

Федеральное государственное образовательное бюджетное учреждение
высшего образования
**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

Калужский филиал Финуниверситета

**Факультет «Экономика и бизнес - технологии»
Кафедра «Бизнес – информатика и информационные технологии»**

КОМПЬЮТЕРНЫЙ ПРАКТИКУМ

Методические указания к лекциям
для студентов, обучающихся по направлению 38.03.05 Бизнес - информатика
профиль «ИТ – менеджмент в бизнесе»
очная и заочная форма обучения

*Одобрено кафедрой «БИ и ИТ»
(протокол № 13 от 26 июня 2018 г.)*

Калуга 2018

Методические указания разработал:

Кандидат физико – математических наук, доцент С.В. Пономарев

Методические указания к лекционным занятиям по дисциплине «Компьютерный практикум» предназначены для студентов, обучающихся по направлению подготовки 38.03.05 Бизнес - информатика, профиль «ИТ – менеджмент в бизнесе» по заочной форме обучения.

В методических указаниях излагаются требования к структуре и содержанию конспекта лекционных занятий, представлены рекомендации по написанию лекционной работы, варианты лекционных заданий.

Содержание

Введение.....	4
1. Методические указания по подготовке и выполнению лекционной работы.....	4
1.1 Структура лекционной работы.....	4
1.2 Требования к содержанию и оформлению текста лекционной работы.....	5
2. Определение варианта лекционной работы.....	5
3. Варианты тем лекционной работы.....	6
4. Критерии оценки.....	7
5. Список рекомендуемой литературы, интернет – ресурсов и справочно-правовых систем.....	8
6. Приложение:.....	9
- Титульный лист.....	9
- Пример выполнения варианта лекционной работы.....	9

Введение

Лекция является главным звеном дидактического цикла обучения. Ее цель - формирование ориентировочной основы для последующего усвоения студентами учебного материала. В ходе лекции преподаватель, применяя методы устного изложения и показа, передает обучаемым знания по основным, фундаментальным вопросам изучаемой дисциплины. Назначение лекции состоит в том, чтобы доходчиво, убедительно и доказательно раскрыть основные теоретические положения изучаемой науки, нацелить обучаемых на наиболее важные вопросы, темы, разделы учебной дисциплины, дать им установку и оказать помощь в овладении научной методологией (методами, способами, приемами) получения необходимых знаний и применения их на практике.

Одним из неоспоримых достоинств лекции является то, что новизна излагаемого материала соответствует моменту ее чтения, в то время как положения учебников, учебных пособий относятся к году их издания. Кроме того, на лекции личное общение преподавателя со студентами предоставляет большие возможности для реализации воспитательных целей.

1. Методические указания по подготовке и выполнению лекционной работы

1.1 Структура лекционной работы

Составление конспекта лекционных занятий

Хорошо составленный конспект помогает усвоить материал. В конспекте кратко излагается основная сущность учебного материала, приводятся необходимые обоснования, табличные данные, схемы, эскизы, расчеты и т.п. Конспект целесообразно составлять целиком на тему. При этом имеется возможность всегда дополнять составленный конспект вырезками и выписками из журналов, газет, статей, новых учебников, брошюр по обмену

опытом, данных из Интернета и других источников. Таким образом конспект становится сборником необходимых материалов, куда студент вносит всё новое, что он изучил, узнал. Такие конспекты представляют, большую ценность при подготовке к занятиям.

1. Первичное ознакомление с материалом изучаемой темы по тексту учебника, картам, дополнительной литературе.
2. Выделение главного в изучаемом материале, составление обычных кратких записей.
3. Подбор к данному тексту опорных сигналов в виде отдельных слов, определённых знаков, графиков, рисунков.
4. Продумывание схематического способа кодирования знаний, использование различного шрифта и т.д.
5. Составление опорного конспекта.

1.2 Требования к содержанию и оформлению текста лекционной работы

Объём конспекта – 1 тетрадная страница на один раздел или один лист формата А 4. Логика изложения (наличие схем, количество смысловых связей между понятиями). Наглядность (наличие рисунков, символов, и пр.; аккуратность выполнения, читаемость конспекта. Грамотность (терминологическая и орфографическая). Отсутствие связанных предложений, только опорные сигналы – слова, словосочетания, символы. Самостоятельность при составлении.

2. Определение варианта лекционной работы

Интерфейс - это контракт, обеспечивающий определенное поведение класса или структуры. Когда класс реализует интерфейс, он как бы говорит потенциальному пользователю: “Я гарантирую, что поддерживаю методы, свойства, события и индексаторы этого интерфейса”.

Интерфейс- семейство явно описанных как public - методов и свойств, которые сгруппированы в единое целое и инкапсулируют какую-либо определенную функциональную возможность.

После того как интерфейс определен, его можно реализовать в некотором классе. Это означает, что класс будет поддерживать все свойства и члены, определяемые данным интерфейсом. Интерфейсы не могут существовать сами по себе. таким же образом, как создается экземпляр класса. У интерфейса нет кода, который бы реализовал его члены; он просто описывает эти члены. Их реализация должна находиться в классах, в которых реализован данный интерфейс.

Итак, интерфейс представляет собой альтернативу абстрактному классу в смысле создания контрактов между классами и их пользователями. Эти контракты создаются с использованием ключевого слова interface, которое объявляет ссылочный тип, инкапсулирующий контракт. Синтаксически интерфейс подобен классу, имеющему только абстрактные методы. Абстрактный класс является базовым для семейства производных классов, в то время как интерфейсы задуманы для “смешивания” их с прочими деревьями наследования. Когда класс реализует интерфейс, он должен реализовать все методы этого интерфейса. По сути дела, он соглашается выполнить контракт, определенный интерфейсом.

Правила использования интерфейсов:

- Нельзя "создать экземпляр интерфейса"
- Интерфейс не может содержать в себе никакого кода
- У класса может быть несколько интерфейсов, и несколько классов могут поддерживать один и тот же интерфейс.
- У интерфейсов нет никаких конструкторов
- Определению интерфейса не разрешено содержать перегрузки операций
- Не допускается указывать никаких модификаторов доступа, а также модификаторов virtual,static

3. Варианты тем лекционной работы

- Определение алгоритма и его свойства
- Методы разработки алгоритма: метод частных целей; метод подъема
- Программирование с отходом назад
- Алгоритмизация вычислительных процессов: способы и изобразительные средства записи алгоритмов
- Линейные процесс обработки информации

- Разветвляющийся процесс обработки информации
- Циклический процесс обработки информации
- Циклы с известным числом повторений
- Циклы с неизвестным числом повторений (итерационные циклы)
- Сложные циклы

Основы программирования на языке C#

- Основные понятия языка.
- Переменные, операции и выражения.
- Операторы.
- Классы.
- Массивы и строки.
- Делегаты, события и потоки выполнения.

4. Критерии оценки

«5» Полнота использования учебного материала. Объем конспекта – 1 тетрадная страница на один раздел или один лист формата А 4. Логика изложения (наличие схем, количество смысловых связей между понятиями). Наглядность (наличие рисунков, символов, и пр.; аккуратность выполнения, читаемость конспекта. Грамотность (терминологическая и орфографическая). Отсутствие связанных предложений, только опорные сигналы – слова, словосочетания, символы. Самостоятельность при составлении.

«4» Использование учебного материала не полное. Объем конспекта – 1 тетрадная страница на один раздел или один лист формата А 4. Не достаточно логично изложено (наличие схем, количество смысловых связей между понятиями). Наглядность (наличие рисунков, символов, и пр.; аккуратность выполнения, читаемость конспекта. Грамотность (терминологическая и орфографическая). Отсутствие связанных предложений, только опорные сигналы – слова, словосочетания, символы. Самостоятельность при составлении.

«3» Использование учебного материала не полное. Объем конспекта – менее одной тетрадной страницы на один раздел или один лист формата А 4. Не

достаточно логично изложено (наличие схем, количество смысловых связей между понятиями). Наглядность (наличие рисунков, символов, и пр.; аккуратность выполнения, читаемость конспекта. Грамотность (терминологическая и орфографическая). Отсутствие связанных предложений, только опорные сигналы – слова, словосочетания, символы. Самостоятельность при составлении. Не разборчивый почерк.

«2» Использование учебного материала не полное. Объем конспекта – менее одной тетрадной страницы на один раздел или один лист формата А 4. Отсутствуют схемы, количество смысловых связей между понятиями. Отсутствует наглядность (наличие рисунков, символов, и пр.; аккуратность выполнения, читаемость конспекта. Допущены ошибки терминологические и орфографические. Отсутствие связанных предложений, только опорные сигналы – слова, словосочетания, символы. Не самостоятельность при составлении. Не разборчивый почерк.

5. Список рекомендуемой литературы, интернет – ресурсов и справочно-правовых систем

5.1. Нормативные акты

Законодательство РФ

5.2 Основная литература

1. Информационные системы и технологии в экономике и управлении : учебник для бакалавров / под ред. В. В. Трофимова. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2014.
2. Гаврилов, М.В. Информатика и информационные технологии: Учебник для бакалавров / М.В. Гаврилов, В.А. Климов; Рецензент Л.В. Кальянов, Н.М. Рыскин. - М.: Юрайт, 2013. - 378 с.

5.3 Дополнительная литература

1. Гаврилов, Л. П. Инновационные технологии в коммерции и бизнесе : учебник для бакалавров / Л. П. Гаврилов. — М. : Издательство Юрайт, 2013. — 372 с. — Серия : Бакалавр.
2. Информационные системы и технологии в экономике и управлении : учебник для бакалавров / под ред. В. В. Трофимова. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2013.
3. Абросимова, М.А. Информационные технологии в государственном и муниципальном управлении: Учебное пособие / М.А. Абросимова. - М.: КноРус, 2013. - 248 с

5.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

<http://www.intuit.ru/> - национальный открытый университет

6. Приложение:

— Титульный лист

В произвольной форме

Ссылка на образцы титульных листов различных видов текущего контроля:
<http://www.old.fa.ru/fil/kaluga/student/Pages/default.aspx>

— Пример выполнения варианта лекционной работы

Рассмотрим интерфейсы за счет представления полного определения одного из интерфейсов, определенного Microsoft - System.IDisposable. Интерфейс IDisposable содержит один метод Dispose (), предназначенный для деинициализации классами, которые выполняют очистку кода:

```
public interface IDisposable
```

```

{
void Dispose ();
}

```

Этот фрагмент показывает, что объявление интерфейса синтаксически очень похоже на объявление абстрактного класса. Однако вы должны помнить, что не допускается указывать никакой реализации ни для одного из членов интерфейса.

Интерфейс может содержать только объявления методов, свойств, индексов и событий. Он содержит только сигнатуры членов. Интерфейс не имеет никаких конструкторов, равно как и никаких полей.

Определению интерфейса также не разрешено содержать перегрузки операций.

Также не допускается указывать модификаторы при определении членов интерфейса. Члены интерфейса всегда неявно являются **public** и не могут быть **virtual** или **static**.

Вернемся к определению интерфейса `IDisposable`. Если класс объявляет, что он реализует метод `Dispose ()`, то он должен будет реализовать интерфейс `IDisposable`, что в терминах C# означает, что он наследуется от `IDisposable`.

//Этот класс ДОЛЖЕН содержать реализацию метода `IDisposable.Dispose()`, иначе-ERROR!

```

class my: IDisposable
{
public void Dispose() { // реализация метода Dispose() }
// остальная часть класса ...
}

```

В этом примере если класс `my` будет наследовать `IDisposable`, но не будет содержать реализации `Dispose ()`, в точности совпадающей с сигнатурой, определенной в интерфейсе `IDisposable`, будет выдана ошибка компиляции, поскольку в этом случае класс нарушит контракт реализации интерфейса.

`IDisposable` - сравнительно простой интерфейс, потому что определяет только один метод. Большинство интерфейсов содержат большее число методов.

Другим хорошим примером интерфейса является тот, который представляет цикл `foreach` в C#. В принципе цикл `foreach` внутри работает так, что опрашивает объект на предмет того, реализует ли он интерфейс `System. Collection. IEnumerable`. Он работает так, то компилятор C# вставляет в IL-код, использующий методы этого интерфейса для прохода по всем членам коллекции. Если же нет, `foreach` генерирует исключение.