ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

КАФЕДРА СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СРЕДСТВ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

Б1.О.17 «Алгоритмические языки и программирование»

Направление подготовки

11.03.03 Конструирование и технология электронных средств

Оценочные материалы – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур), предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной профессиональной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной профессиональной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности общекультурных, общепрофессиональных и профессиональных компетенций, приобретаемых обучающимся в соответствии с этими требованиями.

Контроль знаний проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль успеваемости проводится с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся и принятия необходимых мер по совершенствованию методики преподавания учебной дисциплины (модуля), организации работы обучающихся в ходе учебных занятий и оказания им индивидуальной помощи.

К контролю текущей успеваемости относятся проверка знаний, умений и навыков, приобретенных обучающимися в ходе выполнения индивидуальных заданий на практических занятиях и лабораторных работах. При оценивании результатов освоения практических занятий и лабораторных работ применяется шкала оценки «зачтено – не зачтено». Количество лабораторных и практических работ и их тематика определена рабочей программой дисциплины, утвержденной заведующим кафедрой.

Результат выполнения каждого индивидуального задания должен соответствовать всем критериям оценки в соответствии с компетенциями, установленными для заданного раздела дисциплины.

Промежуточный контроль по дисциплине осуществляется проведением теоретического зачета.

Форма проведения теоретического зачета – устный ответ по вопросам, сформулированным с учетом содержания учебной дисциплины и утвержденным на заседании кафедры. При подготовке к устному ответу обучаемый может составить в письменном виде план ответа, включающий в себя основные понятия и определения и т.п.

Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
1	Основы алгоритмизации и программирования	ОПК-3, ОПК-5	проверка конспектов лекций, защита ПЗ и ЛР, зачет
2	Программирование линейных алгоритмов на языке СИ++	ОПК-3, ОПК-5	проверка конспектов лекций, защита ПЗ и ЛР, зачет
3	Программирование алгоритмов разветвляющейся и циклической структуры	ОПК-3, ОПК-5	проверка конспектов лекций, защита ПЗ и ЛР, зачет
4	Использование структурированных типов данных на языке С++	ОПК-3, ОПК-5	проверка конспектов лекций, защита ПЗ и ЛР, зачет
5	Интерфейс и реализация модуля. Область действия и область видимости в С++. Пространства имён.	ОПК-3, ОПК-5	проверка конспектов лекций, защита ПЗ и ЛР, зачет

6	Основы объектно-ориентированного программирования на С++	ОПК-3, ОПК-5	проверка конспектов лекций, защита ПЗ и ЛР, зачет
7	Наследование в объектно-ориентированном программировании на С++	ОПК-3, ОПК-5	проверка конспектов лекций, защита ПЗ и ЛР, зачет
8	Шаблоны и контейнерные классы в объектно- ориентированном программировании на С++	ОПК-3, ОПК-5	проверка конспектов лекций, защита ПЗ и ЛР, зачет
9	Совместный проект «Разработка программного решения задачи коммивояжера»	ОПК-3, ОПК-5	проверка конспектов лекций, защита ПЗ и ЛР, зачет

Шкала оценки сформированности компетенций

Код компетенции	**Результаты освоения ОПОП** **Содержание компетенций**
ОПК-3	Способен применять методы поиска, хранения, обработки, анализа и представления в требуемом формате информации из различных источников и баз данных, соблюдая при этом основные требования информационной безопасности
ОПК-5	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения

В процессе оценки сформированности знаний, умений и навыков обучающегося по дисциплине, производимой на этапе промежуточной аттестации в форме теоретического зачета, используется оценочная шкала «зачтено – не зачтено»:

Оценка «зачтено» выставляется обучающемуся, который прочно усвоил предусмотренный программный материал; правильно, аргументировано ответил на все вопросы, с приведением примеров; показал глубокие систематизированные знания, владеет приемами рассуждения и сопоставляет материал из разных источников: теорию связывает с практикой, другими темами данного курса, других изучаемых предметов; без ошибок выполнил практическое задание.

Обязательным условием выставленной оценки является правильная речь в быстром или умеренном темпе. Дополнительным условием получения оценки «зачтено» могут стать хорошие успехи при выполнении самостоятельной и лабораторной работы, систематическая активная работа на практических занятиях.

Оценка «не зачтено» выставляется студенту, который не справился с 50% вопросов и заданий билета, в ответах на другие вопросы допустил существенные ошибки. Не может ответить на дополнительные вопросы, предложенные преподавателем. Целостного представления о взаимосвязях элементов курса и использования предметной терминологии у обучающегося нет. Оценивается качество устной и письменной речи, как и при выставлении положительной оценки.

Типовые контрольные задания или иные материалы

1. Дайте понятие класса в ООП. Синтаксис определение класса.
2. Дайте понятие объекта в ООП. Опишите структуру объекта.
3. Как описывается объект класса? Приведите пример.
4. Что такое методы класса? Как производится их определение и вызов?
5. Как определяется видимость компонентов класса?
6. Что такое конструктор класса? Для чего используется и как определяется?
7. Каковы особенности конструкторов классов?
8. Что такое деструктор класса? Как он определяется?

9. Как производится инициализации данных объекта с помощью конструктора?
10. Как определяются указатели на компоненты-функции?
11. Что такое «конструктор», какие его основные особенности?
12. Что такое «деструктор», какие его основные особенности?
13. Какие имена должны быть у конструктора и деструктора?
14. Когда наличие деструктора обязательно?
15. Назначение операций *new* и *delete*.
16. Сколько существуют динамические объекты?
17. Формат и назначение библиотечной функции *malloc*().
18. Формат и назначение библиотечной функции *calloc*().
19. Формат и назначение библиотечной функции *free*().
20. Как создать и уничтожить динамический одномерный массив при помо-щи операций?
21. Как программно контролируется захват памяти под массив?
22. Чем является имя двухмерного массива?
23. Как косвенно обратиться к элементу двухмерного массива?
24. Объясните понятие «функция-друг» класса?
25. Какие существуют способы обращения к элементам класса из функции-друга?

Вопросы к зачету по дисциплине

1) Алгоритм. Свойства алгоритма. Формы записи алгоритмов.
2) Вызов подпрограмм. Формальные и фактические параметры.
3) Двумерные массивы способы их описания, ввода-вывода и обработки.
4) Доступ к элементам класса. Спецификаторы доступа. Пример (класс "Комплексное число 2").
5) Иерархия классов и множественное наследование. Пример (color_triangle).
6) Класс string стандартной библиотеки C++. Операции и конструкторы. Примеры.
7) Класс string стандартной библиотеки C++. Методы присваивания и добавления частей. Примеры.
8) Класс string стандартной библиотеки C++. Методы модификации. Примеры.
9) Класс string стандартной библиотеки C++. Методы поиска. Примеры.
10) Конструкторы и деструкторы класса. Пример (класс "Комплексное число 3").
11) Массивы в C++, объявление и инициализация. Пример (подсчет суммы элементов главной диагонали матрицы).
12) Массивы. Сортировка элементов массивов методом "пузырька".
13) Массивы. Способы описания и использования одномерных массивов.
14) Модульное программирование. Интерфейс модуля и реализация модуля, организация взаимодействия между файлами.
15) Модульное программирование. Общие положения. Реализация в C++.
16) Наследование классов. Ограничение доступа к элементам базового класса. Пример иерархии классов ("геометрические фигуры").
17) Области действия и пространство имён в C++.
18) Объектно-ориентированное программирование. Предпосылки возникновения и основные идеи. Основные концепции ООП.
19) Форматированный ввод-вывод с помощью потоковых классов C++. Простые и параметризованные манипуляторы. Примеры.
20) Форматированный ввод-вывод с помощью потоковых классов C++. Флаги и методы форматирования. Примеры.
21) Подпрограммы. Применение. Структура описания. Отличие процедуры от функции.
22) Понятие класса и объекта в C++. Пример класса "Комплексное число".

23) Потоковые классы стандартной библиотеки C++. Виды потоков. Иерархия потоков. Стандартные потоки.
24) Правила наследования различных методов, конструкторов и деструкторов класса. Примеры.
25) Стандартная библиотека C++. Состав модулей, включение модулей в программу.
26) Структура программы на языке C++, схема компиляции программ.
27) Структуризация. Основы структурного программирования. Преимущества использования методологии структурного программирования.
28) Структурные схемы алгоритмов. Правила выполнения.
29) Структурный тип данных. Структуры в C++. Примеры.
30) Структуры в C++, объявление и инициализация. Примеры (геометрические фигуры).
31) Файловые потоки C++. Создание, открытие и закрытие файлового потока. Примеры.
32) Файловые потоки C++. Контроль состояния потока. Примеры.
33) Строковые потоки C++. Примеры.
34) Функции в C++, объявление, определение и вызов, возвращаемое значение. Примеры.
35) Функции в C++. Параметры функции, передача структурных переменных и массивов в качестве параметров. Параметры со значениями по умолчанию. Примеры.
36) Этапы подготовки и решения задач на ЭВМ, жизненный цикл ПО. Определение транслятора, интерпретатора.
37) Язык программирования C++, история возникновения и области применения.
38) Стандартная библиотека шаблонов STL языка C++. Принципы устройства.
39) Шаблоны функций.
40) Шаблоны классов.
41) Особенности реализации обобщённого программирования.
42) Основные компоненты STL.
43) Контейнеры STL.
44) Последовательные контейнеры. Вектор. Примеры.
45) Последовательные контейнеры. Список. Примеры.
46) Итераторы STL. Категории итераторов. Просмотр элементов контейнера.
47) Алгоритмы STL.
48) Алгоритмы с функциональными параметрами STL.

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

ПОДПИСАНО ЗАВЕДУЮЩИМ КАФЕДРЫ	**ФГБОУ ВО "РГРТУ", РГРТУ,** Корячко Вячеслав Петрович, Заведующий кафедрой САПР	**29.10.2**3 15:43 (MSK)	Простая подпись
ПОДПИСАНО ЗАВЕДУЮЩИМ ВЫПУСКАЮЩЕЙ КАФЕДРЫ	**ФГБОУ ВО "РГРТУ", РГРТУ,** Корячко Вячеслав Петрович, Заведующий кафедрой САПР	**29.10.2**3 15:43 (MSK)	Простая подпись