

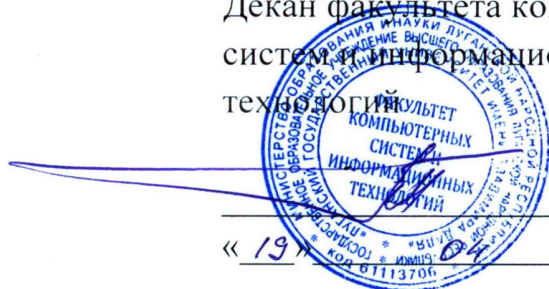
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Факультет компьютерных систем и информационных технологий

Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Декан факультета компьютерных
систем и информационных
технологий



Кочевский А.А.

« 19 » 04 2023 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине

«Программирование мехатронных и робототехнических систем»

15.03.06 Мехатроника и робототехника

«Компьютерные технологии управления в робототехнике и мехатронике»

Разработчик:

старший преподаватель

Синепольский Д.О.

ФОС рассмотрен и одобрен на заседании кафедры информационных и управляющих систем от «18» апреля 2023 г., протокол № 15

Заведующий кафедрой

информационных и управляющих систем

Горбунов А.И.

Луганск 2023 г.

Паспорт
фонда оценочных средств по учебной дисциплине
«Программирование мехатронных и робототехнических систем»

Перечень компетенций (элементов компетенций),
формируемых в результате освоения учебной дисциплины

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые темы учебной дисциплины	Этапы формирования (семестр изучения)
1	ОПК-2.1	Знает основные методы решения задач с использованием программных средств;	Тема 19. Введение в системное программирование Тема 28. Обобщение алгоритмов	начальный (6)
2	ОПК-2.2	Умеет алгоритмизировать решения задач и реализует алгоритмы с использованием программных средств;	Тема 8. Циклы Тема 9. Передача управления Тема 10. Функции	начальный (6)
3	ОПК-2.3	Владеет знаниями в области информационных технологий для поиска, хранения, обработки, анализа и представления информации.	Тема 29. Индексация Тема 30. Простые структуры данных Тема 31. Списки Тема 32. Улучшенные списки Тема 33. Хеширование Тема 34. Деревья поиска Тема 35. Построение и модификация дерева	начальный (6)
	ОПК-14.1	Знает базовые алгоритмы, подходы к написанию ПО;	Тема 20. Постановка задачи сортировки и алгоритм обмена Тема 21. Алгоритмы выбора и вставок Тема 22. Сортировка Шелла	начальный (6)
4	ОПК-14.2	Умеет использовать специализированные библиотеки;	Тема 12. Потоки ввода-вывода Тема 28. Обобщение алгоритмов	начальный (6)

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые темы учебной дисциплины	Этапы формирования (семестр изучения)
	ОПК-14.3	Владеет навыками использования высокоуровневых ЯП.	Тема 1. Программа, формат и подготовка к выполнению Тема 2. Константы и переменные Тема 3. Массивы и строки Тема 4. Средства ввода-вывода Тема 5. Операции Тема 6. Условные вычисления Тема 7. Дополнительные операции языка C++	начальный (5)
			Тема 8. Циклы Тема 9. Передача управления Тема 10. Функции Тема 11. Передача параметров Тема 13. Массивы Тема 14. Многомерные массивы Тема 15. Указатели Тема 16. Указатели и массивы Тема 17. Структуры Тема 18. Массивы структур	начальный (6)
5	ПК-3.2	Способен к написанию программного кода для изделий мехатроники и робототехники с использованием языков программирования, определения и манипулирования данными;	Тема 23. Быстрая сортировка Тема 24. Поиск порядковых статистик Тема 25. Структура данных Куча (Пирамида) Тема 26. Очереди с приоритетом и пирамидальная сортировка Тема 27. Поиск в структурах с произвольным доступом	начальный (6)

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые темы учебной дисциплины	Этапы формирования (семестр изучения)
	ПК-3.3	Проверяет работоспособность и рефакторинг кода программного обеспечения для изделий мехатроники и робототехники.	Тема 19. Введение в системное программирование	начальный (6)

**Показатели и критерии оценивания компетенций,
описание шкал оценивания**

№ п/п	Код контролируемой компетенции	Показатель оценивания (знания, умения, навыки)	Контролируемые темы учебной дисциплины	Наименование оценочного средства
1	ОПК-2.1	Знать: - современную классификацию программного обеспечения; - математические методы, которые используются для оценки эффективности программ; - общие и специализированные алгоритмы, которые используются для упорядочения и поиска данных и их свойства; - методы и структуры, которые используются для эффективного хранения и обработки данных во внутренней памяти; - организацию эффективного хранения и доступ к данным расположенным во внешней памяти; - использование этих алгоритмов и методов в системном программном обеспечении. Уметь: - применять специализированные алгоритмы и структуры,	Тема 19. Тема 28.	Практические работы, контрольные работы, экзамен

№ п/п	Код контролируемой компетенции	Показатель оценивания (знания, умения, навыки)	Контролируемые темы учебной дисциплины	Наименование оценочного средства
		ориентированные на специальные классы задач.		
2	ОПК-2.2	Знать: - основные конструкции процедурного программирования - разбиение программы на функции и организацию связи между ними - систему типов и методы ее расширения, включая пользовательские типы	Тема 8. Тема 9. Тема 10.	Практические работы, контрольные работы, экзамен
3	ОПК-2.3	Знать: - общие и специализированные алгоритмы, которые используются для упорядочения и поиска данных и их свойства; - методы и структуры, которые используются для эффективного хранения и обработки данных во внутренней памяти;	Тема 29. Тема 30. Тема 31. Тема 32. Тема 33. Тема 34. Тема 35.	Практические работы, контрольные работы, экзамен
4	ОПК-14.1	Уметь: - определять оптимальные методы и структуры для представления данных; - кодировать стандартные алгоритмов упорядочения и поиска данных - кодировать алгоритмы для работы с динамическими структурами данных и данными, расположенными во внешней памяти;	Тема 20. Тема 21. Тема 22.	Практические работы, контрольные работы, экзамен

№ п/п	Код контролируемой компетенции	Показатель оценивания (знания, умения, навыки)	Контролируемые темы учебной дисциплины	Наименование оценочного средства
5	ОПК-14.2	Знать: - состав и назначение стандартной библиотеки языка С и С++, включая операции со строками и массивами данными во внешних файлах	Тема 12. Тема 28.	Практические работы, контрольные работы, экзамен
6	ОПК-14.3	Знать: - синтаксис и семантику языка программирования С++, включая - основные конструкции процедурного программирования - разбиение программы на функции и организацию связи между ними - систему типов и методы ее расширения, включая пользовательские типы - другие парадигмы программирования, реализованные с языка С++, и области их применения	Тема 1. Тема 2. Тема 3. Тема 4. Тема 5. Тема 6. Тема 7. Тема 8. Тема 9. Тема 10. Тема 11. Тема 13. Тема 14. Тема 15. Тема 16. Тема 17. Тема 18.	Практические работы, контрольные работы, экзамен
7	ПК-3.2	Уметь: - разрабатывать новое программное обеспечение на языке С++, реализуя заданные алгоритмы и применяя их к решению конкретных задач Владеть: - навыками применения интегрированных инструментальных средств для разработки и отладки программного обеспечения - навыками применения инструментов командной строки для автоматизации процесса сборки программного обеспечения	Тема 23. Тема 24. Тема 25. Тема 26. Тема 27.	Практические работы, контрольные работы, экзамен

№ п/п	Код контролируемой компетенции	Показатель оценивания (знания, умения, навыки)	Контролируемые темы учебной дисциплины	Наименование оценочного средства
8	ПК-3.3	Уметь: - читать и анализировать программу на C++ - вносить изменения и компилировать существующий исходный код на языке C++ с целью настройки программного обеспечения под изменившиеся или уточненные требования - находить и исправлять логические ошибки в коде с учетом особенностей языка C++ - читать и анализировать программу на C++ - формировать тестовых наборы данных; Владеть: - навыками применения интегрированных инструментальных средств для разработки и отладки программного обеспечения - навыками применения инструментов командной строки для автоматизации процесса сборки программного обеспечения - математическими методами оценки эффективности алгоритмов, и идентификации “узких мест” программного обеспечения;	Тема 19.	Практические работы, контрольные работы, экзамен

**Фонды оценочных средств по дисциплине
«Программирование мехатронных и робототехнических систем»**

Вопросы для защиты практических работ

**Практическое занятие № 1
Вычисление выражений**

Цель работы: Научиться программировать вычисление математических выражений с использованием операций, определенных в языке, и функций стандартных библиотек математики и ввода-вывода

Вопросы для защиты:

- 1) В каком заголовочном файле объявлены функции `printf()` и `scanf()`?
- 2) В каком заголовочном файле объявлены стандартные средства потокового ввода вывода C++?
- 3) Как проверить, что функция `scanf()` корректно преобразовала входные данные?
- 4) В каком заголовочном файле объявлены функции `fabs()` и `cos()`?
- 5) Чем отличаются функции `abs()` и `fabs()`?
- 6) Каким символом формата задается позиция в формате `double` для функции `scanf()`? В чем отличие от `printf()`?
- 7) Как задать константу в формате с плавающей точкой, если дробная часть равна нулю?
- 8) Как убедиться, что выражение будет вычисляться по правилам арифметики с плавающей точкой, если в него входят целочисленные элементы?

**Практическое занятие № 2
Операторы условного перехода**

Цель работы: Научиться программировать ветвление с применением условного оператора и условной функции.

Вопросы для защиты

- 1) В каком случае выражение в C считается истинным?
- 2) В конструкции `if .. if .. else` к какому из двух операторов `if` относится оператор `else`? Как это изменить?
- 3) Является ли последовательность `else if` отдельной конструкцией языка C?
- 4) Допустимо ли выражение `(a ? b ? c : d : e)`? Чему будет равно его значение, если `a = b = 1`?
- 5) Если в ветке оператора `if – else` необходимо вычислить несколько выражений, их нужно заключить в составной оператор. Есть ли аналогичное средство для тернарной операции?
- 6) Упростите выражение в операторе `if (a == b ? 0 : 1) ...`

Практическое занятие № 3

Множественный выбор

Цель работы: Научиться использовать логические выражения в связке с условным оператором, перечисления и оператор множественного выбора.

Вопросы для защиты

- 1) Назовите отличия операций `&&` и `||` от `&` и `|`.
- 2) Что такое «вычисление логических выражений по короткой схеме»?
- 3) Должен ли оператор `switch` включать ключевое слово `default`?
- 4) Должен ли случай `default` быть последним в операторе `switch`?
- 5) В чем отличие `enum {...} x;` от `typedef enum {...} x;`.

Практическое занятие № 4

Операторы цикла

Цель работы: Научиться программировать циклические вычисления с применением операторов цикла с пред- и постусловием.

Вопросы для защиты

- 1) Назовите цикл с пост-условием? Какое минимальное количество раз выполняется данный цикл?
- 2) Куда передают управление операторы `break` и `return`, примененные в теле цикла?
- 3) К чему приводит выполнение оператора `continue` в теле цикла?
- 4) В конструкции `switch(...)` `while(...)` `if (...)` `break` к какому оператору относится `break`?
- 5) В конструкции `while(...)` `switch(...)` `if (...)` `break; else continue;` к какому оператору относится `break`?
- 6) Куда передаст управление оператор `continue`, примененный внутри цикла `for(..;...;..)`?

Практическое занятие № 5

Массивы

Цель работы: научиться обрабатывать большие объемы однотипных данных представленные линейными либо многомерными массивами

Вопросы для защиты

- 1) Матрица задана как `int m[3][4]`. Назовите количество строк, столбцов и общее число элементов матрицы.
- 2) Какое из двух объявлений функций `int x(int p[][3])`, `y(int q[4][]);` является допустимым? Почему?
- 3) Если вы работаете с матрицей как с одномерным массивом, что в следующем выражении `m[r + r * z]` является индексом строки, индексом столбца, количеством столбцов?
- 4) Если параметр функции записан как `char z[300]`, будет ли выполняться равенство `sizeof(z) == 300` в теле функции? Почему?

Практическое занятие № 6

Обработка символьных строк

Цель работы: научиться обрабатывать символьные данные представленные потоками символов и символьными строками с использованием стандартных библиотек ввода-вывода и манипуляции символьными строками.

Вопросы для защиты

- 1) Каким образом компилятор сохраняет в памяти строковый литерал в языке С.
- 2) Какой тип имеет строковый литерал? Чему будет равно выражение `sizeof("abc")`? Почему?
- 3) Чем отличаются определения `char s1[] = "abc"`, `*s2 = "def"`;
- 4) Как определить при последовательном просмотре, что достигнут конец строки?
- 5) Сравните функции `fgets()` и `scanf()` с точки зрения ввода символьных последовательностей.
- 6) Как работают квадратные скобки в качестве спецификатора формата `scanf()`? Что задает следующий формат `scanf("%[^']", ...)`?
- 7) В чем заключается связь между функциями `setlocale()` и `toupper()`?

Практическое занятие № 7

Структуры

Цель работы: научиться обрабатывать большие объемы групп данных представленные сочетанием структур и линейных массивов

Вопросы для защиты

- 1) Пусть структура имеет поля `struct { ... char s1[10], *s2; ... }`. Что произойдет при копировании такой структуры «по умолчанию»?
- 2) Всегда ли размер структуры равен сумме размеров ее полей? Почему?
- 3) В заголовочных файлах, которые могут использоваться как в С, так и в С++ программе, можно встретить запись `typedef struct X { ... } X;`. Почему?
- 4) Сравните ключевые слова `struct` и `union`. В чем основное отличие?
- 5) Может ли применение `union` приводить к «неопределенному поведению»?
- 6) Всегда ли операции `(*x).y` и `x->y` эквивалентны в С++?
- 7) Как должна быть определена структура, чтобы в языке С выражение `x->y()` было допустимым?

Практическое занятие № 8

Функции

Цель работы: научиться структурировать решение задачи с помощью функций

Вопросы для защиты

- 1) Чем отличается объявление и определение функции? Сколько объявлений одной и той же функции может быть в программе? При каком условии?
- 2) Что произойдет, если в функцию, ожидающую вещественную величину, передать целочисленное значение, при условии что компилятору известно объявление функции?
- 3) Что произойдет в C++ и современном C при попытке вызвать функцию, ранее не объявленную? В чем отличие от доработанного C?
- 4) Сколько параметров будет передано в функцию в следующем выражении `f(a, (b, c))`? Почему?
- 5) Что такое перегрузка функций, и почему в стандартной библиотеке языка C есть функции `abs()`, `labs()` и `fabs()`?
- 6) Предложите несколько возможных путей вернуть из функции пару целочисленных значений.
- 7) Что такое «концевая рекурсия»? Чем ее можно заменить?
- 8) Можно ли функцию сделать параметром другой функции? Где это может понадобиться? Какую альтернативу предлагает C++?

Практическое занятие № 9

Файлы

Цель работы: Научиться программировать операции с данными, расположенными во внешних файлах и самими файлами.

Вопросы для защиты

- 1) Сравните последовательное сохранение полей структуры в файл с применением функции `fwrite()` к области памяти, хранящей структуру. Какой вариант предпочтителен и почему?
- 2) Объясните назначение второго параметра функции `fopen()`.
- 3) Какие факторы могут ограничивать количество файлов, которые могут быть одновременно открыты с помощью `fopen()`?
- 4) Сравните функции `ftell()` и `fgetpos()`.
- 5) При каких условиях можно определить текущий размер файла функциями `fseek()` и `ftell()`?
- 6) Что делает функция `fflush()`?
- 7) Что возвращает функция `feof()`? Если `feof()` вернула 0, может ли следующее обращение к `fgetc()` вернуть EOF?
- 8) Что такое `stdin`, `stdout` и `stderr`?

Практическое занятие №10

Сортировки с квадратичной сложностью

Цель: Получить навыки кодирования и отладки простых алгоритмов сортировки, подготовки тестовых данных и оценки сложности алгоритмов.

Вопросы для защиты

- 1) Сравните шейкер и пузырьковую сортировку.
- 2) Чем отличается сортировка Шелла от сортировки вставками?
- 3) Среди пузырьковой, выбором и сортировкой вставками, какой алгоритм может не сделать ни одной записи в массив? Является ли он лучшим?
- 4) Чему равно количество обменов в алгоритме сортировки выбором? Почему?
- 5) Почему внешний цикл в сортировках выбора и вставок на один шаг меньше общего числа элементов?
- 6) Почему сокращения шага обмена в сортировке Шелла в два раза не является лучшим решением? Какую последовательность шагов предложил Кнут?
- 7) Что такое верхняя и нижняя асимптотическая оценка сложности?

Практическое занятие №11

Эффективные методы сортировки

Цель: Познакомиться на практике с эффективными методами представления и обработки данных.

Вопросы для защиты

- 1) Почему существует четыре способа записать разбиение Ломутто? Назовите их.
- 2) Почему использование барьерных элементов может ускорить двустороннее разделение?
- 3) Почему в двустороннем разделение нужно обеспечить продвижение индексов после того как произошел обмен?
- 4) Почему в случае повторяющихся ключей двустороннее разделение обеспечит лучший результат в сравнении с односторонним?
- 5) Что такое оптимизация Седжевика для быстрой сортировки?
- 6) В чем заключается устранение концевой рекурсии?
- 7) Почему в задаче поиска медианы можно полностью исключить рекурсию?
- 8) Что такое структура данных «куча»? Что такое представляющий массив?
- 9) Почему в кучу данные всегда вставляются в конец представляющего массива, а извлекаются из его начала?
- 10) Сравните верхнюю асимптотическую оценку быстрой и пирамидальной сортировки. Какая из них на практике оказывается быстрее?

Практическое занятие №12

Поиск в линейных структурах

Цель: Познакомиться на практике с применением алгоритмов поиска для обработки последовательностей.

Вопросы для защиты

- 1) В чем заключается линейный поиск с барьерным элементом? Где располагается барьер? Можно ли применить метод барьерного элемента для двоичного поиска?
- 2) В чем заключается двоичный поиск? Какое требование налагается на данные?
- 3) Двоичный поиск можно организовать с тремя и с двумя сравнениями внутри цикла. В чем разница? Какой метод эффективней?
- 4) Если метод поиска реализован в виде подпрограммы, что лучше возвращать – объект, ссылку на объект, указатель на объект или индекс объекта? Сравните варианты.
- 5) Что такое несамоподобный образец в методе КМП? Почему самоподобные образцы усложняют алгоритм?
- 6) Как вычисляется префикс-функция в алгоритме КМП? Какой смысл вычисляемого значения?
- 7) Какими свойствами должны обладать входные данные, чтобы КМП имел большое преимущества над прямолинейным поиском?

Практическое занятие №13

Косвенное обращение к данным

Цель: Познакомиться на практике с реализацией алгоритмов использующих косвенное обращение к данным

Вопросы для защиты

- 1) Какие методы косвенного вызова доступны в используемом вами языке программирования?
- 2) В чем отличия базовой и прямой индексации? Какой тип данных в каждом из случаев используется в индексе?
- 3) Что такое индекс? Как обратиться к данным в порядке, заданном индексом?
- 4) Можно ли для поиска и сортировки использовать один способ косвенного сравнения данных?
- 5) За счет чего сортировка индексов может оказаться быстрее прямой сортировки данных?
- 6) Почему для индексов, хранящихся во внешних файлах, базовая адресация удобней, чем прямая?
- 7) Что такое модель параллельных массивов? Как она сочетается с базовой адресацией?

Практическое занятие №14

Применение структуры данных «стек» в алгоритмах с возвратом.

Цель: Познакомиться на практике с применением структуры типа «стек» для реализацией алгоритмов разбора и вычисления математических выражений.

Вопросы для защиты

- 1) Что такое префиксная и постфиксная форма записи выражений? Какое у них преимущество перед инфиксной с точки зрения интерпретации?
- 2) Опишите процесс вычисления выражения, записанного в постфиксной форме? Для хранения какого типа данных должен быть настроен стек?
- 3) Опишите алгоритм преобразования записи из инфиксной в постфиксную, для выражений не содержащих скобки. Можно ли скобки добавить только с использованием приоритетов операций?
- 4) Как расширить базовый алгоритм преобразования выражений, чтобы добавить поддержку скобок?
- 5) На какой тип данных должен быть настроен стек в алгоритме преобразования выражения в постфиксную форму?
- 6) Если выражение представить в виде дерева, то как можно из него получить все три формы записи?
- 7) Отличить в инфиксной записи бинарный и унарный минусы можно только по контексту. В чем заключается это отличие?

Практическое занятие №15

Словари на основе хеш-таблиц

Цель: Познакомиться на практике с реализацией словарей на основе хеш-таблиц с использованием методов цепочек и открытой адресации. Исследовать влияние выбора конфигурационных параметров на производительность.

Вопросы для защиты

- 1) Назовите основные операции, реализуемые структурой данных словарь (ассоциативный массив).
- 2) При выполнении каких условий возможно применении метода «прямой адресации»?
- 3) Что такое хеш-функция? Каково ее назначение при реализации словаря на основе хеш-таблицы? Что такое «коллизия»?
- 4) Какие два метода разрешения коллизий хеширования применяются в словарях на основе хеш-таблиц?
- 5) В чем заключается метод цепочек? Какая вспомогательная структура используется?
- 6) Опишите вычисление хеш-функции методом остатка. Как следует при этом выбирать размер хеш-таблицы?

- 7) В чем заключается метод вычисления хеш-функции умножением? Какое преимущество данного метода? Как следует выбирать размер таблиц?
- 8) Что делать, если ключ невозможно напрямую интерпретировать как целочисленные данные?
- 9) Опишите метод «открытой адресации». Где в этом случае сохраняются данные?

Практическое занятие №16

Словари на основе несбалансированных деревьев

Цель: Познакомиться на практике с реализацией словарей на основе двоичных несбалансированных деревьев поиска. Исследовать влияние порядка поступления ключей на производительность.

Вопросы для защиты

- 1) Что представляет собой двоичное дерево поиска? Как происходит вставка нового элемента?
- 2) Опишите аномалии, которые могут возникать в несбалансированных деревьях поиска.
- 3) Опишите процедуру обхода дерева? Зачем могут потребоваться ссылки на родительские элементы?
- 4) Как выполняется поиск по ключу в двоичном дереве поиска?
- 5) Опишите процедуру нерекурсивного обхода дерева.

Критерии и шкала оценивания по оценочному средству защита практических работ

Шкала оценивания	Критерий оценивания
5	Ответы на вопросы к защите практических работ даны на высоком уровне (правильные ответы даны на 90-100% вопросов)
4	Ответы на вопросы к защите практических работ даны на среднем уровне (правильные ответы даны на 75-89% вопросов)
3	Ответы на вопросы к защите практических работ даны на низком уровне (правильные ответы даны на 50-74% вопросов)
2	Ответы на вопросы к защите практических работ даны на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

Вопросы для контрольных работ

1. В каком заголовочном файле объявлены функции printf() и scanf()?
2. В каком заголовочном файле объявлены стандартные средства потокового ввода вывода C++?
3. Как проверить, что функция scanf() корректно преобразовала входные данные?
4. В каком заголовочном файле объявлены функции fabs() и cos()?

5. Чем отличаются функции `abs()` и `fabs()`?
6. Каким символом формата задается позиция в формате `double` для функции `scanf()`? В чем отличие от `printf()`?
7. Как задать константу в формате с плавающей точкой, если дробная часть равна нулю?
8. Как убедиться, что выражение будет вычисляться по правилам арифметики с плавающей точкой, если в него входят целочисленные элементы?
9. В каком случае выражение в C считается истинным?
10. В конструкции `if .. if .. else` к какому из двух операторов `if` относится оператор `else`? Как это изменить?
11. Является ли последовательность `else if` отдельной конструкцией языка C?
12. Допустимо ли выражение `(a ? b ? c : d : e)`? Чему будет равно его значение, если `a = b = 1`?
13. Если в ветке оператора `if – else` необходимо вычислить несколько выражений, их нужно заключить в составной оператор. Есть ли аналогичное средство для тернарной операции?
14. Упростите выражение в операторе `if (a == b ? 0 : 1) ...`
15. Назовите отличия операций `&&` и `||` от `&` и `|`.
16. Что такое «вычисление логических выражений по короткой схеме»?
17. Должен ли оператор `switch` включать ключевое слово `default`?
18. Должен ли случай `default` быть последним в операторе `switch`?
19. В чем отличие `enum {...} x;` от `typedef enum {...} x;`.
20. Назовите цикл с пост-условием? Какое минимальное количество раз выполняется данный цикл?
21. Куда передают управление операторы `break` и `return`, примененные в теле цикла?
22. К чему приводит выполнение оператора `continue` в теле цикла?
23. В конструкции `switch(...) while(...) if (...) break` к какому оператору относится `break`?
24. В конструкции `while(...) switch(...) if (...) break; else continue;` к какому оператору относится `break`?
25. Куда передаст управление оператор `continue`, примененный внутри цикла `for(;;;..)`?
26. Матрица задана как `int m[3][4]`. Назовите количество строк, столбцов и общее число элементов матрицы.
27. Какое из двух объявлений функций `int x(int p[][3]), y(int q[4][]);` является допустимым? Почему?
28. Если вы работаете с матрицей как с одномерным массивом, что в следующем выражении `m[r + r * z]` является индексом строки, индексом столбца, количеством столбцов?
29. Если параметр функции записан как `char z[300]`, будет ли выполняться равенство `sizeof(z) == 300` в теле функции? Почему?

30. Каким образом компилятор сохраняет в памяти строковый литерал в языке C.
31. Какой тип имеет строковый литерал? Чему будет равно выражение `sizeof("abc")`? Почему?
32. Чем отличаются определения `char s1[] = "abc"`, `*s2 = "def"`;
33. Как определить при последовательном просмотре, что достигнут конец строки?
34. Сравните функции `fgets()` и `scanf()` с точки зрения ввода символьных последовательностей.
35. Как работают квадратные скобки в качестве спецификатора формата `scanf()`? Что задает следующий формат `scanf("%[^']", ...)`?
36. В чем заключается связь между функциями `setlocale()` и `toupper()`?
37. Пусть структура имеет поля `struct { ... char s1[10], *s2; ... }`. Что произойдет при копировании такой структуры «по умолчанию»?
38. Всегда ли размер структуры равен сумме размеров ее полей? Почему?
39. В заголовочных файлах, которые могут использоваться как в C, так и в C++ программе, можно встретить запись `typedef struct X { ... } X;`. Почему?
40. Сравните ключевые слова `struct` и `union`. В чем основное отличие?
41. Может ли применение `union` приводить к «неопределенному поведению»?
42. Всегда ли операции `(*x).u` и `x->u` эквивалентны в C++?
43. Как должна быть определена структура, чтобы в языке C выражение `x->u()` было допустимым?
44. Чем отличается объявление и определение функции? Сколько объявлений одной и той же функции может быть в программе? При каком условии?
45. Что произойдет, если в функцию, ожидающую вещественную величину, передать целочисленное значение, при условии что компилятору известно объявление функции?
46. Что произойдет в C++ и современном C при попытке вызвать функцию, ранее не объявленную? В чем отличие от доработанного C?
47. Сколько параметров будет передано в функцию в следующем выражении `f(a, (b, c))`? Почему?
48. Что такое перегрузка функций, и почему в стандартной библиотеке языка C есть функции `abs()`, `labs()` и `fabs()`?
49. Предложите несколько возможных путей вернуть из функции пару целочисленных значений.
50. Что такое «концевая рекурсия»? Чем ее можно заменить?
51. Можно ли функцию сделать параметром другой функции? Где это может понадобиться? Какую альтернативу предлагает C++?
52. Сравните последовательное сохранение полей структуры в файл с применением функции `fwrite()` к области памяти, хранящей структуру. Какой вариант предпочтителен и почему?
53. Объясните назначение второго параметра функции `fopen()`.

54. Какие факторы могут ограничивать количество файлов, которые могут быть одновременно открыты с помощью `open()`?
55. Сравните функции `ftell()` и `fgetpos()`.
56. При каких условиях можно определить текущий размер файла функциями `fseek()` и `ftell()`?
57. Что делает функция `fflush()`?
58. Что возвращает функция `feof()`? Если `feof()` вернула 0, может ли следующее обращение к `fgetc()` вернуть EOF?
59. Что такое `stdin`, `stdout` и `stderr`?
60. Сравните шейкер и пузырьковую сортировку.
61. Чем отличается сортировка Шелла от сортировки вставками?
62. Среди пузырьковой, выбором и сортировкой вставками, какой алгоритм может не сделать ни одной записи в массив? Является ли он лучшим?
63. Чему равно количество обменов в алгоритме сортировки выбором? Почему?
64. Почему внешний цикл в сортировках выбора и вставок на один шаг меньше общего числа элементов?
65. Почему сокращения шага обмена в сортировке Шелла в два раза не является лучшим решением? Какую последовательность шагов предложил Кнут?
66. Что такое верхняя и нижняя асимптотическая оценка сложности?
67. Почему существует четыре способа записать разбиение Ломутто? Назовите их.
68. Почему использование барьерных элементов может ускорить двустороннее разделение?
69. Почему в двустороннем разделение нужно обеспечить продвижение индексов после того как произошел обмен?
70. Почему в случае повторяющихся ключей двустороннее разделение обеспечит лучший результат в сравнении с односторонним?
71. Что такое оптимизация Седжевика для быстрой сортировки?
72. В чем заключается устранение концевой рекурсии?
73. Почему в задаче поиска медианы можно полностью исключить рекурсию?
74. Что такое структура данных «куча»? Что такое представляющий массив?
75. Почему в кучу данные всегда вставляются в конец представляющего массива, а извлекаются из его начала?
76. Сравните верхнюю асимптотическую оценку быстрой и пирамидальной сортировки. Какая из них на практике оказывается быстрее?
77. В чем заключается линейный поиск с барьерным элементом? Где располагается барьер? Можно ли применить метод барьерного элемента для двоичного поиска?

78. В чем заключается двоичный поиск? Какое требование налагается на данные?
79. Двоичный поиск можно организовать с тремя и с двумя сравнениями внутри цикла. В чем разница? Какой метод эффективней?
80. Если метод поиска реализован в виде подпрограммы, что лучше возвращать – объект, ссылку на объект, указатель на объект или индекс объекта? Сравните варианты.
81. Что такое несамоподобный образец в методе КМП? Почему самоподобные образцы усложняют алгоритм?
82. Как вычисляется префикс-функция в алгоритме КМП? Какой смысл вычисляемого значения?
83. Какими свойствами должны обладать входные данные, чтобы КМП имел большие преимущества над прямолинейным поиском?
84. Какие методы косвенного вызова доступны в используемом вами языке программирования?
85. В чем отличия базовой и прямой индексации? Какой тип данных в каждом из случаев используется в индексе?
86. Что такое индекс? Как обратиться к данным в порядке, заданном индексом?
87. Можно ли для поиска и сортировки использовать один способ косвенного сравнения данных?
88. За счет чего сортировка индексов может оказаться быстрее прямой сортировки данных?
89. Почему для индексов, хранящихся во внешних файлах, базовая адресация удобней, чем прямая?
90. Что такое модель параллельных массивов? Как она сочетается с базовой адресацией?
91. Что такое префиксная и постфиксная форма записи выражений? Какое у них преимущество перед инфиксной с точки зрения интерпретации?
92. Опишите процесс вычисления выражения, записанного в постфиксной форме? Для хранения какого типа данных должен быть настроен стек?
93. Опишите алгоритм преобразования записи из инфиксной в постфиксную, для выражений не содержащих скобки. Можно ли скобки добавить только с использованием приоритетов операций?
94. Как расширить базовый алгоритм преобразования выражений, чтобы добавить поддержку скобок?
95. На какой тип данных должен быть настроен стек в алгоритме преобразования выражения в постфиксную форму?
96. Если выражение представить в виде дерева, то как можно из него получить все три формы записи?
97. Отличить в инфиксной записи бинарный и унарный минусы можно только по контексту. В чем заключается это отличие?

Типовые варианты контрольных работ

ВАРИАНТ 1

98. Опишите алгоритм преобразования записи из инфиксной в постфиксную, для выражений не содержащих скобки. Можно ли скобки добавлять только с использованием приоритетов операций?
99. Отличить в инфиксной записи бинарный и унарный минусы можно только по контексту. В чем заключается это отличие?
100. В чем отличия базовой и прямой индексации? Какой тип данных в каждом из случаев используется в индексе?
101. Почему в задаче поиска медианы можно полностью исключить рекурсию?
102. Объясните назначение второго параметра функции `fork()`.

ВАРИАНТ 2

1. Какой тип имеет строковый литерал? Чему будет равно выражение `sizeof("abc")`? Почему?
2. Как работают квадратные скобки в качестве спецификатора формата `scanf()`? Что задает следующий формат `scanf("%[^']", ...)`?
3. Чем отличается объявление и определение функции? Сколько объявлений одной и той же функции может быть в программе? При каком условии?
4. Предложите несколько возможных путей вернуть из функции пару целочисленных значений.
5. Что такое «концевая рекурсия»? Чем ее можно заменить?

ВАРИАНТ 3

1. Что такое `stdin`, `stdout` и `stderr`?
2. Сравните шейкер и пузырьковую сортировку.
3. Почему в двустороннем разделении нужно обеспечить продвижение индексов после того как произошел обмен?
4. В чем заключается линейный поиск с барьерным элементом? Где располагается барьер? Можно ли применить метод барьерного элемента для двоичного поиска?
5. В чем отличия базовой и прямой индексации? Какой тип данных в каждом из случаев используется в индексе?

Критерии и шкала оценивания по оценочному средству контрольная работа

Шкала оценивания	Критерий оценивания
5	Контрольная работа выполнена на высоком уровне (правильные ответы даны на 90-100% вопросов)
4	Контрольная работа выполнена на среднем уровне (правильные ответы даны на 75-89% вопросов)
3	Контрольная работа выполнена на низком уровне (правильные ответы даны на 50-74% вопросов)
2	Контрольная работа выполнена на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

Оценочные средства для промежуточной аттестации (зачет)

1. Какие парадигмы программирования реализованы в С и в С++
2. Назовите этапы подготовки программы к выполнению
3. Какие функции выполняет редактор связей
4. Что такое единица трансляции
5. Для чего применяется верхний и нижний регистр в тексте программы
6. С какой функции начинается выполнение программы
7. Что такое блочный оператор, как он соотносится с функцией
8. Как досрочно завершить программу
9. В каком файле объявлена функция exit()
10. Как записываются комментарии
11. Как представлены данные в С++ программе
12. Как записывается символьная константа
13. Что такое cout
14. Где обычно расположены операторы #include
15. Назовите основные характеристики переменной
16. По каким правилам записываются имена переменных
17. Что такое тип, что он определяет
18. Назовите скалярные типы языка С++
19. Где может объявляться переменная
20. Как соотносятся целый и символьный тип
21. Как осуществляется присваивание переменным
22. Как задать тип литеральной константы
23. Как записать восьмеричную и шестнадцатеричную константы
24. Как записываются константы с плавающей точкой
25. Как записываются строковые константы
26. Как строковые константы представлены в памяти
27. Что такое специальные символьные константы, как задать перевод строки
28. Что задает константа '\a'
29. Что такое символьный массив, как он соотносится с символьной константой
30. Как задать размер массива

31. Как напечатать содержание символьного массива
32. Как изменить отдельные символы в массиве
33. В каком случае компилятор автоматически рассчитает размер символьного массива
34. Чем символьный массив отличается от строки
35. Как занести значение в символьный массив
36. Что задает директива `#define`
37. Какая существует альтернатива `#define`
38. В чем отличие `#define` от `const` для задания констант
39. Какие объекты программы применяются для взаимодействия с консолью
40. Как осуществить вывод с новой строки
41. Что задает последовательность `\t`
42. Что такое манипуляторы потоков, в каком файле они определены
43. Как вывести значение в восьмеричной и шестнадцатеричной системе, как вернуться в десятичную систему
44. Как задать ширину поля при выводе
45. Как задать точность вывода значений с плавающей точкой
46. Как вводятся строки с помощью `cin >>`
47. Назовите основные математические операции
48. Как найти остаток от деления
49. В чем отличие бинарных и унарных операций
50. Как соотносятся приоритеты унарных и бинарных операций, какой приоритет операции присвоения
51. В каком порядке связываются операции присваивания, что такое многократное присваивание
52. Что такое составные операции, в чем их преимущество
53. Как производится преобразование типов в смешанных выражениях
54. Как задается явное приведение типа
55. Какие операции отношения существуют в C++
56. Как записать «не равно»
57. Какой синтаксис оператора `if`
58. Какие значения выражений рассматриваются как ИСТИНА, какие как ЛОЖНЫЕ
59. Как записывается оператор `else`
60. Какие логические операции существуют в C++, какое их основное свойство
61. Как заменить логическими операторами цепочку вложенных операторов `if`
62. Что такое тернарный оператор (условная операция), в чем его отличие от оператора `if`
63. Какой приоритет условной операции
64. Что такое операции инкремента и декремента
65. Что отличаются префиксная и постфиксная формы инкремента
66. Какой приоритет операций инкремента

67. Что такое оператор `sizeof()`, какой тип имеет размер единица
68. Что такое операция следования, какой ее приоритет и назначение
69. Чем поразрядные операции отличаются от логических
70. Почему нет операции логического XOR и что является эквивалентом
71. Какой операцией можно инвертировать заданный бит целого
72. Как проверить четность целого
73. Что такое операции сдвига, в чем разница для знаковых и беззнаковых типов
74. Из каких элементов состоит цикл `for`
75. Каким считается явно не указанное контрольное выражение в цикле `for`
76. Как записать «вечный цикл» с помощью `for`, `while`
77. Как записывается цикл `while`
78. В чем принципиальное отличие `for` и `while` от `do-while`
79. К чему приводит оператор `break` в теле цикла, вложенного цикла
80. Куда передает управление оператор `continue` внутри цикла `for`
81. В каком случае `switch` более предпочтителен чем `if-else`
82. В каком случае `if` нельзя заменить на `switch`
83. Является ли операторный блок обязательным элементом `switch`
84. В каком месте может располагаться выбор `default`
85. Что будет если пропустить `break` в `switch`
86. К чему будет относиться `continue` внутри `switch`
87. Где может располагаться метка для оператора `goto`
88. Для чего может потребоваться `goto`, всегда ли существует альтернатива
89. Какую роль в программе играют функции
90. Назовите основные элементы функции
91. Как вызвать функцию
92. Как завершить функцию и вернуть значение
93. Что такое локальные переменные, как они соотносятся с формальными параметрами
94. Как между собой соотносятся локальные переменные разных функций
95. Как соотносятся между собой локальные переменные одной и той же рекурсивной функции
96. Что такое статические переменные, как они соотносятся с локальными и глобальными переменными
97. Что такое передача параметру по значению, что будет если внутри функции изменить формальный параметр
98. Что такое передача по ссылке, какие объекты всегда передаются по ссылке
99. Что такое прототип (описание) функции, чем он отличается от определения
100. Что определяет прототип функции
101. В каком заголовочном файле определены стандартные символьные функции
102. Как сравнить две строки
103. Как преобразовать строку в число

104. В чем разница между функциями `abs()` и `fabs()`
105. Как взвести число в произвольную степень
106. Зачем нужна функция `atan2()`
107. Чем ввод-вывод в C++ отличается от других языков
108. Чем отличается буферизированный и не буферизированный ввод
109. Как в C++ устанавливается связь со стандартными потоками ввода-вывода
110. Что такое `cin`, `cout` и `cerr`. В каком заголовочном файле они объявлены
111. В чем отличие `cin.get(char *)` от `cin.operator>>(char *)`
112. Что такое последовательные файлы, какие операции с ними допустимы
113. Какие устройства могут рассматриваться как последовательные файлы
114. Как открыть и закрыть файл, какие режимы доступа существуют
115. В каком классе определены режимы доступа
116. В чем разница между `istream` и `ifstream`
117. Какую роль играют классы `ifstream` и `ofstream`
118. Как добавить данные в существующий файл
119. Что такое файлы с произвольным доступом, каким объектам операционной системы они соответствуют
120. Как установить позицию чтения/записи в файле с произвольным доступом
121. Как организовать файл с двоичными записями
122. Как проверить состояние файла
123. Как вернуться к началу файла
124. Как удалить дисковый файл
125. Как объявить одномерный массив
126. Какой индекс у первого элемента массива
127. Как инициализировать массив при объявлении, что будет если инициализировать массив не полностью
128. Что содержат не инициализированные явно локальные массивы, глобальные массивы
129. Запишите операцию эквивалентную `*(a+1)` с помощью квадратных скобок
130. Почему `*a` и `a[0]` эквивалентны
131. Как вычислить размер массива, количество его элементов
132. Как массивы передаются в функцию
133. Как ведет себя `sizeof()` для формальных параметров типа массив, почему
134. Как объявляется многомерный массив
135. В каком порядке указывается число строк и столбцов в двумерном массиве
136. Как многомерные массивы хранятся в памяти
137. Как инициализировать многомерный массив при инициализации

138. Как в функцию передаются многомерные массивы, какой из индексов следует опустить при описании
139. Ведет ли себя sizeof() для многомерных массивов в качестве формальных параметров также как и для одномерных
140. Что хранит переменная указатель
141. Как получить адрес объекта
142. Как получить доступ к объекту при наличии указателя
143. Какой синтаксис объявления указателей
144. Как объявить массив указателей
145. Что общего и в чем разница между массивом и указателем при использовании в выражении
146. К чему приведет инкремент переменной указателя, прибавления к нему целочисленной константы
147. Чему равна разница указателей, если они указывают на элементы одного массива
148. Что такое «массив с рваными краями»
149. Как задать и инициализировать массив из пяти строк, из пяти указателей на строки, в чем разница
150. Как создать массив оператором new
151. Какую форму delete необходимо использовать с динамическими массивами
152. Что такое «утечка памяти»
153. Как записать указатель четвертого уровня косвенности
154. Как записать функцию, принимающую указатель на другую функцию в качестве параметра
155. Что такое структура, каково ее назначение
156. Что общего и в чем отличие структуры и массива
157. Как инициализировать структурную переменную при создании
158. Как обратиться к полю структурной переменной
159. Назовите два способа обращения к полям структуры заданной косвенно через указатель
160. Каким образом структурные переменные передаются в функции и возвращаются из них
161. Как объявить массив структур
162. Как происходит копирование структур
163. Как сохранить структуру в файл и считать обратно
164. Как с помощью структуры передать массив в функцию «по значению»
165. Как передать по ссылке член структуры

Критерии и шкала оценивания по оценочному средству промежуточный контроль (зачет)

Характеристика знания предмета и ответов	Шкала оценивания
Студент глубоко и в полном объеме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач	зачтено
Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.	
Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.	
Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.	не зачтено

Оценочные средства для итоговой аттестации (экзамен)

1. Асимптотические оценки O , Ω , Θ . Интуитивный смысл.
2. Постановка задачи сортировки. Свойства сортировок.
3. Пузырьковая сортировка.
4. Шейкер сортировка
5. Сортировка с помощью выбора
6. Сортировка с помощью вставок
7. Сортировка Шелла. Классический алгоритм.
8. Сортировка Шелла. Выбор шага.
9. Быстрая сортировка. Одностороннее разделение.
10. Быстрая сортировка. Двустороннее разделение.
11. Быстрая сортировка. Оптимизация Седжвика. Рандомизированное разделение.
12. Быстрая сортировка. Разделение по медиане трех элементов.
13. Порядковые статистики. Одностороннее разделение.
14. Порядковые статистики. Двустороннее разделение.
15. Порядковые статистики. Разделение по медиане трех элементов.
16. Порядковые статистики. Устранение рекурсии.
17. Быстрая сортировка. Частичное устранение рекурсии.

- 18.Операции с кучей. Пирамидальная сортировка.
- 19.Операции с кучей. Очереди с приоритетом.
- 20.Сравнение методов сортировки.
- 21.Поиск минимума и максимума.
- 22.Линейный поиск. Применение барьеров.
- 23.Двоичный поиск и его ускорение.
- 24.Интерполяционный поиск.
- 25.Прямой поиск подстроки.
- 26.Алгоритм КМП.
- 27.Итерационное вычисление префикс функции.
- 28.Пессимизация.
- 29.Связь между алгоритмами и данными. Обобщенное программирование.
- 30.Связь между алгоритмами и данными. Косвенное определение операции.
- 31.Связь между алгоритмами и данными. Базовая индексация.
- 32.Базовая индексация. Массивы остатков.
- 33.Связь между алгоритмами и данными. Прямая индексация.
- 34.Абстрактный тип данных. Стек.
- 35.Абстрактный тип данных. Очередь.
- 36.Абстрактный тип данных. Дека.
- 37.Абстрактный тип данных. Очереди с приоритетом.
- 38.Абстрактный тип данных. Связное и динамическое распределение.
- 39.Абстрактная структура данных. Список.
- 40.Однонаправленные списки.
- 41.Двунаправленные списки.
- 42.Кольцевые списки и барьерные элементы.
- 43.Список. Связное распределение.
- 44.Список. Динамическое распределение
- 45.Поиск в списке.
- 46.Сортирующие списки.
- 47.Список. Самоорганизующийся поиск.
- 48.Список. Недостатки барьерного элемента.
- 49.Абстрактная структура данных. Словарь.
- 50.Словарь. Прямая адресация.
- 51.Словарь. Хеш-таблицы. Метод цепочек
- 52.Вычисление хеш-функций.
- 53.Словарь. Хеш-таблицы. Открытая адресация.
- 54.Открытая адресация. Линейное исследование.
- 55.Открытая адресация. Двойное хеширование.
- 56.Словарь. Метод цепочек и открытая адресация. Размер таблиц
- 57.Двоичное дерево поиска. Представление дерева поиска
- 58.Двоичное дерево. Рекурсивный обход.
- 59.Двоичное дерево. Поиск минимума и максимума.
- 60.Двоичное дерево. Поиск по ключу
- 61.Двоичное дерево. Поиск предшествующего и последующего элемента

62. Двоичное дерево. Нерекursивный обход дерева
63. Двоичное дерево. Вставка элемента
64. Двоичное дерево. Удаление элемента.
65. Производительность двоичных деревьев.
66. Двоичное дерево. Дублирование ключей.

Типовой экзаменационный билет.

ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. В.ДАЛЯ

Кафедра информационных и управляющих систем

Вариант 1

1. Операции с кучей. Очереди с приоритетом.
2. Абстрактный тип данных. Связное и динамическое распределение.
3. Двоичное дерево. Вставка элемента.

Утверждено на заседании кафедры информационных и управляющих систем _____, протокол _____

Заведующий кафедрой

А.И. Горбунов

Экзаменатор

Д.О. Синепольский

Критерии и шкала оценивания результатов промежуточной аттестации «экзамен»

Шкала оценивания (интервал баллов)	Критерий оценивания
отлично (5)	Студент глубоко и в полном объёме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.
Хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)

Экспертное заключение

Представленный фонд оценочных средств (далее – ФОС) по дисциплине «Программирование мехатронных и робототехнических систем» соответствует требованиям ФГОС ВО.

Предлагаемые формы и средства текущего и промежуточного контроля адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 15.03.06 Мехатроника и робототехника.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины представлены в полном объеме.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанный и представленный для экспертизы фонд оценочных средств рекомендуется к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической
комиссии факультета компьютерных
систем и информационных
технологий



Ветрова Н. Н.