и системам управления техническими (технологическими) объектами. Умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	процессам и системам управления техническими (технологическими) объектами. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
владеть: • основными технологиями программирования	Обучающийся не владеет навыками использования основных технологий программирования.	Обучающийся в неполном объеме владеет навыками использования основных технологий программирования. Допускаются значительные ошибки, проявляется недостаточность владения навыками по ряду показателей. Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	Обучающийся частично владеет навыками поддержки использования основных технологий программирования. Навыки освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся в полном объеме владеет навыками использования основных технологий программирования. Свободно применяет полученные навыки в ситуациях повышенной сложности.

ОПК-4 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**

знать: типовые алгоритмы обработки данных	Обучающийся допускает существенные ошибки при раскрытии содержания и особенностей типовых алгоритмов обработки данных.	Обучающийся демонстрирует частичное знание содержания типовых алгоритмов обработки данных.	Обучающийся демонстрирует знание содержания и особенностей типовых алгоритмов обработки данных.	Обучающийся владеет полной системой знаний о типовых алгоритмов обработки данных. Свободно оперирует приобретенными знаниями.
уметь: • использовать язык программирования для создания программы	Обучающийся не умеет использовать язык программирования для создания программы.	Обучающийся демонстрирует неполное умение использовать язык программирования для создания программы.	Обучающийся демонстрирует частичное умение использовать язык программирования для создания программы.	Обучающийся. готов и умеет умение использовать язык программирования для создания программы. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
владеть: • навыками чтения и составления технической документации на программный продукт	Обучающийся владеет навыками чтения и составления технической документации на программный продукт	Обучающийся владеет отдельными навыками чтения и составления технической документации на программный продукт. Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	Обучающийся демонстрирует возможность навыками чтения и составления технической документации на программный продукт. Навыки освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые,	Обучающийся демонстрирует полное владение навыками чтения и составления технической документации на программный продукт. Свободно применяет полученные навыки в ситуациях повышенной сложности.

			нестандартные ситуации.	

Шкалы оценивания результатов промежуточной аттестации и их описание

Форма промежуточной аттестации: зачет.

Промежуточная аттестация обучающихся в форме зачёта проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «зачтено» или «не зачтено».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы.

Шкала оценивания	**Описание**
Зачтено	Выполнены все виды учебной работы, предусмотренные учебным планом: студенты прошли текущий контроль в форме компьютерного тестирования, выступили с докладами. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Не зачтено	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

Форма промежуточной аттестации: экзамен.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка

степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Основы программирования и алгоритмизации в машиностроении» (прошли промежуточный контроль, выполнили и защитили лабораторные работы).

Шкала оценивания	**Описание**
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателям, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности, не испытывает затруднений при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует частичное соответствие знаний, умений, навыков приведенным в таблицах показателям, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателям, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует полное отсутствие или недостаточное соответствие знаний, умений, навыков приведенным в таблицах показателям, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент не может оперировать знаниями и умениями при их переносе на новые ситуации.

Фонды оценочных средств представлены в Приложении 1 к рабочей программе.

7. Учебно-методическое и информационное обеспечение дисциплины.

а) основная литература:

1. Конова, Е.А. Алгоритмы и программы. Язык С++. [Электронный ресурс] / Е.А. Конова, Г.А. Поллак. — Электрон. дан. — СПб. : Лань, 2017. — 384 с. — Режим доступа: http://e.lanbook.com/book/90158 — Загл. с экрана.
2. Окулов, С.М. Задачи по программированию. [Электронный ресурс] / С.М. Окулов, Т.В. Ашихмина, Н.А. Бушмелева, М.А. Корчёмкин. — Электрон. дан. — М. : Издательство "Лаборатория знаний", 2017. — 826 с. — Режим доступа: http://e.lanbook.com/book/94162 — Загл. с экрана.
3. Павловская Т.А. С/С++. Структурное и объектно-ориентированное программирование : практикум /Т.А.Павловская, Ю.А.Щупак .-СПб. :Питер , 2011.-352с.:ил. -(Учебное пособие)
4. Павловская Т.А. С/С++. Программирование на языке высокого уровня: для магистров и бакалавров : учеб. для вузов . Гриф МО /Т.А.Павловская .-СПб. : Питер , 2014.-461с.:ил. -(Учебник для вузов)

б) дополнительная литература:

1. Алгоритмизация задач. Массивы. [Электронный ресурс] — Электрон. дан. — М. : МГТУ им. Н.Э. Баумана, 2009. — 24 с. — Режим доступа: http://e.lanbook.com/book/52572 — Загл. с экрана.
2. Долгов, А.И. Алгоритмизация прикладных задач. [Электронный ресурс] — Электрон. дан. — М. : ФЛИНТА, 2016. — 136 с. — Режим доступа: http://e.lanbook.com/book/85872 — Загл. с экрана.
3. Иванов, И.П. Сборник задач по курсу «Алгоритмы и структуры данных». [Электронный ресурс] / И.П. Иванов, А.Ю. Голубков, С.Ю. Скоробогатов. — Электрон. дан. — М. : МГТУ им. Н.Э. Баумана, 2013. — 32 с. — Режим доступа: http://e.lanbook.com/book/52435 — Загл. с экрана.
4. Иванова Г.С. Технология программирования. – М.: Изд-во КноРус, 2011.
5. Керниган, Б. Язык программирования С / Бриан В. Керниган, Деннис Ритчи .-2-е изд., перераб. и доп .-М. : Вильямс ,2012.-304с.:ил. -(Серия книг по программированию от Prentice Hall)
6. Москвитина, О.А. Сборник примеров и задач по программированию. [Электронный ресурс] / О.А. Москвитина, В.С. Новичков, А.Н. Пылькин. — Электрон. дан. — М. : Горячая линия-Телеком, 2014. — 245 с. — Режим доступа: http://e.lanbook.com/book/64090 — Загл. с экрана.
7. Окулов, С.М. Алгоритмы обработки строк. [Электронный ресурс] — Электрон. дан. — М. : Издательство "Лаборатория знаний", 2015. — 258 с. — Режим доступа: http://e.lanbook.com/book/66113 — Загл. с экрана.
8. Шилдт, Герберт С++. Базовый курс : пер. с англ. /Герберт Шилдт .-3-е изд .-М. :Вильямс , 2014.-624с.:ил.

в) программное обеспечение и интернет-ресурсы:

Программное обеспечение:

Интегрированная среда разработки Microsoft Visual Studio 2013 и выше.

Интернет-ресурсы включают учебно-методические материалы в электронном виде, представленные на сайте http://lib.mami.ru/ в разделе «Электронные ресурсы»

Полезные учебно-методические и информационные материалы представлены на сайтах:

- www.programmersforum.ru – форум программистов – Клуб программистов;
- www.cyberguru.ru – сайт о программировании, языках программирования – Кибергуру;
- www.codenet.ru – CodeNet - все для программиста;
- http://school.sgu.ru/ – портал обучения информатике и программированию.

8. Материально-техническое обеспечение дисциплины.

Специализированная учебная аудитория 2614 кафедры «Автоматика и управление», оснащенная мультимедийными средствами обучения и персональными компьютерами.

Персональный компьютер

Системные требования к оборудованию:

- Процессор с тактовой частотой 1,6 ГГц или большей
- ОЗУ объемом 1 ГБ (1,5 ГБ для работы на виртуальной машине)
- 5 ГБ доступного пространства на жестком диске
- Жесткий диск с частотой вращения 5400 об/мин
- Видеокарта с поддержкой DirectX 9 и разрешения экрана не менее 1024x768

Методические материалы по дисциплине:

1. Поповкин А.В., Сидорова М.Н., Туманов А.А. Методические указания по выполнению лабораторных работ по дисциплине «Основы программирования и алгоритмизации в машиностроении». Часть 1. Электронный ресурс для студ., обуч. по направ. 220400.62 ... и 220700.62 ... под ред. В.И. Харитонова; Университет машиностроения (МАМИ), каф. «Автоматика и процессы управления» - М., 2014.
2. Поповкин А.В., Сидорова М.Н., Туманов А.А., Валамов Д. О. Методические указания по выполнению лабораторных работ по дисциплине «Основы программирования и алгоритмизации в машиностроении». Часть 2. CD для студ., обуч. по направ. 220400.62 ... и 220700.62 ... под ред. В.И. Харитонова; Университет машиностроения (МАМИ), каф. «Автоматика и процессы управления» - М., 2014.

9. Методические рекомендации для самостоятельной работы студентов

Самостоятельная работа является одним из видов учебных занятий. Цель самостоятельной работы – практическое усвоение студентами вопросов алгоритмизации и разработки прикладных программ, рассматриваемых в процессе изучения дисциплины.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия

Задачи самостоятельной работы студента:

- развитие навыков самостоятельной учебной работы;
- освоение содержания дисциплины;
- углубление содержания и осознание основных понятий дисциплины;
- использование материала, собранного и полученного в ходе самостоятельных занятий для эффективной подготовки к экзамену.

Виды внеаудиторной самостоятельной работы:

- самостоятельное изучение отдельных тем дисциплины;
- подготовка к лекционным занятиям;
- подготовка к лабораторным работам;
- выполнение домашних заданий по закреплению тем;

Для выполнения любого вида самостоятельной работы необходимо пройти следующие этапы:

- определение цели самостоятельной работы;
- конкретизация познавательной задачи;
- самооценка готовности к самостоятельной работе;
- выбор адекватного способа действия, ведущего к решению задачи;
- планирование работы (самостоятельной или с помощью преподавателя) над заданием;
- осуществление в процессе выполнения самостоятельной работы самоконтроля (промежуточного и конечного) результатов работы и корректировка выполнения работы;
- рефлексия;
- презентация работы.

Вопросы, выносимые на самостоятельную работу (ОПК-2, ОПК-4)

1. Жизненный цикл программного обеспечения (ЖЦ ПО).
2. Основные и вспомогательные процессы.
3. Структура современных моделей ЖЦ ПО.
4. Стандарты ISO ЖЦ ПО.
5. Группа стандартов IEEE. ЖЦ ПО.

6. Группа стандартов СММ ЖЦ ПО.
7. Классификация и назначение документации на программное обеспечение.
8. Документирование в процессе разработки программного обеспечения.
9. Стандартизация документирования программ и данных.
10. Стандартизация управления конфигурацией программного обеспечения.
11. Реинжиниринг программного обеспечения.
12. Задачи стандартизации программных средств.
13. CASE-модель жизненного цикла программных средств.
14. Концептуальные основы CASE-технологии.
15. Механизмы защиты программных продуктов.
16. Принципы и методы тестирования.
17. Проектирование тестовых наборов данных.
18. Критерии завершенности тестирования.
19. Обработка результатов тестирования и отладки программ.
20. Технология облачных вычислений.
21. Экстремальное программирование.
22. Качество программного обеспечения и методы его контроля.
23. Эффективность языков программирования.
24. Международные стандарты языков программирования.
25. Инструментальные среды программирования фирмы BorlandInternationalInc.
26. Инструментальные среды программирования фирмы фирмы Microsoft).

10. Методические рекомендации для преподавателя

Основное внимание при изучении дисциплины «Основы программирования и алгоритмизации в машиностроении» следует уделять изучению структур данных и алгоритмов их обработки. Необходимо обеспечить понимание студентами технологий разработки программ.

Для активизации учебного процесса при изучении дисциплины эффективно применение презентаций по различным темам лекций.

Теоретическое изучение основных вопросов разделов дисциплины должно завершаться практической работой.

Для проведения занятий по дисциплине используются средства обучения:

- учебники, информационные ресурсы Интернета;
- справочные материалы и нормативно-техническая документация;
- методические указания для выполнения лабораторных работ.

Приложение 1 к

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Направление подготовки: 15.03.01 «Машиностроение»

Профиль: «Высокоэффективные технологические процессы и оборудование»

Форма обучения: заочная

Вид профессиональной деятельности (в соответствии с ФГОС ВО):

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ

«Основы программирования и алгоритмизации в машиностроении»

Состав: 1. Паспорт фонда оценочных средств

2. Описание оценочных средств

Составитель:

к.т.н., доц. Кузнецов А.В.

Москва, 2022 год

Таблица 1

ПОКАЗАТЕЛИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Основы программирования и алгоритмизации в машиностроении					
ФГОС ВО 15.03.01 Машиностроение					
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие **компетенции**:					
КОМПЕТЕНЦИИ		**Перечень компонентов**	**Технология формирования компетенции**	**Форма оценочного средства****	**Степени уровней освоения компетенций**
ИН-ДЕКС	**ФОРМУЛИРОВ-КА**				
ОПК-2	Способен применять основные методы, способы и средства получения, хранения, переработки информации при решении задач профессиональной деятельности	**знать:** • структуры данных, используемые для представления типовых информационных объектов **уметь:** • разрабатывать алгоритмы решения задач применительно к процессам и системам управления техническими (технологическими) объектами **владеть:** • основными технологиями программирования	лекция, самостоятельная работа, лабораторные работы	Э, ЛР, УО, Т, ДС	**Базовый уровень:** воспроизводство полученных знаний в ходе текущего контроля; умение решать типовые задачи, принимать профессиональные и управленческие решения по известным алгоритмам, правилам и методикам **Повышенный уровень:** практическое применение полученных знаний в процессе выполнения лабораторных работ и курсовой работы; готовность решать практические задачи повышенной сложности, нетиповые задачи, принимать профессиональные и управленческие решения в условиях неполной определенности, при недостаточном документальном, нормативном и методическом

ОПК-4	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.	**знать:** • типовые алгоритмы обработки данных **уметь:** • использовать язык программирования для создания программы **владеть:** навыками чтения и составления технической документации на программный продукт	лекция, самостоятельная работа, лабораторные	Э, ЛР, УО, Т, ДС	**Базовый уровень:** воспроизводство полученных знаний в ходе текущего контроля; умение решать типовые задачи, принимать профессиональные и управленческие решения по известным алгоритмам, правилам и методикам **Повышенный уровень:** практическое применение полученных знаний в процессе выполнения лабораторных работ и курсовой работы; готовность решать практические задачи повышенной сложности, нетиповые задачи, принимать профессиональные и управленческие решения в условиях неполной определенности, при недостаточном документальном, нормативном и методическом обеспечении

**- Сокращения форм оценочных средств см. в приложении 2 к РП.

Варианты экзаменационных билетов

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет машиностроения
Кафедра «Автоматика и управление»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № ZZ
для проведения экзамена по дисциплине
«Основы программирования и алгоритмизации в машиностроении»
Курс 1, семестр 2

---

1. Защита программного обеспечения.
2. Стандартизация алгоритмов.
3. Алгоритм цикла с управляющей переменной.

Билет рассмотрен и утвержден на заседании кафедры.

Протокол от « ДД » ММ 20XX г. № YY

Зав. каф. «Автоматика и управление» _____ / _____/

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет машиностроения
Кафедра «Автоматика и управление»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № ZZ
для проведения экзамена по дисциплине
«Основы программирования и алгоритмизации в машиностроении»
Курс 2, семестр 3

---

1. Оценка сложности программ.
2. Указатели в C/C++.
3. Программирование циклических алгоритмов.

Билет рассмотрен и утвержден на заседании кафедры.
Протокол от « ДД » ММ 20XX г. № YY

Зав. каф. «Автоматика и управление» _____ / _____/

Перечень вопросов на зачет (2 семестр)
(код компетенции ОПК-2, ОПК-4)

1. Этапы решения задач на ЭВМ.
2. Определение алгоритма. Свойства алгоритма.
3. Способы представления алгоритмов. Примеры.
4. Запись алгоритмов блок-схемами. Основные элементы блок-схем.
5. Стандартизация алгоритмов.
6. Основные алгоритмические структуры. Примеры.
7. Алгоритмы с ветвлением. Пример алгоритма.
8. Алгоритм цикла с предусловием. Пример алгоритма.
9. Алгоритм цикла с постусловием. Пример алгоритма.
10. Алгоритм цикла с управляющей переменной. Пример алгоритма.
11. Методы разработки и анализа алгоритмов.
12. Языки программирования: классификация и основные характеристики.
13. Состав алгоритмического языка (символы, лексемы, выражения, операторы, взаимосвязь элементов).
14. Системы программирования. Понятие, классификация, примеры.
15. Схема обработки прикладных программ в среде абстрактной системы программирования компилируемого типа (ввод, трансляция, компоновка, выполнение).
16. Процесс трансляции и компиляции.
17. Интегрированная среда программирования (понятие, структура, примеры).
18. Структура программы.
19. Обработка массивов.
20. Программирование ветвящихся алгоритмов.
21. Программирование циклических алгоритмов.
22. Базовые типы данных языка C/C++.
23. Форматы вывода данных языка C/C++.
24. Указатели в C/C++.
25. Структуры в C/C++.

Перечень вопросов на экзамен (3 семестр)
(код компетенции ОПК-3, ОПК-4)

1. Простые структуры и типы данных.
2. Линейные статические структуры данных.
3. Временные структуры данных
4. Составные линейные типы данных.
5. Составные нелинейные типы данных.
6. Внешние структуры данных.
7. Методы сортировки данных.
8. Методы сортировки массивов.
9. Одномерные массивы: задачи поиска, замены и перестановок элементов массива.
10. Двумерные массивы: задачи поиска, замены и суммирования элементов двумерного массива.
11. Алгоритмы поиска в линейных структурах.
12. Алгоритм перебора с возвратом.
13. Рекурсивные алгоритмы.
14. Защита программного обеспечения.
15. Современные алгоритмы обработки данных.
16. Параллельные алгоритмы.
17. Параллельный поиск.

18. Параллельные алгоритмы на графах.
19. Параллельная сортировка.
20. Оценка сложности программ. Критерии оценки сложности программ
21. Отладка программ.
22. Оценка алгоритмов сортировки.
23. Оценка сложности программ. Пространственная сложность.
24. Оценка сложности программ. Временная сложность.
25. Оценка сложности программ. Асимптотическая оценка сложности.
26. Классификация ошибок по этапу обработки программ.
27. Тестирование программ.
28. Нисходящая и восходящая разработка программного обеспечения.
29. Технология визуального программирования.
30. Уровни языков программирования.
31. Процедурная парадигма программирования.
32. Объектно-ориентированная парадигма программирования
33. Технология визуального программирования.
34. Технологии для создания и работы интернет приложений.
35. Технология структурного программирования.

Перечень лабораторных работ

№ п/п	Наименование	Оснащение	Кол-во часов
1	Программирование линейных алгоритмов на языке С.	Персональные компьютеры Системные требования к оборудованию: • Процессор с тактовой частотой 1,6 ГГц или большей • ОЗУ объемом 1 ГБ (1,5 ГБ для работы на виртуальной машине) • 5 ГБ доступного пространства на жестком диске • Жесткий диск с частотой вращения 5400 об/мин • Видеокарта с поддержкой DirectX 9 и разрешения экрана не менее 1024 x 768 Программное обеспечение Интегрированная среда разработки Microsoft Visual Studio 2013 и выше.	0,5
2	Программирование ветвящихся и циклических алгоритмов на языке С		0,5
3	Массивы. Применение массивов для последовательного хранения нескольких значений на языке С		1
4	Методы сортировки массивов		1
5	Указатели		1
6	Работа с функциями		1
7	Работа с файлами. Чтение из файла. Запись в файл.		1
8	Работа с текстовыми и бинарными файлами.		1
9	Символьные переменные и строки в С.		1
10	Преобразование и форматирование строк в С.		1
11	Применение структур в С.		1
12	Переменные и типы данных в С++. Функции ввода-вывода cin, cout.		0,5
13	Применение массивов в С++.		0,5
14	Функции в С++.		1
15	Применение указателей в С++.		1
16	Разработка динамических массивов в С++.		1
17	Разработка программы с конструкторами и деструкторами.		1
18	Работа с файлами в потоковом режиме.		1

Примерный перечень вопросов для устного опроса

1. Как создать новое консольное приложение?
2. Какое расширение имеют программы на C++?
3. Что такое компиляция?
4. Как скомпилировать программу?
5. Почему среда разработки Microsoft Visual Studio называется интегрированной?
6. Что такое отладка?
7. Для чего применяется ключевое слово void в C++?
8. Что такое main() в C++?
9. С помощью чего показывается начало и конец тела функции или класса в C++?
10. Какова роль комментариев?
11. Какие два вида комментариев существуют?
12. Чувствителен ли к регистру язык программирования C++?
13. Можно ли выводить на устройство вывода текст на кириллице при написании программ на языке программирования C++ в среде Microsoft Visual Studio?
14. Что такое блок комментариев и когда он применяется?
15. Какова структура программ на C++ и что такое хороший стиль?
16. Что такое директива препроцессора и каково ее функциональное предназначение?
17. Что такое условие?
18. К чему прибегают, когда возникает необходимость повторного использования операторов?
19. Какие типы циклов существуют?
20. Как работает цикл типа while?
21. В чем отличие принципа работы цикла типа do…while от while?
22. В каких случаях применяют цикл типа do…while?
23. Что такое одномерный массив?
24. Как нумеруются элементы массива?
25. Как размещается массив в пямяти компьютера?
26. С помощью чего осуществляется доступ к каждому элементу массива?
27. Как задается размерность массива?
28. Как задается размерность массива при помощи константы?
29. Как производится вычисление суммы элементов массива?
30. Как работает сортировка массива простым выбором?
31. Как работает метод пузырьковой сортировки?
32. Что такое двумерный массив?
33. Что такое указатель?
34. Как происходит побайтовое смещение при использовании указателей?
35. Что такое функция?
36. Как называется информация, передаваемая в функцию для обработки?
37. Обязательно ли описывать (объявлять) функцию перед её вызовом?
38. Что такое хороший стиль программирования?
39. Как осуществляется передача параметров в функцию?
40. Как выполняется передача параметров по значению?
41. Как выполняется передача параметров по адресу?
42. Какими двумя способами выполняются операции распределения памяти в C и C++?
43. Какие существуют два способа хранения информации в оперативной памяти ПК?
44. Как осуществляется работа с динамической памятью в C++?
45. Как осуществляется проверка на выделение памяти?
46. Каким образом реализуется работа с двумерными динамическими массивами в C++?

Примерный перечень тем докладов (презентаций)

47. Особенности современных методологий и технологий разработки ПС.
48. Технология сборочного программирования.
49. Направления развития и модели концепции открытых систем.
50. Особенности и возможности Internet-технологии.
51. Особенности и возможности Intranet-технологии.
52. Инструментальные средства создания Intranet-приложений.
53. Технологии параллельного программирования.
54. Компонентные технологии и разработка распределенного ПО.
55. Руководство программным проектом.
56. Технологии коллективной разработки ПО.
57. Рефакторинг программного обеспечения.
58. Быстрая разработка программного обеспечения.
59. Управление рисками проектов программных средств.
60. Управление качеством проектов программных средств.
61. Организация коллективной работы по созданию проектов.
62. Особенности технологии производства программных продуктов.
63. Инструментарий технологии программирования.
64. Промышленные технологии проектирования ПО.
65. Направления развития технологий программирования.
66. Методология гибкой разработки программного обеспечения (Agile Development Method).
67. Разработка программного обеспечения с использованием экстремального программирования (Extreme Programming).
68. Разработка программного обеспечения с использованием методологии SCRUM.
69. Разработка прикладного программного обеспечения с использованием технологии TDD (test-driven development).
70. Применение методологии рефакторинга (Refactoring) при разработке прикладного программного обеспечения.
71. Применение методологии Pattern matching при разработке прикладного программного обеспечения.

Образцы вопросов из фонда тестовых заданий

1. **Совокупность правил образования цепочек символов, образующих идентификаторы, операторы, операции и другие лексические компоненты языка – это**
   A. Синтаксис;
   B. Лексика;
   C. Алфавит;
   D. Семантика.

2.Любая информация, представленная в формализованном виде и пригодная для обработки алгоритмом – это
   A. Константы;
   B. Данные;
   C. Переменные.

3.Какой диапазон значений имеет тип int для 32-разрядных вычислительных систем:
   A. от 0 до 255
   B. от -32768 до 32767
   C. от 0 до 65535
   D. от 0 до 4 294 967 295

4.Какой размер в байтах имеет переменная вещественного типа float
   A. 2

B. 4
C. 8
D. 10

5.Дан массив int L[3][3] = { { 2, 3, 4 }, { 3, 4, 8 }, { 1, 0, 9 } };. Чему будет равно значение элемента этого массива L[1][2]

A. 2
B. 3
C. 4
D. 8

6.Для объявления размера массива должна использоваться_____, потому что она делает программу более масштабируемой

A. переменная
B. константа
C. именованная константа
D. символ

7.Процесс упорядоченного размещения элементов в массиве называется_____

A. Сортировка
B. Поиск
C. Проверка
D. Изменение

8.Какой тип данных отсутствует в Си в отличие от большинства других языков:

A. Real
B. Integer
C. String
D. Char

9.Идентификатор - это:

A. последовательность латинских букв, цифр и символа «_», начинающаяся с буквы или символа «_»
B. неизменяемые объекты языка (константы)
C. последовательность латинских и русских букв
D. способ кодирования, допустимые преобразования над значением данной переменной

10.Этап, занимающий наибольшее время, при разработке программы:

A. тестирование
B. сопровождение
C. проектирование
D. программирование
E. формулировка требований

11.Оценка сложности линейного алгоритма равна

A. O(1)
B. O(n)
C. $O(N^2)$
D. O(n!)

12.Авторское право на программный продукт действует в течение

A. всей жизни автора и 50 лет после его смерти
B. всей жизни автора и 100 лет после его смерти
C. всей жизни автора

13.Что выполняется раньше:

A. компиляция
B. отладка
C. компоновка
D. тестирование

14.Способы оценки качества:

A. сравнение с аналогами
B. наличие документации
C. оптимизация программы
D. структурирование алгоритма

15.Процесс упорядоченного размещения элементов в массиве называется_____

A. Сортировка
B. Поиск
C. Проверка
D. Изменение

16.Выражения - это:

A. конструкции, включающие константы (литералы), переменные, знаки операций, скобки для управления порядком выполнения операций, обращения к функциям
B. основные строительные блоки программы; в языке Си++ указанием на наличие выражения служит символ «точка с запятой», стоящий в конце него
C. набор символов и операций
D. операторы, выполняющие определенные действия с переменными

17.Какой тип данных отсутствует в Си в отличие от большинства других языков:

A. Real
B. Integer
C. String
D. Char

18.Идентификатор - это:

A. последовательность латинских букв, цифр и символа «_», начинающаяся с буквы или символа «_»
B. неизменяемые объекты языка (константы)
C. последовательность латинских и русских букв
D. способ кодирования, допустимые преобразования над значением данной переменной

19.Интерфейс

A. представляет собой набор функций, которые реализуют класс
B. представляет собой описание набора функций
C. представляет собой описание набора функций, которые реализуют класс
D. представляет собой описание набора функций

20.Этап, занимающий наибольшее время, при разработке программы:

A. тестирование
B. сопровождение
C. проектирование
D. программирование
E. формулировка требований

Приложение 2
к рабочей программе

Перечень оценочных средств по дисциплине
«Основы программирования и алгоритмизации в машиностроении»

№ ОС	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1.	Лабораторные работы (ЛР)	Оценка способности студента применить полученные ранее знания для проведения анализа, опыта, эксперимента и выполнения последующих расчетов, а также составления	Перечень лабораторных работ и их оснащение
2.	Устный опрос, собеседование, (УО)	Средство контроля, организованное как специальная беседа педагогического работника с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.	Вопросы по темам/разделам дисциплины
3.	Тест (Т)	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
4.	Устный опрос (Э -экзамен)	Диалог преподавателя со студентом, цель которого – систематизация и уточнение имеющихся у студента знаний, проверка его индивидуальных возможностей усвоения материала	Комплект экзаменационных билетов

Приложение

Структура и содержание дисциплины «Основы программирования и алгоритмизации в машиностроении»
15.03.01 Машиностроение
(бакалавриат)

n/n	Раздел	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов, и трудоемкость в часах					Виды самостоятельной работы студентов					Формы аттестации	
				Л	П/С	Лаб	СРС	КСР	К.Р.	К.П.	УО	ДС	Т	Э	З
	Второй семестр														
1.	**Программирование как составляющая процесса разработки программного обеспечения** Общая характеристика процесса разработки программного обеспечения. Ключевые этапы разработки. Модели процесса разработки программного обеспечения. Методологии разработки программного обеспечения. Защита программного обеспечения.	2										+			
2.	**Алгоритмические основы программирования** Понятие алгоритма. Исполнители алгоритмов. Свойства алгоритмов. Типы алгоритмов. Формы представления алгоритмов. Методы разработки и анализа алгоритмов	2									+				

3.	**Системы программирования** Понятие системы программирования. Программные компоненты системы программирования. Редактор текста. Трансляторы. Компоновщик. Отладчик. Библиотеки подпрограмм. Языки программирования. Уровни языков. Поколения языков. Интегрированные среды разработки.	2										+			
4.	**Язык программирования С/С++** Элементы языка С. Базовые типы данных. Операции языка С. Управляющие операторы. Операторы форматного ввода и вывода. Структура программы. Математические функции. Форматы вывода данных. Особенности арифметических операций. Программирование ветвящихся и циклических алгоритмов.	2									+				
5.	**Язык программирования С/С++** Массивы. Обработка массивов. Указатели. Функции. Файлы. Символьные переменные и строки. Преобразование строк. Форматирование строк. Структуры.	2									+				
6.	**Язык программирования С/С++** Объектно-ориентированное программирование. Классы. Дружественные функции и классы. Статические элементы и функции. Константные функции. Перегрузка операций. Шаблоны	2										+			
7.	Программирование линейных алгоритмов на языке С.	2									+				

8.	Программирование ветвящихся и циклических алгоритмов на языке С	2									+				
9.	Массивы. Применение массивов для последовательного хранения нескольких значений на языке С	2										+			
10.	Методы сортировки массивов	2									+				
11.	Указатели	2									+				
12.	Работа с функциями	2										+			
13.	Работа с файлами. Чтение из файла. Запись в файл.	2									+				
14.	Работа с текстовыми и бинарными файлами.	2									+				
15.	Символьные переменные и строки в С.	2										+			
	Форма аттестации												+		**З**
	Всего часов по дисциплине в первом семестре	16		6		10	60								
	Третий семестр														
1.	**Структуры и алгоритмы обработки данных** Понятие о структурах данных. Простые структуры и типы данных.	3									+				
2.	**Структуры и алгоритмы обработки данных** Составные линейные типы данных. Массив. Очередь. Стек. Дек. Линейные списки.	3									+				

3.	**Структуры и алгоритмы обработки данных** Составные нелинейные типы данных. Древовидные структуры данных. Элементы теории графов.	3									+				
4.	**Структуры и алгоритмы обработки данных** Внешние структуры данных. Файлы. Базы данных.	3									+				
5.	**Структуры и алгоритмы обработки данных** Алгоритмы обработки данных. Поиск. Сортировка.	3									+				
6.	**Структуры и алгоритмы обработки данных** Параллельные алгоритмы. Параллельный поиск. Параллельная сортировка. Параллельные алгоритмы на графах. Современные алгоритмы обработки данных.	3									+				
7.	**Технологии программирования** Технология структурного программирования. Технология объектно-ориентированного программирования. Технология визуального программирования. Технологии для создания и работы интернет приложений.	3									+				

8.	**Оценка сложности алгоритмов и программ** Сравнение алгоритмов. Критерии оценки сложности. Виды сложностей. Пространственная сложность. Временная сложность. Асимптотическая оценка сложности	3									+				
9.	Преобразование и форматирование строк в С.	3									+				
10.	Применение структур в С.	3									+				
11.	Переменные и типы данных в С++. Функции ввода-вывода cin, cout.	3									+				
12.	Применение массивов в С++.	3									+				
13.	Функции в С++.	3									+				
14.	Применение указателей в С++.	3									+				
15.	Разработка динамических массивов в С++.	3									+				
16.	Разработка программы с конструкторами и деструкторами.	3									+				
17.	Работа с файлами в потоковом режиме.	3									+				
	Форма аттестации								+				+	**Э**	
	Всего часов по дисциплине в третьем семестре	8		2		6	60								
	Всего часов по дисциплине	144		8		16	120								