Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра систем сбора и обработки данных

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н., доцент И.Л. Рева

ДОКУМЕНТ ПОДПИСАН ПРОСТОЙ НЕКВАЛИФИЦИРОВАННОЙ ЭЛЕКТРОННОЙ ПОДПИСЬЮ
31.08.2026
Владелец: И.Л. Рева
Срок действия: не ограничен
Адрес хранения электронного документа:
https://www.nstu.ru/university/info/sveden/education

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ДИСЦИПЛИНЫ

Программирование

Образовательная программа: 09.03.02 Информационные системы и технологии, профиль: Информационные системы в промышленности и бизнесе

Новосибирск 2026

1. Обобщенная структура фонда оценочных средств дисциплины

Обобщенная структура фонда оценочных средств по **дисциплине** Программирование представлена в Таблице. Результаты обучения по дисциплине соотнесены с уровнями сформированности компетенций и соотнесенных с ними индикаторов. Индикаторы достижения компетенций измеряемы с помощью средств текущей и промежуточной аттестации по дисциплине.

Таблица

Формируемые компетенции	**Индикаторы компетенций**	**Темы**	**Этапы оценки компетенций и соотнесенных с ними индикаторов**	
			Мероприятия текущего контроля (контрольная работа, курсовой проект, РГЗ(Р), реферат и др.)	**Промежуточная аттестация (экзамен, зачет)**
ОПК-4 Способен участвовать в разработке технической документации, связанной с профессиональной деятельностью с использованием стандартов, норм и правил	3. Иметь навыки: составления технической документации на различных этапах жизненного цикла информационной системы.	Два примера разработки классов (дата и матрица переменной размерности). Действие и результат при выполнении элемента-функции или переопределения операции. Типы результата: пустой, базовый, указатель и ссылка на объект, объект. Временные объекты. Конструктор копирования и оператор присваивания, различия между ними. Друзья класса (функции и классы). Иерархия классов. Базовый и производные классы. Доступность их элементов. Наследование. Синтаксис конструирования объектов производных классов. Классы. Переопределение стандартных операций. Краткий обзор особенностей C++ в сравнении с C: присваивание структур, структуры как формальные параметры и результаты функций, переопределение операций и функций; элементы-функции; операторы управления динамической памятью; ссылки. Множественное наследование. Виртуальные базовые классы. Абстрактные классы. Шаблоны. Назначение и состав библиотеки STL. Контейнеры, категории контейнеров, последовательные контейнеры. Итераторы. Объекты-функции. Векторы. Списки. Стеки. Очереди. Множество и мультимножество. Универсальные алгоритмы. Определение класса. Элементы-данные и элементы-функции	Курсовая работа, разделы: Титульный лист. Формулировка задания. Пояснения к разработанной программе. Листинг программы с комментариями. Вид экранных форм. Заключение. Список литературы и источников. Отчет по лабораторной работе № 1-12. РГЗ, разделы: Титульный лист. Формулировка задания. Пояснения к разработанной программе. Листинг программы с комментариями. Вид экранных форм. Заключение. Список литературы и источников.	Экзамен, вопросы 1-15 (2 семестр), 1-22 (3 семестр)

		класса. Общие, частные и защищенные элементы класса. Объекты и их свойства, время жизни и инициализация. Определение конструктора и деструктора, их свойства. Оператор присваивания. Типы объектов: внешний, автоматический, динамический, временный. Переопределение (перегрузка) операций. Пример. Переопределение функций. Пример. Особенности переопределения некоторых операций: преобразования типа; индексации; динамической памяти, присваивания. Полиморфизм и виртуальные функции. Абстрактные классы и чисто виртуальные функции. Виртуальные деструкторы. Понятие объектно-ориентированного проектирования (ООП). Понятие объекта, объекта и класса, связь с переменной и типом данных. Последовательные текстовые файлы. Варианты в порядке увеличения сложности. На четверку: Вариант 1. Открыть заданный текстовый файл и вывести на экран строку с заданным номером Имя файла и номер строки должны передаваться в программу в виде аргументов командной строки. Вариант 2. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной не меняя порядка строк. Имена файлов задаются в виде аргументов командной строки. На пятерку: Вариант 3. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в обратном порядке строк. Имена файлов задаются в виде аргументов командной строки. Вариант 4. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в порядке убывания их длины. Имена файлов задаются в виде аргумен		

		Разработать абстрактный базовый класс object, а для него предусмотреть чистые виртуальные методы. Разработанный ранее класс - ТД, сделать наследником от object. При этом переопределить чистые виртуальные методы класса object,наполнив их конкретным содержанием. Разработать класс - тип данных (ТД) с переопределением операторов. Разработать функции ввода, вывода, базовый набор содержательных операций над ТД (например, арифметических и сравнения) - 4 часа. Варианты типов данных: а) целые произвольной длины, представленные строкой цифр; б) то же, но во внутреннем двоичном представлении (динамический массив байтов); в) матрицы переменной размерности; г) разреженная матрица переменной размерности, ненулевые элементы представлены динамическим массивом с элементами (x, y, v); д) дроби; е) дата и время; ж) комплексные числа; з) разрежены матрицы переменной размерности, ненулевые элементы представлены односвязным списком. Разработать шаблонный класс (с использованием стандартной библиотеки шаблонов STL), реализующей определенную структуру хранения данных (т.н. "контейнерный" класс). Данные в контейнере хранятся либо в виде указателей на объекты, либо в виде самих объектов. Структуры хранения данных: а) двоичное дерево; б) массив; в) односвязный линейный список; г) двусвязный циклический список. Разработать демонстрационную программу Создание и использование динамически присоединяемых библиотек. Функции. Перегрузка функций. Функции. Аргументы по умолчанию. Цель: углубить знания стандартной библиотеки шаблонов (STL). Общие требования: " для обращения к элементам заданного контейнера обязательно		

		использовать соответствующие итераторы. " Если задание предусматривает выполнение сортировки, поиска или перебора диапазона, эти действия необходимо выполнить с помощью алгоритмов библиотеки или встроенных в контейнеры функций STL " Обработка ошибок должна быть реализована с помощью выбрасывания исключений - объектов класса Exception. " Работу программы проверить на файлах очень большого размера. Вариант 1. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; Считать все строки заданного входного файла в структуру данных - список страниц. Страница - вектор из 24 строк, объектов класса AnsiString (последняя страница может быть неполной). Вывести на экран страницу с заданным номером. Вариант 2. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; Считать все строки заданного входного файла в структуру данных - список страниц. С 1. Работа с инспектором объектов. " Создать приложение с главным окном, появляющимся в центре экрана, с размерами 200 на 300 (высота, ширина). " Сменить название класса формы на MForm, сменить заголовок формы. " Запретить изменение размеров окна и распахивание во весь экран. " Экспериментальным путем выяснить за что отвечает каждое из свойств класс TForm. " Нарисовать собственную иконку используя редактор ресурсов и назначить ее приложению. 2. Обработка событий. " Доработать предыдущее приложение так, чтобы размеры и положение главной формы вычислялось автоматически при ее создании. Положение - правый нижний угол, высота и ширина равны четверти размеров экрана. (Событие OnCreate, объект Screen) " Поместить на форму стандартную кнопку. Сделать так, чтобы она "убегала" от курсора мыши. (Событие OnMouseMove, свойства Left, Top) 3. Диалоговая панель "		

		Доработать приложение, поместив вторую кнопку и создав вторую форму. Сделать так, чтобы при нажатии на кнопку появлялась форма		
ОПК-5 Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем	3. Иметь навыки: инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем.	Двоичные файлы произвольного доступа. Варианты в порядке увеличения сложности. На четверку: Вариант 1. Двоичный файл записей фиксированной длины. Написать функции записи в файл с заданным именем массива чисел типа double и чтения из файла числа с заданным номером. Программа должна работать следующим образом: если в командной строке один аргумент (это должно быть имя файла), то вызывается функция записи введенного с клавиатуры массива чисел, если аргументов два (имя файла и номер числа), то вызывается функция чтения из файла числа с заданным номером и вывод его на экран. На пятерку: Вариант 2. Открыть заданный текстовый файл и записать все его строки в двоичный файл записей переменной длины (каждая строка записывается в виде записи из двух полей [длина][содержимое]). После чего необходимо открыть созданный файл и вывести его содержимое на экран. Имя входного (текстового) файла задается в виде аргумента командной строки. Имя выходного (двоичного) файла генерируется программой. Дружественные функции и дружественные классы. Конструктор и операция NEW. Классы окон. Дочерние окна. Элементы окон: полосы прокрутки, панель состояния, инструментальная панель. Классы. Переопределение стандартных операций. Назначение и состав библиотеки STL. Контейнеры, категории контейнеров, последовательные контейнеры. Итераторы. Объекты-функции. Векторы. Списки. Стеки. Очереди. Множество и мультимножество. Универсальные алгоритмы. Понятие DLL. Создание DLL. Присоединение DLL к программе. DLL ресурсов. Пример создания DLL. Понятие	Курсовая работа, разделы: Титульный лист. Формулировка задания. Пояснения к разработанной программе. Листинг программы с комментариями. Вид экранных форм. Заключение. Список литературы и источников. Отчет по лабораторной работе № 1-12. РГЗ, разделы: Титульный лист. Формулировка задания. Пояснения к разработанной программе. Листинг программы с комментариями. Вид экранных форм. Заключение. Список литературы и источников.	Экзамен, вопросы 1-15 (2 семестр), 1-22 (3 семестр)

		графического интерфейса устройства (GDI). Контексты устройств. Графические классы. Графические объекты. Построение геометрических фигур. Создание типов кистей и перьев. Отображение теста. Растровые изображения. Создание шрифтов. Понятие многодокументного интерфейса (MDI). Рабочая область приложения, дочерние окна, просмотр и упорядочивание окон. Создание приложения MDI с помощью MFC. MDI и технологии Drag and Drop. Пример программного приложения MDI. Создание шаблона приложения. Создание однооконного приложения с помощью App Wizard. Создание многооконного приложения. Понятие ресурсов. Типы ресурсов. Создание ресурсов. Редактирование ресурсов. Пример разработки ресурсов. Последовательные текстовые файлы. Варианты в порядке увеличения сложности. На четверку: Вариант 1. Открыть заданный текстовый файл и вывести на экран строку с заданным номером Имя файла и номер строки должны передаваться в программу в виде аргументов командной строки. Вариант 2. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной не меняя порядка строк. Имена файлов задаются в виде аргументов командной строки. На пятерку: Вариант 3. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в обратном порядке строк. Имена файлов задаются в виде аргументов командной строки. Вариант 4. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в порядке убывания их длины. Имена файлов задаются в виде аргумен		

		Потоковый ввод-вывод в C++. Разработать абстрактный базовый класс object, а для него предусмотреть чистые виртуальные методы. Разработанный ранее класс - ТД, сделать наследником от object. При этом переопределить чистые виртуальные методы класса object,наполнив их конкретным содержанием. Разработать шаблонный класс (с использованием стандартной библиотеки шаблонов STL), реализующей определенную структуру хранения данных (т.н. "контейнерный" класс). Данные в контейнере хранятся либо в виде указателей на объекты, либо в виде самих объектов. Структуры хранения данных: а) двоичное дерево; б) массив; в) односвязный линейный список; г) двусвязный циклический список. Разработать демонстрационную программу Раннее и позднее (динамическое) связывание. Виртуальные функции. Чисто абстрактные классы. Создание и использование динамически присоединяемых библиотек. Создание консоли. Ввод и вывод в консоли. Обработка событий клавиатуры и мыши. Стандартный класс std::string. Конструкторы строк. Доступ к символам строк. Операции поиска. Операции вставки, замены и удаления символов строки. Операторы ввода-вывода строк. Таблица сообщений. Сообщения от клавиатуры. Сообщения от мыши. Управление исключениями. Понятие исключения. Обработка исключительных ситуаций. Генерирование исключений. Перехватывание исключений. Использование блоков try/catch. Неожиданные исключения и обработка завершения. Файлы программы для Windows. Типы данных в Windows. Обмен сообщениями между приложениями и ОС Windows. Понятие о Windows-APL. Структура приложения Windows. Определение класса окна. Регистрация класса окна. Создание и отображение окна. Цикл обработки сообщений.		

		Пример простой Windows программы. Библиотека Microsoft Foundation Classes (MFC). Краткая характеристика библиотеки. Разработка приложений Windows с использованием MFC. Пример программы. Функции. Перегрузка функций. Функции. Аргументы по умолчанию. Цель: освоить стандартную библиотеку шаблонов (STL). Общие требования: " для обращения к элементам заданного контейнера обязательно использовать соответствующие итераторы. " Если задание предусматривает выполнение сортировки, ее необходимо выполнить с помощью алгоритма sort из STL. " Обработка ошибок должна быть реализована с помощью выбрасывания исключений - объектов класса Exception. " Работу программы проверить на файлах очень большого размера. Вариант 1. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Считать все строки входного файла в контейнер vector<AnsiString>. Отсортировать строки в контейнере в алфавитном порядке, преобразовать к верхнему регистру и записать их в выходной файл. Вариант 2. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Считать позиции начал строк входного файла в контейнер deque<int>. Используя эти данные и реверсивный итератор записать в выходной файл Цель: углубить знания стандартной библиотеки шаблонов (STL). Общие требования: " для обращения к элементам заданного контейнера обязательно использовать соответствующие итераторы. " Если задание предусматривает выполнение сортировки, поиска или перебора диапазона, эти действия необходимо выполнить с помощью алгоритмов библиотеки или встроенных в контейнеры функций STL " Обработка		

		ошибок должна быть реализована с помощью выбрасывания исключений - объектов класса Exception. " Работу программы проверить на файлах очень большого размера. Вариант 1. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; Считать все строки заданного входного файла в структуру данных - список страниц. Страница - вектор из 24 строк, объектов класса AnsiString (последняя страница может быть неполной). Вывести на экран страницу с заданным номером. Вариант 2. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; Считать все строки заданного входного файла в структуру данных - список страниц. С Шаблоны функций. Перегрузка и специализации шаблонов функций. Шаблоны функций сортировки. Шаблоны классов. Специализация шаблонов классов. Статические члены шаблонов классов. Класс - шаблон для вектора. 1. Работа с инспектором объектов. " Создать приложение с главным окном, появляющимся в центре экрана, с размерами 200 на 300 (высота, ширина). " Сменить название класса формы на MForm, сменить заголовок формы. " Запретить изменение размеров окна и распахивание во весь экран. " Экспериментальным путем выяснить за что отвечает каждое из свойств класс TForm. " Нарисовать собственную иконку используя редактор ресурсов и назначить ее приложению. 2. Обработка событий. " Доработать предыдущее приложение так, чтобы размеры и положение главной формы вычислялось автоматически при ее создании. Положение - правый нижний угол, высота и ширина равны четверти размеров экрана. (Событие OnCreate, объект Screen) " Поместить на форму стандартную кнопку. Сделать так, чтобы она "убегала" от курсора мыши. (Событие OnMouseMove, свойства Left, Top) 3. Диалоговая панель " Доработать приложение, поместив вторую кнопку и		

		создав вторую форму. Сделать так, чтобы при нажатии на кнопку появлялась форма		
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	1. Знать: методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий	Виртуальные функции. Чисто абстрактные классы. Два примера разработки классов (дата и матрица переменной размерности). Двоичные файлы произвольного доступа. Варианты в порядке увеличения сложности. На четверку: Вариант 1. Двоичный файл записей фиксированной длины. Написать функции записи в файл с заданным именем массива чисел типа double и чтения из файла числа с заданным номером. Программа должна работать следующим образом: если в командной строке один аргумент (это должно быть имя файла), то вызывается функция записи введенного с клавиатуры массива чисел, если аргументов два (имя файла и номер числа), то вызывается функция чтения из файла числа с заданным номером и вывод его на экран. На пятерку: Вариант 2. Открыть заданный текстовый файл и записать все его строки в двоичный файл записей переменной длины (каждая строка записывается в виде записи из двух полей [длина][содержимое]). После чего необходимо открыть созданный файл и вывести его содержимое на экран. Имя входного (текстового) файла задается в виде аргумента командной строки. Имя выходного (двоичного) файла генерируется программой. Действие и результат при выполнении элемента-функции или переопределения операции. Типы результата: пустой, базовый, указатель и ссылка на объект, объект. Временные объекты. Конструктор копирования и оператор присваивания, различия между ними. Дружественные функции и дружественные классы. Конструктор и операция NEW. Друзья класса (функции и классы). Иерархия классов. Базовый и производные классы. Доступность их элементов. Наследование. Синтаксис конструирования объектов производных классов. Классы окон. Дочерние окна. Элементы	Курсовая работа, разделы: Пояснения к разработанной программе. Листинг программы с комментариями. Вид экранных форм. Отчет по лабораторной работе № 1-12. РГЗ, разделы: Пояснения к разработанной программе. Листинг программы с комментариями. Вид экранных форм.	Экзамен, вопросы 1-15 (2 семестр), 1-22 (3 семестр)

		окон: полосы прокрутки, панель состояния, инструментальная панель. Классы. Переопределение стандартных операций. Краткий обзор особенностей C++ в сравнении с C: присваивание структур, структуры как формальные параметры и результаты функций, переопределение операций и функций; элементы-функции; операторы управления динамической памятью; ссылки. Множественное наследование. Виртуальные базовые классы. Абстрактные классы. Шаблоны. Назначение и состав библиотеки STL. Контейнеры, категории контейнеров, последовательные контейнеры. Итераторы. Объекты-функции. Векторы. Списки. Стеки. Очереди. Множество и мультимножество. Универсальные алгоритмы. Определение класса. Элементы-данные и элементы-функции класса. Общие, частные и защищенные элементы класса. Объекты и их свойства, время жизни и инициализация. Определение конструктора и деструктора, их свойства. Оператор присваивания. Типы объектов: внешний, автоматический, динамический, временный. Переопределение (перегрузка) операций. Пример. Переопределение функций. Пример. Особенности переопределения некоторых операций: преобразования типа; индексации; динамической памяти, присваивания. Полиморфизм и виртуальные функции. Абстрактные классы и чисто виртуальные функции. Виртуальные деструкторы. Понятие DLL. Создание DLL. Присоединение DLL к программе. DLL ресурсов. Пример создания DLL. Понятие графического интерфейса устройства (GDI). Контексты устройств. Графические классы. Графические объекты. Построение геометрических фигур. Создание типов кистей и перьев. Отображение теста. Растровые изображения. Создание шрифтов. Понятие многодокументного интерфейса (MDI). Рабочая область приложения, дочерние		

		окна, просмотр и упорядочивание окон. Создание приложения MDI с помощью MFC. MDI и технологии Drag and Drop. Пример программного приложения MDI. Создание шаблона приложения. Создание однооконного приложения с помощью App Wizard. Создание многооконного приложения. Понятие объектно-ориентированного проектирования (ООП). Понятие объекта, объекта и класса, связь с переменной и типом данных. Понятие ресурсов. Типы ресурсов. Создание ресурсов. Редактирование ресурсов. Пример разработки ресурсов. Последовательные текстовые файлы. Варианты в порядке увеличения сложности. На четверку: Вариант 1. Открыть заданный текстовый файл и вывести на экран строку с заданным номером Имя файла и номер строки должны передаваться в программу в виде аргументов командной строки. Вариант 2. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной не меняя порядка строк. Имена файлов задаются в виде аргументов командной строки. На пятерку: Вариант 3. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в обратном порядке строк. Имена файлов задаются в виде аргументов командной строки. Вариант 4. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в порядке убывания их длины. Имена файлов задаются в виде аргумен Потоковый ввод-вывод в C++. Разработать абстрактный базовый класс object, а для него предусмотреть чистые виртуальные методы. Разработанный ранее класс -		

		ТД, сделать наследником от object. При этом переопределить чистые виртуальные методы класса object,наполнив их конкретным содержанием. Разработать класс - тип данных (ТД) с переопределением операторов. Разработать функции ввода, вывода, базовый набор содержательных операций над ТД (например, арифметических и сравнения) - 4 часа. Варианты типов данных: а) целые произвольной длины, представленные строкой цифр; б) то же, но во внутреннем двоичном представлении (динамический массив байтов); в) матрицы переменной размерности; г) разреженная матрица переменной размерности, ненулевые элементы представлены динамическим массивом с элементами (x, y, v); д) дроби; е) дата и время; ж) комплексные числа; з) разрежены матрицы переменной размерности, ненулевые элементы представлены односвязным списком. Разработать шаблонный класс (с использованием стандартной библиотеки шаблонов STL), реализующей определенную структуру хранения данных (т.н. "контейнерный" класс). Данные в контейнере хранятся либо в виде указателей на объекты, либо в виде самих объектов. Структуры хранения данных: а) двоичное дерево; б) массив; в) односвязный линейный список; г) двусвязный циклический список. Разработать демонстрационную программу Раннее и позднее (динамическое) связывание. Виртуальные функции. Чисто абстрактные классы. Создание и использование динамически присоединяемых библиотек. Создание консоли. Ввод и вывод в консоли. Обработка событий клавиатуры и мыши. Стандартный класс std::string. Конструкторы строк. Доступ к символам строк. Операции поиска. Операции вставки, замены и удаления символов строки. Операторы ввода-вывода строк. Таблица		

		сообщений. Сообщения от клавиатуры. Сообщения от мыши. Управление исключениями. Понятие исключения. Обработка исключительных ситуаций. Генерирование исключений. Перехватывание исключений. Использование блоков try/catch. Неожиданные исключения и обработка завершения. Файлы программы для Windows. Типы данных в Windows. Обмен сообщениями между приложениями и ОС Windows. Понятие о Windows-APL. Структура приложения Windows. Определение класса окна. Регистрация класса окна. Создание и отображение окна. Цикл обработки сообщений. Пример простой Windows программы. Библиотека Microsoft Foundation Classes (MFC). Краткая характеристика библиотеки. Разработка приложений Windows с использованием MFC. Пример программы. Функции. Перегрузка функций. Функции. Аргументы по умолчанию. Цель: освоить стандартную библиотеку шаблонов (STL). Общие требования: " для обращения к элементам заданного контейнера обязательно использовать соответствующие итераторы. " Если задание предусматривает выполнение сортировки, ее необходимо выполнить с помощью алгоритма sort из STL. " Обработка ошибок должна быть реализована с помощью выбрасывания исключений - объектов класса Exception. " Работу программы проверить на файлах очень большого размера. Вариант 1. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Считать все строки входного файла в контейнер vector<AnsiString>. Отсортировать строки в контейнере в алфавитном порядке, преобразовать к верхнему регистру и записать их в выходной файл. Вариант 2. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой).		

		Считать позиции начал строк входного файла в контейнер deque<int>. Используя эти данные и реверсивный итератор записать в выходной файл Цель: углубить знания стандартной библиотеки шаблонов (STL). Общие требования: " для обращения к элементам заданного контейнера обязательно использовать соответствующие итераторы. " Если задание предусматривает выполнение сортировки, поиска или перебора диапазона, эти действия необходимо выполнить с помощью алгоритмов библиотеки или встроенных в контейнеры функций STL " Обработка ошибок должна быть реализована с помощью выбрасывания исключений - объектов класса Exception. " Работу программы проверить на файлах очень большого размера. Вариант 1. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; Считать все строки заданного входного файла в структуру данных - список страниц. Страница - вектор из 24 строк, объектов класса AnsiString (последняя страница может быть неполной). Вывести на экран страницу с заданным номером. Вариант 2. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; Считать все строки заданного входного файла в структуру данных - список страниц. С Шаблоны функций. Перегрузка и специализации шаблонов функций. Шаблоны функций сортировки. Шаблоны классов. Специализация шаблонов классов. Статические члены шаблонов классов. Класс - шаблон для вектора. 1. Работа с инспектором объектов. " Создать приложение с главным окном, появляющимся в центре экрана, с размерами 200 на 300 (высота, ширина). " Сменить название класса формы на MForm, сменить заголовок формы. " Запретить изменение размеров окна и распахивание во весь экран. " Экспериментальным путем выяснить за что отвечает каждое из свойств класс TForm.		

		" Нарисовать собственную иконку используя редактор ресурсов и назначить ее приложению. 2. Обработка событий. " Доработать предыдущее приложение так, чтобы размеры и положение главной формы вычислялось автоматически при ее создании. Положение - правый нижний угол, высота и ширина равны четверти размеров экрана. (Событие OnCreate, объект Screen) " Поместить на форму стандартную кнопку. Сделать так, чтобы она "убегала" от курсора мыши. (Событие OnMouseMove, свойства Left, Top) 3. Диалоговая панель " Доработать приложение, поместив вторую кнопку и создав вторую форму. Сделать так, чтобы при нажатии на кнопку появлялась форма		
ОПК-6	2. Уметь: применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий	Два примера разработки классов (дата и матрица переменной размерности). Двоичные файлы произвольного доступа. Варианты в порядке увеличения сложности. На четверку: Вариант 1. Двоичный файл записей фиксированной длины. Написать функции записи в файл с заданным именем массива чисел типа double и чтения из файла числа с заданным номером. Программа должна работать следующим образом: если в командной строке один аргумент (это должно быть имя файла), то вызывается функция записи введенного с клавиатуры массива чисел, если аргументов два (имя файла и номер числа), то вызывается функция чтения из файла числа с заданным номером и вывод его на экран. На пятерку: Вариант 2. Открыть заданный текстовый файл и записать все его строки в двоичный файл записей переменной длины (каждая строка записывается в виде записи из двух полей [длина][содержимое]). После чего необходимо открыть созданный файл и вывести его содержимое на экран. Имя входного (текстового) файла задается в виде аргумента командной строки. Имя выходного (двоичного) файла генерируется программой.	Курсовая работа, разделы: Пояснения к разработанной программе. Листинг программы с комментариями. Вид экранных форм. Отчет по лабораторной работе № 1-12. РГЗ, разделы: Пояснения к разработанной программе. Листинг программы с комментариями. Вид экранных форм.	Экзамен, вопросы 1-15 (2 семестр), 1-22 (3 семестр)

		Действие и результат при выполнении элемента-функции или переопределения операции. Типы результата: пустой, базовый, указатель и ссылка на объект, объект. Временные объекты. Конструктор копирования и оператор присваивания, различия между ними. Друзья класса (функции и классы). Иерархия классов. Базовый и производные классы. Доступность их элементов. Наследование. Синтаксис конструирования объектов производных классов. Классы окон. Дочерние окна. Элементы окон: полосы прокрутки, панель состояния, инструментальная панель. Классы. Переопределение стандартных операций. Краткий обзор особенностей C++ в сравнении с C: присваивание структур, структуры как формальные параметры и результаты функций, переопределение операций и функций; элементы-функции; операторы управления динамической памятью; ссылки. Множественное наследование. Виртуальные базовые классы. Абстрактные классы. Шаблоны. Назначение и состав библиотеки STL. Контейнеры, категории контейнеров, последовательные контейнеры. Итераторы. Объекты-функции. Векторы. Списки. Стеки. Очереди. Множество и мультимножество. Универсальные алгоритмы. Определение класса. Элементы-данные и элементы-функции класса. Общие, частные и защищенные элементы класса. Объекты и их свойства, время жизни и инициализация. Определение конструктора и деструктора, их свойства. Оператор присваивания. Типы объектов: внешний, автоматический, динамический, временный. Переопределение (перегрузка) операций. Пример. Переопределение функций. Пример. Особенности переопределения некоторых операций: преобразования типа; индексации; динамической памяти, присваивания. Полиморфизм и виртуальные функции. Абстрактные классы и чисто виртуальные функции.		

		Виртуальные деструкторы. Понятие DLL. Создание DLL. Присоединение DLL к программе. DLL ресурсов. Пример создания DLL. Понятие графического интерфейса устройства (GDI). Контексты устройств. Графические классы. Графические объекты. Построение геометрических фигур. Создание типов кистей и перьев. Отображение теста. Растровые изображения. Создание шрифтов. Понятие многодокументного интерфейса (MDI). Рабочая область приложения, дочерние окна, просмотр и упорядочивание окон. Создание приложения MDI с помощью MFC. MDI и технологии Drag and Drop. Пример программного приложения MDI. Создание шаблона приложения. Создание однооконного приложения с помощью App Wizard. Создание многооконного приложения. Понятие объектно-ориентированного проектирования (ООП). Понятие объекта, объекта и класса, связь с переменной и типом данных. Понятие ресурсов. Типы ресурсов. Создание ресурсов. Редактирование ресурсов. Пример разработки ресурсов. Последовательные текстовые файлы. Варианты в порядке увеличения сложности. На четверку: Вариант 1. Открыть заданный текстовый файл и вывести на экран строку с заданным номером Имя файла и номер строки должны передаваться в программу в виде аргументов командной строки. Вариант 2. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной не меняя порядка строк. Имена файлов задаются в виде аргументов командной строки. На пятерку: Вариант 3. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в обратном порядке строк. Имена		

		файлов задаются в виде аргументов командной строки. Вариант 4. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в порядке убывания их длины. Имена файлов задаются в виде аргумен Потоковый ввод-вывод в C++. Разработать абстрактный базовый класс object, а для него предусмотреть чистые виртуальные методы. Разработанный ранее класс - ТД, сделать наследником от object. При этом переопределить чистые виртуальные методы класса object,наполнив их конкретным содержанием. Разработать класс - тип данных (ТД) с переопределением операторов. Разработать функции ввода, вывода, базовый набор содержательных операций над ТД (например, арифметических и сравнения) - 4 часа. Варианты типов данных: а) целые произвольной длины, представленные строкой цифр; б) то же, но во внутреннем двоичном представлении (динамический массив байтов); в) матрицы переменной размерности; г) разреженная матрица переменной размерности, ненулевые элементы представлены динамическим массивом с элементами (x, y, v); д) дроби; е) дата и время; ж) комплексные числа; з) разрежены матрицы переменной размерности, ненулевые элементы представлены односвязным списком. Разработать шаблонный класс (с использованием стандартной библиотеки шаблонов STL), реализующей определенную структуру хранения данных (т.н. "контейнерный" класс). Данные в контейнере хранятся либо в виде указателей на объекты, либо в виде самих объектов. Структуры хранения данных: а) двоичное дерево; б) массив; в) односвязный линейный список; г) двусвязный циклический список. Разработать		

		демонстрационную программу Создание и использование динамически присоединяемых библиотек. Создание консоли. Ввод и вывод в консоли. Обработка событий клавиатуры и мыши. Стандартный класс std::string. Конструкторы строк. Доступ к символам строк. Операции поиска. Операции вставки, замены и удаления символов строки. Операторы ввода-вывода строк. Таблица сообщений. Сообщения от клавиатуры. Сообщения от мыши. Управление исключениями. Понятие исключения. Обработка исключительных ситуаций. Генерирование исключений. Перехватывание исключений. Использование блоков try/catch. Неожиданные исключения и обработка завершения. Файлы программы для Windows. Типы данных в Windows. Обмен сообщениями между приложениями и ОС Windows. Понятие о Windows-APL. Структура приложения Windows. Определение класса окна. Регистрация класса окна. Создание и отображение окна. Цикл обработки сообщений. Пример простой Windows программы. Библиотека Microsoft Foundation Classes (MFC). Краткая характеристика библиотеки. Разработка приложений Windows с использованием MFC. Пример программы. Функции. Перегрузка функций. Функции. Аргументы по умолчанию. Цель: освоить стандартную библиотеку шаблонов (STL). Общие требования: " для обращения к элементам заданного контейнера обязательно использовать соответствующие итераторы. " Если задание предусматривает выполнение сортировки, ее необходимо выполнить с помощью алгоритма sort из STL. " Обработка ошибок должна быть реализована с помощью выбрасывания исключений - объектов класса Exception. " Работу программы проверить на файлах очень большого размера. Вариант 1. Заданы имена входного текстового файла (существует на диске) и выходного файла		

		(создается программой). Считать все строки входного файла в контейнер vector<AnsiString>. Отсортировать строки в контейнере в алфавитном порядке, преобразовать к верхнему регистру и записать их в выходной файл. Вариант 2. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Считать позиции начал строк входного файла в контейнер deque<int>. Используя эти данные и реверсивный итератор записать в выходной файл Цель: углубить знания стандартной библиотеки шаблонов (STL). Общие требования: " для обращения к элементам заданного контейнера обязательно использовать соответствующие итераторы. " Если задание предусматривает выполнение сортировки, поиска или перебора диапазона, эти действия необходимо выполнить с помощью алгоритмов библиотеки или встроенных в контейнеры функций STL " Обработка ошибок должна быть реализована с помощью выбрасывания исключений - объектов класса Exception. " Работу программы проверить на файлах очень большого размера. Вариант 1. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; Считать все строки заданного входного файла в структуру данных - список страниц. Страница - вектор из 24 строк, объектов класса AnsiString (последняя страница может быть неполной). Вывести на экран страницу с заданным номером. Вариант 2. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; Считать все строки заданного входного файла в структуру данных - список страниц. С Шаблоны функций. Перегрузка и специализации шаблонов функций. Шаблоны функций сортировки. Шаблоны классов. Специализация шаблонов классов. Статические члены шаблонов классов. Класс - шаблон для вектора. 1. Работа с		

		инспектором объектов. " Создать приложение с главным окном, появляющимся в центре экрана, с размерами 200 на 300 (высота, ширина). " Сменить название класса формы на MForm, сменить заголовок формы. " Запретить изменение размеров окна и распахивание во весь экран. " Экспериментальным путем выяснить за что отвечает каждое из свойств класс TForm. " Нарисовать собственную иконку используя редактор ресурсов и назначить ее приложению. 2. Обработка событий. " Доработать предыдущее приложение так, чтобы размеры и положение главной формы вычислялось автоматически при ее создании. Положение - правый нижний угол, высота и ширина равны четверти размеров экрана. (Событие OnCreate, объект Screen) " Поместить на форму стандартную кнопку. Сделать так, чтобы она "убегала" от курсора мыши. (Событие OnMouseMove, свойства Left, Top) 3. Диалоговая панель " Доработать приложение, поместив вторую кнопку и создав вторую форму. Сделать так, чтобы при нажатии на кнопку появлялась форма		

2. Методика оценки этапов формирования компетенций по дисциплине

Уровни сформированности компетенций проверяется при проведении мероприятий текущей аттестации (контроля) в процессе изучения дисциплины, указанных в таблице раздела 1.

В 2 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (работа) (РГЗ(Р)). Требования к выполнению РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

В 3 семестре обязательным этапом текущей аттестации является курсовая работа. Требования к выполнению курсовой работы, состав и правила оценки сформулированы в паспорте курсовой работы.

Промежуточная аттестация по **дисциплине** проводится в 2 семестре - в форме экзамена, в 3 семестре - в форме экзамена, который направлен на оценку сформированности компетенций ОПК-4, ОПК-5, ОПК-6 и соотнесенных с ними индикаторов. (см. таблицу раздела 1).

Общие правила выставления оценок текущей и промежуточной аттестации по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе дисциплины.

На основании приведенных далее критериев можно оценить уровни сформированности компетенций ОПК-4, ОПК-5, ОПК-6, закрепленных за дисциплиной.

3. Общая характеристика уровней освоения компетенций

Продвинутый. Теоретическое содержание курса освоено полностью. Студент демонстрирует систематическое и глубокое понимание учебного материала и способность к самостоятельному пополнению и обновлению знаний в ходе дальнейшей учебной работы и профессиональной деятельности. Сформированы необходимые навыки практической работы. Все учебные задания, предусмотренные программой обучения, выполнены качественно, без замечаний. Уровень сформированности компетенций и соотнесенных с ними индикаторов, закрепленных за дисциплиной, оценен числом баллов, входящим в диапазон продвинутого уровня.

Базовый. Теоретическое содержание курса освоено в объеме, необходимом для дальнейшей учебы и профессиональной деятельности. Навыки практической работы сформированы на базовом уровне. Все предусмотренные программой обучения учебные задания выполнены с небольшими погрешностями. Уровень сформированности компетенций и соотнесенных с ними индикаторов, закрепленных за дисциплиной, оценен числом баллов в пределах базового уровня.

Пороговый. Теоретическое содержание курса освоено, необходимом для дальнейшей учебы и профессиональной деятельности. Некоторые практические навыки работы с сформированы с незначительными пробелами. Учебные задания, предусмотренные программой обучения, выполнялись с ошибками, исправленными под руководством преподавателя. Уровень сформированности компетенций и соотнесенных с ними индикаторов, закрепленных за дисциплиной, оценен числом баллов в пределах порогового уровня.

Ниже порогового. Теоретическое содержание курса освоено фрагментарно. Необходимые навыки практической работы сформированы минимально. Большинство учебных заданий, предусмотренных программой обучения, не выполнены. Уровень сформированности компетенций и соотнесенных с ними индикаторов, закрепленных за дисциплиной, оценен числом баллов, входящих в диапазон ниже порогового уровня.