

**Приложение 1 к РПД Объектно-ориентированный анализ и
программирование
09.03.02 Информационные системы и технологии
Направленность (профиль) – Информационные системы и технологии
Форма обучения – заочная
Год набора - 2014**

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

1.	Кафедра	Информатики и вычислительной техники
2.	Направление подготовки	09.03.02 Информационные системы и технологии
3.	Направленность (профиль)	Информационные системы и технологии
4.	Дисциплина (модуль)	Объектно-ориентированный анализ и программирование
5.	Форма обучения	заочная
6.	Год набора	2014

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 лабораторные 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 лабораторным занятиям

Лабораторные 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 лабораторных 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. Каждое выполненное задание обучающийся обязан оформить в виде отчета и защитить его.

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, предвещающая информация и др.);
- использовать при говорении и письме перифраз, синонимичные средства, слова-описания 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.

1.5. Методические рекомендации по созданию презентации

Алгоритм создания презентации:

1 этап – определение цели презентации

2 этап – подробное раскрытие информации,

3 этап – основные тезисы, выводы.

Следует использовать 10-15 слайдов. При этом:

- первый слайд – титульный. Предназначен для размещения названия презентации, имени докладчика и его контактной информации;
- на втором слайде необходимо разместить содержание презентации, а также краткое описание основных вопросов;
- оставшиеся слайды имеют информативный характер.

Обычно подача информации осуществляется по плану: тезис – аргументация – вывод.

Требования к оформлению и представлению презентации:

1. Читабельность (видимость из самых дальних уголков помещения и с различных устройств), текст должен быть набран 24-30-ым шрифтом.
2. Тщательно структурированная информация.
3. Наличие коротких и лаконичных заголовков, маркированных и нумерованных списков.
4. Каждому положению (идее) надо отвести отдельный абзац.
5. Главную идею надо выложить в первой строке абзаца.
6. Использовать табличные формы представления информации (диаграммы, схемы) для иллюстрации важнейших фактов, что даст возможность подать материал компактно и наглядно.
7. Графика должна органично дополнять текст.
8. Выступление с презентацией длится не более 10 минут;

1.6. Методические рекомендации по составлению глоссария

1. Внимательно прочитайте и ознакомьтесь с текстом. Вы встретите в нем много различных терминов, которые имеются по данной теме.

2. После того, как вы определили наиболее часто встречающиеся термины, вы должны составить из них список. Слова в этом списке должны быть расположены в строго алфавитном порядке, так как глоссарий представляет собой не что иное, как словарь специализированных терминов.

3. После этого начинается работа по составлению статей глоссария. Статья глоссария - это определение термина. Она состоит из двух частей: 1. точная формулировка термина в именительном падеже; 2. содержательная часть, объемно раскрывающая смысл данного термина.

При составлении глоссария важно придерживаться следующих правил:

- стремитесь к максимальной точности и достоверности информации;
- старайтесь указывать корректные научные термины и избегать всякого рода жаргонизмов. В случае употребления такового, дайте ему краткое и понятное пояснение;
- излагая несколько точек зрения в статье по поводу спорного вопроса, не принимайте ни одну из указанных позиций. Глоссарий - это всего лишь констатация имеющихся фактов;
- также не забывайте приводить в пример контекст, в котором может употребляться данный термин;
- при желании в глоссарий можно включить не только отдельные слова и термины, но и целые фразы.

1.7. Методические рекомендации для занятий в интерактивной форме

В учебном процессе, помимо чтения лекций и аудиторных занятий, используются интерактивные формы. В сочетании с внеаудиторной работой это способствует формированию и развитию профессиональных навыков обучающихся.

Интерактивное обучение представляет собой способ познания, осуществляемый в формах совместной деятельности обучающихся, т.е. все участники образовательного процесса взаимодействуют друг с другом, совместно решают поставленные проблемы, моделируют ситуации, обмениваются информацией, оценивают действие коллег и свое собственное поведение, погружаются в реальную атмосферу делового сотрудничества по разрешению проблем.

В курсе изучаемой дисциплины «Объектно-ориентированный анализ и программирование» в интерактивной форме часы используются в виде: групповой дискуссии.

Тематика занятий с использованием интерактивных форм

№ п/п	Тема	Интерактивная форма	Часы, отводимые на интерактивные формы	
			Лекции	Лабораторные занятия
1.	Язык программирования C#.	Групповая дискуссия	-	2
ИТОГО			2 часа	

План лабораторных работ:

3 семестр

Лабораторная работа № 1. Классы и объекты в C++

План:

1. Основные понятия и принципы ООП. Классы и объекты.
2. Конструкторы и деструкторы. Конструкторы с параметром. Конструктор копирования.
3. Инкапсуляция: скрытые поля и методы. Методы set и get.
4. Указатель this.
5. Заголовочные файлы и файлы реализации в программе на C++.

Литература: [1, с. 119-169].

Вопросы для групповой дискуссии:

1. Что такое инкапсуляция ?
2. Дайте формальное определение класса и объекта класса.
3. Что такое конструктор?
4. Что такое конструктор по умолчанию? Назовите его особенности.
5. Какие типы конструкторов вы можете назвать?
6. Для чего необходимы деструкторы?
7. Для чего в программе на C++ используются h-файл и сpp-файл?

Задание для самостоятельной работы

1. Определить пользовательский класс в соответствии с вариантом.
2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.
3. Определить в классе деструктор.
4. Определить в классе компоненты-функции для просмотра и установки полей данных.
5. Определить указатель на экземпляр класса.
6. Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения (какой объект какой конструктор или деструктор вызвал).
7. Показать в программе использование указателя на объект.
8. Лабораторная должна состоять из трёх файлов:
 - заголовочный h-файл с определением класса,
 - сpp-файл с реализацией класса,
 - сpp-файл демонстрационной программы.

Лабораторная работа № 2. Методы класса, конструкторы и деструкторы. Перегрузка методов.

План:

1. Методы класса, типы методов.
2. Спецификаторы доступа к полям класса.
3. Перегрузка методов класса.
4. Поточные классы в языке C++.

Литература: [1, с. 119-169].

Вопросы для групповой дискуссии:

1. Перечислите спецификаторы доступа к полям класса.
2. Перечислите типы методов класса.
3. Для чего используется перегрузка методов?
4. Какие методы используются для работы потоковыми классами.

Задание для самостоятельной работы

1. Выбрать класс в соответствии с вариантом.
2. Определить и реализовать в классе конструкторы, деструктор, функции Input (ввод с клавиатуры) и Print (вывод на экран), перегрузить метод присваивания.
3. Написать программу тестирования класса и выполнить тестирование.
4. Дополнить определение класса заданными перегруженными операциями (в соответствии с вариантом).
5. Реализовать эти операции. Выполнить тестирование.

Лабораторная работа № 3. Перегрузка операторов.

План:

1. Перегрузка операторов.
2. Файловые потоки. Виды потоковых устройств.
3. Потоки и типы, определенные пользователем.
4. Ошибки потоков.

Литература: [1, с. 119-169].

Вопросы для групповой дискуссии:

1. Как выполняется перегрузка операторов?
2. Как происходит чтение из файла, с помощью файлового потока C++?
3. Как выполняется запись в файл, с помощью файлового потока C++?

Задание для самостоятельной работы

1. Определить пользовательский класс в соответствии с вариантом задания.
2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.
3. Определить в классе деструктор.
4. Определить в классе компоненты-функции для просмотра и установки полей данных.
5. В демонстрационной программе показать использование перегруженных операторов. Каждый вызов перегруженного оператора должен сопровождаться выдачей соответствующего сообщения на экран.
6. В демонстрационной программе должны создаваться и разрушаться объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения (какой объект какой конструктор или деструктор вызвал).
7. Лабораторная работа должна состоять из трёх файлов:
 - заголовочный h-файл с определением класса,
 - сpp-файл с реализацией класса,
 - сpp-файл демонстрационной программы.

Литература: [1, с. 169-181].

Лабораторная работа № 4. Наследование и виртуальные функции

План:

1. Принцип наследования в ООП.
2. Множественное наследование.
3. Построение иерархия классов. Родительские и дочерние классы.
4. Принцип полиморфизма в ООП. Виды полиморфизма.
5. Виртуальные функции.
6. Абстрактные классы.

Литература: [1, с. 169-181].

Написать программу, в которой создается иерархия классов. Включить полиморфные объекты в связанный список, используя статические компоненты класса. Показать использование виртуальных функций.

Вопросы для групповой дискуссии:

1. В чем заключается принцип наследования в ООП.
2. Какой класс называют абстрактным?
3. Для чего нужны абстрактные классы?

4. Что такое виртуальные функции?
5. Что такое чисто виртуальные функции, и как они определяются в программе на C++?
6. Что такое динамическое связывание?
7. В чем отличие статического связывания от динамического?
8. В чем заключается принцип полиморфизма в ООП?

Задание для самостоятельной работы

1. Для определения иерархии классов связать отношением наследования классы, приведенные в задании. Из перечисленных классов выбрать один, который будет стоять во главе иерархии. Это абстрактный класс.
2. Определить в классах все необходимые конструкторы и деструктор.
3. Компонентные данные класса специфицировать как **protected**.
4. Пример определения статических компонентов:

```
static person* begin; // указатель на начало списка
static void print(void); // просмотр списка
```
5. Статическую компоненту-данные инициализировать вне определения класса, в глобальной области.
6. Для добавления объекта в список предусмотреть метод класса, т.е. объект сам добавляет себя в список. Например, `a.Add()` – объект `a` добавляет себя в список.
7. Включение объекта в список можно выполнять при создании объекта, т.е. поместить операторы включения в конструктор. В случае иерархии классов, включение объекта в список должен выполнять **только** конструктор базового класса. Вы должны продемонстрировать оба этих способа.
8. Список просматривать путем вызова виртуального метода **Show** каждого объекта.
9. Статический метод просмотра списка вызывать не через объект, а через класс.
10. Определение классов, их реализацию, демонстрационную программу поместить в отдельные файлы.

4 семестр

Лабораторная работа № 1. Объектно-ориентированное программирование в C#.

План:

1. Разбор простой программы на C#.
1. Формальное определение класса в C#.
2. Варианты метода `Main()`.
3. Модификаторы доступа C#.
4. Построение цепочки вызовов конструкторов с использованием `this`.
5. Средства инкапсуляции C#: инкапсуляция с использованием традиционных методов доступа и изменения; инкапсуляция с использованием свойств `.NET`; свойства, доступные только для чтения и только для записи; статические свойства. Статические конструкторы
6. Наследование в C#. Работа с конструктором базового класса. Множественное наследование. Ключевое слово `sealed`
7. Поддержка полиморфизма в C#. Ключевые слова `virtual` и `override`. Абстрактные классы. Полиморфный интерфейс. Соккрытие методов
8. Правила приведения к базовому и производному классу. Ключевое слово `as`
9. Применение модели включения – делегирования. Ключевое слово `is`

Литература: [4, с. 28-160].

Вопросы для групповой дискуссии:

8. Какую роль играет метод `Main()` в программе на C# ?
9. Что является объектом приложения в программе на C# ?

10. Что означает ключевое слово `static` в сигнатуре метода `Main()`?
11. Является ли `C#` чувствительным к регистру языком программирования?
12. Предложите различные варианты метода `Main()`.
13. Дайте формальное определение класса.
14. Что такое конструктор?
15. Что такое конструктор по умолчанию? Назовите его особенности.
16. В каком случае конструктор по умолчанию не будет создан автоматически?
17. Что означает ключевое слово `this`?
18. Перечислите модификаторы доступа в `C#`.
19. Что означает принцип инкапсуляции?
20. Какие возможности для инкапсуляции предоставляет `C#`? Предложите несколько способов.
21. Какие свойства называются статическими?
22. Для чего применяют статические конструкторы?
23. Что означает принцип наследования?
24. Для чего применяется ключевое слово `base`?
25. Как в `C#` взаимодействуют конструкторы базового и производного классов?
26. Возможно ли множественное наследование в `C#`?
27. Каков смысл ключа `sealed`?
28. Что означает принцип полиморфизма?
29. Если метод помечен как `virtual`, то, что это означает?
30. Когда применяют ключевое слово `override`?
31. Что такое абстрактный класс и для чего его используют?
32. Что такое полиморфный интерфейс?
33. Для каких целей применяют абстрактные методы?
34. Что такое сокрытие (`shadowing`) методов?
35. Перечислите правила приведения к базовому и производному классу.
36. Что означает ключевое слово `as`?
37. Что означает ключевое слово `is`?
38. Что означает модель включения-делегирования?
39. Приведите пример внешнего и внутреннего класса в модели включения-делегирования.
40. Какие типы в `C#` называют вложенными?

Задание для самостоятельной работы

1. Выполнить в среде разработки Visual Studio следующие проекты (консольные приложения):
 - пример 1 (*Модель сотрудника некоторой организации*);
 - пример 2 (*Иерархия геометрических фигур*);
 - пример 3 (*Программная модель автомобиля*).
2. Подготовиться к ответам на вопросы.

Лабораторная работа № 2. Введение в разработку Windows приложений в IDE Visual Studio на языке `C#`. Элементы управления в Windows – приложениях

План:

1. Интегрированная среда разработки приложений Visual Studio .NET.
2. Создание приложения типа Windows Application.
3. Режимы дизайна и кода.
4. Свойства проекта.
5. Компиляция программ.
6. Ввод и форматирование текста.

7. Форматирование ячеек и таблиц.
8. Ввод формул. Вычислительные возможности электронных таблиц.
9. Элементы управления. Работа с элементами управления.
10. Создание главного меню, контекстного меню, диалоговых окон.
11. Удаление элементов управления в режиме работы приложения.
12. Создание пользовательских элементов управления.

Литература: [2, с. 13-177].

Вопросы для групповой дискуссии:

1. Какие компоненты содержит окно Solution Explorer в IDE Visual Studio?
2. В чем состоит назначение окна Class View в IDE Visual Studio?
3. Какие возможности для разработчика представляет окно Properties в IDE Visual Studio?
4. Для чего служит окно Object Browser в IDE Visual Studio?
5. Что означают свойства: Name, AcceptButton, BackgroundImage, BackgroundImage, ControlBox, DrawGrid, StartPosition, WindowState формы?
6. Что означает режим дизайна и режим кода?
7. Можно ли добавлять код после события InitializeComponent() ?
8. Каким образом происходит обработка событий в приложениях типа Windows application в IDE Visual Studio?
9. Перечислите основные события для кнопки (button)?
10. Для чего в программах необходимы элементы управления?
11. Какие элементы управления входят в группу: командных объектов, текстовых объектов, переключателей, контейнеров, диалоговых окон, меню?
12. Какое приложение называется MDI-приложением?
13. Какие необходимо произвести действия для создания в проекте новой формы?
14. Для чего необходимы компоненты: RichTextBox, IsMdiContaine, ContextMenuStrip?
15. Как добавить на форму элемент управления в режиме работы приложения (Run time)?
16. Как можно удалить элемент управления во время работы приложения?
17. Приведите пример добавления пункта в контекстное меню в режиме работы приложения.

Задание для самостоятельной работы

1. Реализовать и протестировать приложение, работающее с рассмотренными выше элементами управления.
2. Подготовиться к ответам на вопросы.

Лабораторная работа № 3. Диаграммы классов в Visual Studio и разработка динамически подключаемых библиотек в среде .NET.

План:

1. Диаграмма классов стандарта UML в среде Visual Studio.
2. Что такое рефлексия типов.
3. Класс Type. Получение объекта класса Type. Возможности класса Type
4. Типы пространства имен System.Reflection.
5. Загрузка сборки. Вывод информации о типах в сборке. Вывод информации о членах класса. Вывод информации о параметрах метода. Применение позднего связывания.
6. Класс System.Activator.
7. Применение динамическихборок.
8. Пространство имен System.Reflection.Emit.
9. Создаем динамическую сборку. Как использовать динамическую сборку?

Литература: [4, с. 201-240].

Вопросы для групповой дискуссии:

1. Для чего необходима диаграмма классов?
2. Какие элементы присутствуют на диаграммах классов?
3. Перечислите основные типы взаимодействия классов и как они изображаются на диаграммах.
4. Для чего необходим класс Type?
5. В чем состоит методика создания динамически подключаемых библиотек в среде .NET?

Задание для самостоятельной работы

1. Согласно варианту задания разработать диаграмму классов
2. На основе, разработанной диаграммы классов, создать каркас библиотеки классов (*описания классов и их связи*) в среде Visual Studio, используя прямое и обратное проектирование (*получение кода по модели; получение модели по коду*).
3. По исходному коду каркаса библиотеки классов получить схему в виде диаграммы классов.
4. Разработать программную реализацию библиотеки классов (*в виде *.dll файла*) на языке C# в среде Visual Studio. Созданная библиотека классов должна реализовывать все запрограммированные ранее, варианты использования системы.
5. Создать клиентское приложение для тестирования основных возможностей, созданной библиотеки классов. Предусмотреть обработку исключительных ситуаций и выдачу диагностических сообщений об ошибочных и аварийных состояниях системы.
6. Подготовить отчет: диаграмма классов, краткое описание программной реализации библиотеки классов и тестирующего приложения.
7. Подготовиться к ответам на вопросы.

Лабораторная работа № 4. Обработка многопоточных данных

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: [3, с. 44-179].

Вопросы для групповой дискуссии:

1. В чем состоит причина разработки многопоточных приложений?
2. В чем отличие классов AutoResetEvent и ManualResetEvent?
3. Для чего нужен метод WaitOne класса AutoResetEvent?
4. С чем образно можно сравнить класс AutoResetEvent?
5. В чем состоит назначение метода Set класса AutoResetEvent?
6. В чем состоит идея безопасного взаимодействия? Приведите пример безопасного взаимодействия потоков.

Задание для самостоятельной работы

1. Реализовать, используя механизм общей памяти, передачу данных между двумя потоками. Первый поток проводит вычислительную работу, оперируя данными, считываемыми из ранее заготовленного файла. Результат вычислений для каждого цикла

вычислений появляется асинхронно. При появлении результата вычислений первый поток должен передавать его второму потоку для дальнейших вычислений. Второй поток также производит над результатом математические операции и конечный результат пишет в файл.

2. Требования реализации:

- 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#;
- подготовить отчет: описание классов (поля, методы); результаты тестовых прогонов программы.

3. Подготовиться к ответам на вопросы.