**Приложение 1 к РПД Современные технологии программирования
09.04.02 Информационные системы и технологии
Направленность (профиль): Информационные системы предприятий и
учреждений
Форма обучения – заочная
Год набора - 2018**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ
ДИСЦИПЛИНЫ (МОДУЛЯ)**

1.	Кафедра	Информатики и вычислительной техники
2.	Направление подготовки	09.04.02 Информационные системы и технологии
3.	Направленность (профиль)	Информационные системы предприятий и учреждений
4.	Дисциплина (модуль)	Современные технологии программирования
5.	Форма обучения	заочная
6.	Год набора	2018

1. Методические рекомендации

Приступая к изучению дисциплины, обучающемуся необходимо внимательно ознакомиться с тематическим планом занятий, списком рекомендованной литературы. Следует уяснить последовательность выполнения индивидуальных учебных заданий. Самостоятельная работа обучающегося предполагает работу с научной и учебной литературой, умение создавать тексты. Уровень и глубина усвоения дисциплины зависят от активной и систематической работы на лекциях, изучения рекомендованной литературы, выполнения контрольных письменных заданий.

При изучении дисциплины обучающиеся выполняют следующие задания:

- изучают рекомендованную научно-практическую и учебную литературу;
- выполняют задания, предусмотренные для самостоятельной работы.

Основными видами аудиторной работы обучающихся являются лекции и практические/семинарские занятия.

1.1. Методические рекомендации по организации работы обучающихся во время проведения лекционных занятий

В ходе лекций преподаватель излагает и разъясняет основные, наиболее сложные понятия темы, а также связанные с ней теоретические и практические проблемы, дает рекомендации на семинарское занятие и указания на самостоятельную работу.

Знакомство с дисциплиной происходит уже на первой лекции, где от обучающегося требуется не просто внимание, но и самостоятельное оформление конспекта. При работе с конспектом лекций необходимо учитывать тот фактор, что одни лекции дают ответы на конкретные вопросы темы, другие – лишь выявляют взаимосвязи между явлениями, помогая обучающемуся понять глубинные процессы развития изучаемого предмета как в истории, так и в настоящее время.

Конспектирование лекций – сложный вид вузовской аудиторной работы, предполагающий интенсивную умственную деятельность обучающегося. Конспект является полезным тогда, когда записано самое существенное и сделано это самим обучающимся. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Целесообразно вначале понять основную мысль, излагаемую лектором, а затем записать ее. Желательно запись осуществлять на одной странице листа или оставляя поля, на которых позднее, при

самостоятельной работе с конспектом, можно сделать дополнительные записи, отметить непонятные места.

Конспект лекции лучше подразделять на пункты, соблюдая красную строку. Этому в большой степени будут способствовать вопросы плана лекции, предложенные преподавателям. Следует обращать внимание на акценты, выводы, которые делает лектор, отмечая наиболее важные моменты в лекционном материале замечаниями «важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек, подчеркивая термины и определения.

Целесообразно разработать собственную систему сокращений, аббревиатур и символов. Однако при дальнейшей работе с конспектом символы лучше заменить обычными словами для быстрого зрительного восприятия текста.

Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть теоретическим материалом.

1.2. Методические рекомендации по подготовке к практическим занятиям (лабораторным /семинарам)

Подготовку к каждому практическому занятию обучающийся должен начать с ознакомления с его планом, отражающим содержание предложенной темы. Продумывание и изучение вопросов плана основывается на проработке текущего материала лекции, и изучения рекомендованной обязательной и дополнительной литературы. Новые понятия по изучаемой теме необходимо проработать и внести в глоссарий.

Результат такой работы должен проявиться в способности обучающегося свободно ответить на теоретические вопросы практикума, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении лабораторных заданий и контрольных работ.

В процессе подготовки к практическим занятиям, обучающимся необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал из-за лимита аудиторных часов. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме.

Практические занятия завершают изучение наиболее важных тем учебной дисциплины. Они служат для закрепления изученного материала, развития умений и навыков подготовки докладов, сообщений, приобретения опыта устных публичных выступлений, ведения дискуссии, аргументации и защиты выдвигаемых положений, а также для контроля преподавателем степени подготовленности обучающихся по изучаемой дисциплине. На практических занятиях обучающиеся совместно с преподавателем обсуждают выданные им задания, задают интересующие их вопросы и выполняют на компьютерах самостоятельно или в группах свои задания, используя программное обеспечение представленное в рабочей программе.

Семинар предполагает свободный обмен мнениями по избранной тематике. Он начинается со вступительного слова преподавателя, формулирующего цель занятия и характеризующего его основную проблематику. Затем, как правило, заслушиваются сообщения обучающихся. Обсуждение сообщения совмещается с рассмотрением намеченных вопросов. Сообщения, предполагающие анализ публикаций по отдельным вопросам семинара, заслушиваются обычно в середине занятия. Поощряется выдвижение и обсуждение альтернативных мнений. В заключительном слове преподаватель подводит

итоги обсуждения и объявляет оценки выступавшим обучающимся. В целях контроля подготовленности обучающихся и привития им навыков краткого письменного изложения своих мыслей преподаватель в ходе семинарских занятий может осуществлять текущий контроль знаний в виде тестовых заданий.

При подготовке к семинару обучающиеся имеют возможность воспользоваться консультациями преподавателя. Кроме указанных тем обучающиеся вправе, по согласованию с преподавателем, избирать и другие интересующие их темы.

Качество учебной работы обучающихся преподаватель оценивает с использованием технологической карты дисциплины, размещенной на сайте филиала МАГУ.

1.3. Методические рекомендации по работе с литературой

Работу с литературой целесообразно начать с изучения общих работ по теме, а также учебников и учебных пособий. Далее рекомендуется перейти к анализу монографий и статей, рассматривающих отдельные аспекты проблем, изучаемых в рамках курса, а также официальных материалов и неопубликованных документов (научно-исследовательские работы, диссертации), в которых могут содержаться основные вопросы изучаемой проблемы.

Работу с источниками надо начинать с ознакомительного чтения, т.е. просмотреть текст, выделяя его структурные единицы. При ознакомительном чтении закладками отмечаются те страницы, которые требуют более внимательного изучения.

В зависимости от результатов ознакомительного чтения выбирается дальнейший способ работы с источником. Если для разрешения поставленной задачи требуется изучение некоторых фрагментов текста, то используется метод выборочного чтения. Если в книге нет подробного оглавления, следует обратить внимание ученика на предметные и именные указатели.

Избранные фрагменты или весь текст (если он целиком имеет отношение к теме) требуют вдумчивого, неторопливого чтения с «мысленной проработкой» материала. Такое чтение предполагает выделение: 1) главного в тексте; 2) основных аргументов; 3) выводов. Особое внимание следует обратить на то, вытекает тезис из аргументов или нет.

Необходимо также проанализировать, какие из утверждений автора носят проблематичный, гипотетический характер и уловить скрытые вопросы.

Понятно, что умение таким образом работать с текстом приходит далеко не сразу. Наилучший способ научиться выделять главное в тексте, улавливать проблематичный характер утверждений, давать оценку авторской позиции – это сравнительное чтение, в ходе которого обучающийся знакомится с различными мнениями по одному и тому же вопросу, сравнивает весомость и доказательность аргументов сторон и делает вывод о наибольшей убедительности той или иной позиции.

Если в литературе встречаются разные точки зрения по тому или иному вопросу из-за сложности прошедших событий и правовых явлений, нельзя их отвергать, не разобравшись. При наличии расхождений между авторами необходимо найти рациональное зерно у каждого из них, что позволит глубже усвоить предмет изучения и более критично оценивать изучаемые вопросы. Знакомясь с особыми позициями авторов, нужно определять их схожие суждения, аргументы, выводы, а затем сравнивать их между собой и применять из них ту, которая более убедительна.

Следующим этапом работы с литературными источниками является создание конспектов, фиксирующих основные тезисы и аргументы. Можно делать записи на отдельных листах, которые потом легко систематизировать по отдельным темам изучаемого курса. Другой способ – это ведение тематических тетрадей-конспектов по одной какой-либо теме. Большие специальные работы монографического характера целесообразно конспектировать в отдельных тетрадях. Здесь важно вспомнить, что конспекты пишутся на одной стороне листа, с полями и достаточным для исправления и ремарок межстрочным расстоянием (эти правила соблюдаются для удобства

редактирования). Если в конспектах приводятся цитаты, то непременно должно быть дано указание на источник (автор, название, выходные данные, № страницы). Впоследствии эта информация может быть использована при написании текста реферата или другого задания.

Таким образом, при работе с источниками и литературой важно уметь:

- сопоставлять, сравнивать, классифицировать, группировать, систематизировать информацию в соответствии с определенной учебной задачей;
- обобщать полученную информацию, оценивать прослушанное и прочитанное;
- фиксировать основное содержание сообщений; формулировать, устно и письменно, основную идею сообщения; составлять план, формулировать тезисы;
- готовить и презентовать развернутые сообщения типа доклада;
- работать в разных режимах (индивидуально, в паре, в группе), взаимодействуя друг с другом;
- пользоваться реферативными и справочными материалами;
- контролировать свои действия и действия своих товарищей, объективно оценивать свои действия;
- обращаться за помощью, дополнительными разъяснениями к преподавателю, другим обучающимся.
- пользоваться лингвистической или контекстуальной догадкой, словарями различного характера, различного рода подсказками, опорами в тексте (ключевые слова, структура текста, предваряющая информация и др.);
- использовать при говорении и письме перифраз, синонимичные средства, словоописания общих понятий, разъяснения, примеры, толкования, «словотворчество»;
- повторять или перефразировать реплику собеседника в подтверждении понимания его высказывания или вопроса;
- обратиться за помощью к собеседнику (уточнить вопрос, переспросить и др.);
- использовать мимику, жесты (вообще и в тех случаях, когда языковых средств не хватает для выражения тех или иных коммуникативных намерений).

1.4. Методические рекомендации по подготовке к сдаче экзамена

Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, обучающийся ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене обучающийся демонстрирует то, что он приобрел в процессе изучения дисциплины.

В условиях применяемой в МАГУ балльно-рейтинговой системы подготовка к экзамену включает в себя самостоятельную и аудиторную работу обучающегося в течение всего периода изучения дисциплины и непосредственную подготовку в дни, предшествующие экзамену по разделам и темам дисциплины.

При подготовке к экзамену обучающимся целесообразно использовать не только материалы лекций, а и рекомендованные основную и дополнительную литературу.

При подготовке к промежуточной аттестации целесообразно:

- внимательно изучить перечень вопросов и определить, в каких источниках находятся сведения, необходимые для ответа на них;
- внимательно прочитать рекомендованную литературу;
- составить краткие конспекты ответов (планы ответов).

Качество учебной работы обучающихся преподаватель оценивает с использованием технологической карты дисциплины, размещенной на сайте филиала МАГУ.

2. Планы лабораторных работ:

Лабораторная работа № 1. Методы поиска и структуры данных.

План:

1. Хэширование, постановка задачи.
2. Хэш-функции.
3. Методы разрешения коллизий в хэшировании.
4. Trie-дерево. Реализация trie-дерева.

Литература: [3, с. 942-956].

Вопросы для групповой дискуссии:

1. Что такое хэширование?
2. Перечислите основные хэш-функции.
3. Когда возникает коллизия в хэшировании?
4. Какие вы знаете методы разрешения коллизий.
5. Как происходит вставка, удаление и поиск элемента в префиксном (Trie) дереве?
6. Приведите пример алгоритма сжатия данных, с помощью бинарного дерева.

Задание для самостоятельной работы

1. Реализовать модуль работы с хэш таблицей, согласно варианту задания.
2. Реализовать Trie-дерево, согласно варианту задания.

Примерные варианты:

1. Хэширование. Задан текстовый файл сформировать список слов, употребляющихся в тексте более 5 раз.
2. Хэширование. Задан набор записей следующей структуры: табельный номер, ФИО, заработная плата. По табельному номеру найти остальные сведения.
3. Из файла, содержащего текст на русском языке, слова помещаются в хэш-таблицу. Определить среднее число коллизий для нескольких заданных хэш-функций.
4. В Trie – дереве подсчитать количество слов, которые содержат определенное количество согласных.
5. Дан текстовый файл, который содержит слова, возможно, повторяющиеся. Распечатать их в алфавитном порядке, указав число вхождений каждого слова в текст.
6. Дано Trie – дерево и некоторое слово. Вывести все формы данного слова, содержащиеся в дереве. Формой слова считается некоторое другое слово, отличающееся от заданного не более чем на n последних символов, где n – вводит пользователь.

Лабораторная работа № 2. Шаблоны классов. Стандартная библиотека шаблонов.

План:

1. Идея обобщенного программирования.
2. Определение и использование шаблона класса.
3. Параметры шаблонов.
4. Использование классов функциональных объектов для настройки шаблонных классов.
5. Обработка исключительных ситуаций.
6. Основные концепции STL: контейнеры, итераторы, алгоритмы.
7. Использование последовательный контейнеров.
8. Использование ассоциативных контейнеров: множества, словари.

Литература: [3, с. 942-956].

Вопросы для групповой дискуссии:

1. В чем состоит идея обобщенного программирования?
2. Как выглядит определение шаблонного класса на языке C++?
3. Что означает и инстанцирование шаблона?
4. Какой класс называют функциональным?
5. Из каких компонент состоит стандартная библиотека шаблонов?

6. Для чего нужны итераторы? Приведите пример использования итератора.
7. Приведите пример последовательного и ассоциативного контейнера.

Задание для самостоятельной работы:

Задание 1.

1. Требуется создать шаблон некоторого целевого класса А, возможно, реализованный с применением некоторого серверного класса В. Это означает, что объект класса В используется как элемент класса А. В качестве серверного класса может быть указан либо класс, созданный программистом либо класс из стандартной библиотеки - например, `std::vector`. Варианты целевых или серверных классов, создаваемых программистом, приведены в таблице:

Имя класса	Назначение
Vect	одномерный динамический массив
List	двунаправленный список
Stack	стек
BinaryTree	бинарное дерево
Queue	односторонняя очередь
Deque	двусторонняя очередь
Set	множество
SparseArray	разряженный массив

2. Во всех вариантах необходимо предусмотреть генерацию и обработку исключений для возможных ошибочных ситуаций.

3. Во всех вариантах показать в клиенте `main()` использование созданного класса, включая ситуации, приводящие к генерации исключений. Показать инстанцирование шаблона для типов `int`, `double`, `std::string`.

4. Варианты заданий приведены в таблице:

№	Целевой шаблонный класс	Реализация с применением
1	Vect	<code>std::list</code>
2	Stack	Динамический массив
3	Stack	Vect
4	Set	Динамический массив
5	Queue	List
6	Deque	Vect

Задание 2.

1. Написать программу, в которой используются контейнерные классы STL, согласно варианту задания.

2. Примерные варианты заданий:

- Написать программу, отыскивающую проход по лабиринту, с использованием контейнерного класса `stack` из STL. Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Программа находит проход через лабиринт, двигаясь от заданного входа. После отыскания прохода программа выводит найденный путь в виде координат квадратов.

- Написать программу, моделирующую управление каталогом в файловой системе. Для каждого файла в каталоге содержатся следующие сведения: имя файла, дата создания, количество обращений к файлу. Программа должна обеспечивать:

- начальное формирование каталога файлов;
- вывод каталога файлов;
- удаление файлов, дата создания которых раньше заданной;
- выборку файла с наибольшим количеством обращений.

Выбор моделируемой функции должен осуществляться с помощью меню. Для представления каталога использовать контейнерный класс list из STL.

- Написать программу, реализующую игру «Крестики-нолики» между двумя игроками: пользователем и компьютером (роботом).

Лабораторная работа № 3. Обработка многопоточных данных. Безопасное взаимодействие.

План:

1. Процессы и потоки в операционной системе.
2. Важнейшие средства синхронизации.
3. Блокирование и потоковая безопасность.
4. Потокобезопасность и типы .NET Framework.
5. Оператор lock.
6. Классы: EventWaitHandle, Mutex и Semaphore.
7. Создание межпроцессных EventWaitHandle.
8. Сценарий работы с потоками – фоновая обработка задач из очереди.

Литература: [1, с. 169-216].

Вопросы для групповой дискуссии:

2. В чем отличие процессов от потоков?
3. В чем отличие классов AutoResetEvent и ManualResetEvent?
4. Для чего нужен метод WaitOne класса AutoResetEvent?
5. С чем образно можно сравнить класс AutoResetEvent?
6. В чем состоит назначение метода Set класса AutoResetEvent?
7. В чем состоит идея безопасного взаимодействия?
8. Приведите пример безопасного взаимодействия потоков.
9. Приведите пример небезопасного взаимодействия потоков.
10. Какие проблемы могут возникать при взаимодействии нескольких потоков.
11. С чем образно можно сравнить класс Semaphore?

Задание для самостоятельной работы

1. Реализовать, используя механизм общей памяти, передачу данных между двумя потоками. Первый поток проводит вычислительную работу, оперируя данными, считываемых из ранее заготовленного файла. Результат вычислений для каждого цикла вычислений появляется асинхронно. Время появления результата на каждом цикле вычислений зависит от множества различных факторов. При появлении результата вычислений первый поток должен передавать его второму потоку для дальнейших вычислений. Исходный файл – содержит ~10000 строк чисел типа float. Первый поток считывает их последовательно по ~1000 и производит над ними вычислительные операции (например нахождение среднеквадратичного). Результат каждого вычислительного цикла передается второму потоку. Второй поток также производит над ним мат. операции и конечный результат пишет в файл.
2. Реализовать безопасную синхронизацию потоков.
3. Требования реализации:
 - использование системных объектов синхронизации обязательно;
 - все записи считывания и записи в файл дублировать на консоль (форму);
 - моменты переключения между потоками сообщать на консоль (форму);
 - программа должна быть построена только в ОО стиле, на языке C#;
 - отчет к работе оформить в doc-формате;
 - отчет: описание классов (поля, методы); результаты тестовых прогонов программы.
4. Подготовиться к ответам на вопросы.

Лабораторная работа № 4. Создание библиотеки классов и клиентского приложения в среде Visual Studio на языке C#.

План:

1. Выбор идеи проекта.
2. Разработка диаграммы вариантов использования.
3. Разработка сценариев для вариантов использования.
4. Модель взаимодействия объектов проектируемой системы на диаграмме Sequence.
5. Первоначальный макет диаграммы классов. Установка атрибутов, методов.
6. Паттерны проектирования. Каталог паттернов GoF.
7. Выбор паттерна проектирования для своего проекта.
8. Модификация диаграммы классов в соответствии с выбранным паттерном проектирования каталога: GoF.
9. Архитектурные паттерны проектирования.
10. Создание каркаса библиотеки классов в среде разработки Visual Studio, на основе разработанной диаграммы классов.
11. Прямое и обратное проектирование.

Литература: [2, с.66-94]; [4, с. 4-51].

Вопросы для групповой дискуссии:

1. Перечислите основные принципы объектно-ориентированного подхода.
2. Что означает принципы абстрагирования и инкапсуляции?
3. Перечислите основные диаграммы языка UML.
4. Какие цели преследует построение Use Case диаграмм?
5. Перечислите основные элементы диаграмм Use Case.
6. Перечислите основные элементы диаграмм Sequence.
7. Какие связи применяются на диаграммах классов?
8. В чем состоит принцип компонентно-ориентированной разработки.
9. Дайте определение интерфейса.
10. Для чего используются паттерны проектирования?
11. Перечислите категории каталога GoF и несколько паттернов из разных категорий.

Задание для самостоятельной работы:

1. Для выбранного варианта задания разработать функциональные требования и диаграмму классов по стандарту UML.
2. Выполнить генерацию кода по созданной диаграмме классов и перенести его в среду программирования.
3. Разработать проект, в среде программирования, в виде библиотеки классов. Созданная библиотека классов, должна реализовывать запроюктированные, варианты использования.
4. Создать клиентское приложение для тестирования основных возможностей, созданной библиотеки классов.
5. Предусмотреть обработку исключительных ситуаций и выдачу диагностических сообщений об ошибочных и аварийных состояниях системы.
6. Подготовиться к ответам на вопросы.