

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

**«УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра геоинформационных систем

УТВЕРЖДАЮ

Проректор по учебной работе



Н.Г. Зарипов

«31» 08 2015 г.

РАБОЧАЯ ПРОГРАММА

УЧЕБНОЙ ДИСЦИПЛИНЫ

**«ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ИНФОРМАЦИОННЫХ
СИСТЕМ»**

Уровень подготовки: высшее образование – подготовка бакалавров

Направление подготовки бакалавров

09.03.02 Информационные системы и технологии

(код и наименование направления подготовки)

Направленность подготовки(профиль) Геоинформационные системы

Квалификация (степень) выпускника

Бакалавр.

Форма обучения

очная

Уфа 2015

Содержание

1. Место дисциплины в структуре образовательной программы	3
2. Перечень результатов обучения	3
3. Содержание и структура дисциплины (модуля).....	4
4. Учебно-методическое обеспечение самостоятельной работы студентов	6
5. Фонд оценочных средств	6
6. Учебно-методическое и информационное обеспечение дисциплины (модуля) .	12
7. Образовательные технологии	21
8. Методические указания по освоению дисциплины	21
9. Материально-техническое обеспечение дисциплины.....	23
10. Адаптация рабочей программы для лиц с ОВЗ	23

1. Место дисциплины в структуре образовательной программы

Дисциплина «Инструментальные средства информационных систем» является обязательной дисциплиной вариативной части.

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности) 09.03.02 Информационные системы и технологии, утвержденного приказом Министерства образования и науки Российской Федерации от "12" марта 2015 г. № 219.

Целью освоения дисциплины является формирование у будущих бакалавров в области информационных систем и технологий базовых знаний и навыков администрирования информационных систем.

Задачи:

- Изучить основные классы инструментальных средств, используемых при разработке и эксплуатации информационных систем.
- Понять принципы построения инструментальных средств и их взаимодействия с операционной системой и приложениями.
- Овладеть отдельными образцами программных инструментальных средств.
- Сформировать комплекс знаний и умений, направленных на эффективное использование инструментальных средств при решении технологических задач в процессе разработки и эксплуатации информационных систем.

Входные компетенции:

№	Компетенция	Код	Уровень освоения, определяемый этапом формирования компетенции	Название дисциплины (модуля), сформировавшего данную компетенцию
1.	Способность к проектированию базовых и прикладных информационных технологий	ПК-11	Базовый, 1 этап	Инфокоммуникационные системы и сети

Исходящие компетенции:

№	Компетенция	Код	Уровень освоения, определяемый этапом формирования компетенции	Название дисциплины (модуля), для которой данная компетенция является входной
1.	Способность участвовать в работах по доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных систем	ПК-15	Пороговая, 4 этап	Администрирование информационных систем

2. Перечень результатов обучения

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций.

Планируемые результаты обучения по дисциплине:

№	Формируемые компетенции	Код	Знать	Уметь	Владеть
1	Способностью	ПК-13	Основные	Разрабатывать	Методами

	разрабатывать средства автоматизированного проектирования информационных технологий		принципы и алгоритмы автоматизированного проектирования информационных объектов	методы и алгоритмы преобразования сложных структур данных	разработки, отладки и использования программного обеспечения для обработки сложных структур данных
2	Способность участвовать в работах по доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных систем	ПК-15	Архитектуру и принципы работы основных типов информационных систем.	Определять и модифицировать конфигурацию информационных систем	Методами отладки и тестирования программного обеспечения. Технологиями сохранения и восстановления конфигурации информационных систем.

3. Содержание и структура дисциплины (модуля)

Общая трудоемкость дисциплины составляет 3 зачетных единиц (108 часов).

Трудоемкость дисциплины по видам работ

Вид работы	Трудоемкость, час.
	6 семестр
Лекции (Л)	14
Практические занятия (ПЗ)	-
Лабораторные работы (ЛР)	20
КСР	3
Курсовая проект работа (КР)	-
Расчетно - графическая работа (РГР)	-
Самостоятельная работа (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам, рубежному контролю и т.д.)	62
Подготовка и сдача экзамена	-
Подготовка и сдача зачета	9
Вид итогового контроля (зачет, экзамен)	зачет

Содержание разделов и формы текущего контроля

№	Наименование и содержание раздела	Количество часов				Литература, рекомендуемая студентам	Виды интерактивных образовательных технологий	
		Аудиторная Работа			СРС			Всего
		Л	ЛР	КСР				
1	Введение Цели и задачи дисциплины, общие сведения об инструментальных средствах на примере широко известных решений.	2			6	8	Р. 6.1 - №1,2, Р. 6.2 - №1	Классическая лекция
2	Кодирование алгоритмов Оценка вычислительной сложности алгоритмов. Компиляторы и интерпретаторы. Линковщики, оптимизаторы и препроцессоры.	2	4	1	10	16	Р. 6.1 - №1,2 Р. 6.2 –№ 2	Классическая лекция
3	Кодирование пользовательского интерфейса Визуальное программирование. Интегральные среды разработки. Средства прототипирования пользовательского интерфейса.	2	4	1	10	16	Р. 6.1 -№ 1,2, Р. 6.2 -№ 1, 2	Классическая лекция
4	Отладка и тестирование Ошибки и их классификация по месту в цикле разработки. Отладчики. Средства автоматизированного тестирования.	2	4	1	10	16	Р. 6.1 - №1,2 Р. 6.2 –№ 2	Классическая лекция
5	Средства виртуализации Виртуальные машины и среды. Виртуальные устройства и ресурсы.	2	4		10	16	Р. 6.1 - №1,2 Р. 6.2 –№ 2	Классическая лекция
6	Защита данных Средства архивирования и резервного копирования данных. Средства шифрования и удостоверения данных.	2	4		10	16	Р. 6.1 -№ 1,2, Р. 6.2 -№ 1, 2	Классическая лекция
7	Заключение	2			6	8	Р. 6.1 - №1,2	Классическая лекция

Лабораторные работы

№ ЛР	№ раздела	Наименование лабораторных работ	Кол-во часов
1	2	Использование средств автозапуска приложений	4
2	3	Создание пользовательских интерфейсов в среде VisualStudio	4
3	4	Отладка и тестирование	4
4	5	Выполнение приложений в виртуальной среде	4
5	6	Работа с файловым менеджером	4

4. Учебно-методическое обеспечение самостоятельной работы студентов

Тема 1. Введение

Вопросы для самостоятельного изучения (подготовке к обсуждению):

1. Информационные системы с открытым кодом.
2. Программное обеспечение для создания электронных учебников.

Тема 2. Кодирование алгоритмов

Вопросы для самостоятельного изучения (подготовке к обсуждению):

1. Редакторы программного кода.
2. Способы записи алгоритма.

Тема 3. Кодирование пользовательского интерфейса

Вопросы для самостоятельного изучения (подготовке к обсуждению):

1. Типы и режимы.
2. Средства визуализации.

Тема 4. Отладка и тестирование

Вопросы для самостоятельного изучения (подготовке к обсуждению):

1. Классификация ошибок.
2. План тестирования.

Тема 5. Средства виртуализации

Вопросы для самостоятельного изучения (подготовке к обсуждению):

1. Визуализация как этап анализа данных.
2. Классификация визуализации.

Тема 6. Защита данных

Вопросы для самостоятельного изучения (подготовке к обсуждению):

1. Виды шифрования.
2. Порядок доступа к персональным данным.

Тема 7. Заключение

Вопросы для самостоятельного изучения (подготовке к обсуждению):

1. Вспомогательные программы, реализующие часто используемые системные действия.
2. Система для конструирования нечетких алгоритмов.

Для концентрации внимания студентов именно на этих темах им предлагаются домашние задания в виде самостоятельного поиска информации в открытых источниках (например, в сети Интернет).

5. Фонд оценочных средств

Оценка уровня освоения дисциплины осуществляется в виде текущего и промежуточного контроля успеваемости бакалавров, и на основе критериев оценки уровня освоения дисциплины.

Контроль представляет собой набор заданий и проводится в форме контрольных мероприятий по оцениванию фактических результатов обучения студентов и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

- учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине и пр.);
- степень усвоения теоретических знаний;
- уровень овладения практическими умениями и навыками по всем видам учебной работы;
- результаты самостоятельной работы.

Активность обучающегося на занятиях оценивается на основе выполненных работ и заданий, предусмотренных ФОС дисциплины.

Оценивание проводится преподавателем независимо от наличия или отсутствия обучающегося (по уважительной или неуважительной причине) на занятии. Оценка носит комплексный характер и учитывает достижения обучающегося по основным компонентам учебного процесса за текущий период.

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Уровень освоения, определяемый этапом формирования компетенции	Наименование оценочного средства
1	Введение. Цели и задачи дисциплины.	ПК-15	Пороговый, 1 этап	Контрольные вопросы
2	Кодирование алгоритмов	ПК-15	Пороговый, 1 этап	Контрольные вопросы; контроль выполнения лабораторных работ
3	Кодирование пользовательского интерфейса	ПК-15	Пороговый, 1 этап	Контрольные вопросы; контроль выполнения лабораторных работ
4	Отладка и тестирование	ПК-15	Пороговый, 1 этап	Контрольные вопросы; контроль выполнения лабораторных работ

5	Средства виртуализации	ПК-15	Пороговый, 1 этап	Контрольные вопросы; контроль выполнения лабораторных работ
6	Защита данных	ПК-15	Пороговый, 1 этап	Контрольные вопросы; контроль выполнения лабораторных работ
7	Заключение	ПК-15	Пороговый, 1 этап	Контрольные вопросы

Вопросы для контроля

1. Интерпретаторы и компиляторы. Классификация.
2. Основные принципы устройства компилятора.
3. Пакеты прикладных программ.
4. Математические пакеты общего и специального назначения. Основные характеристики и назначение.
5. Программные среды, классификация, характеристики.
6. Операционные системы семейства *Windows*. Архитектура, основные возможности, настройка и администрирование.
7. Инструментальные средства разработки ИС: *VS, NetBeans, Eclipse, Delphi*.
8. Основные функции текстового, табличного процессора. Характеристики интегрированных пакетов.
9. Утилиты управления жестким диском.
10. Утилиты для работы с файлами.
11. Средства исполнения программного кода: интерпретаторы и компиляторы.
12. Типы программных модулей: исполняемые, компоненты ActiveX, библиотеки.
13. Средства и технологии отладки программного кода: трассировка, точки останова и пошаговое выполнение.
14. Псевдокод и средства его исполнения.
15. Сообщения об ошибках: ошибки трансляции и ошибки исполнения.
16. Средства виртуализации: виртуальные ресурсы, среды, машины. Средства виртуализации в *Windows*.
17. Сообщения об ошибках: исключения и логические ошибки.
18. Типы пользовательских интерфейсов и средства их разработки.
19. Сборка модулей: статические, динамические и отладочные библиотеки.

20. Интерфейс командной строки и его применение для управления ОС. Средства работы с интерфейсом командной строки в *Windows*.
21. Архивирование и резервное копирование данных. Утилиты и встроенные средства *Windows* для архивирования и резервного копирования.
22. Шифрование данных: основные принципы и средства.
23. Несимметричные алгоритмы шифрования и электронные подписи.
24. Кодирование и представление информации: целочисленные значения, числа с плавающей точкой.
25. Кодирование и представление информации: текстовые кодировки и способы представления строчных значений.

Критерии оценки контрольных вопросов:

- оценка «зачтено» выставляется студенту, при правильном ответе на вопрос, соответствующей темы изучения. Правильность ответа определяется корректностью формулировок и достаточным объемом освещения темы.
- оценка «не зачтено» выставляется студенту, при некорректном ответе на вопрос, или упущении существенных моментов рассматриваемой темы.

5.2 Типовые оценочные материалы

Оценочные материалы для лабораторных работ

Раздел (тема) дисциплины: **Кодирование алгоритмов.**

Задание выполняется на лабораторной работе №1. **Использование средств автозапуска приложений.**

Пороговый уровень

1. Настроить автозапуск через папку автозапуска
2. Настроить автозапуск через планировщик

Примерные теоретические вопросы для защиты лабораторной работы:

1. Как отключить автозапуск приложений?
2. Как создать задачу в windows?

Раздел (тема) дисциплины: **Кодирование пользовательского интерфейса**

Задание выполняется на лабораторной работе №2. **Создание пользовательских интерфейсов в среде *VisualStudio*.**

Пороговый уровень

1. Создать приложение
2. Создать графический интерфейс программы
3. Ознакомиться с автоматически сформированным кодом, генерирующим графический интерфейс программы

Примерные теоретические вопросы для защиты лабораторной работы:

1. Что такое окно свойств *Properties*?
2. Элементы управления режима дизайна и кода.

Раздел (тема) дисциплины: **Отладка и тестирование**

Задание выполняется на лабораторной работе №3. **Отладка и тестирование.***Пороговый уровень*

1. Создать проект
2. Изучить средства отладки

Примерные теоретические вопросы для защиты лабораторной работы:

1. Что такое и для чего используются точки останова? Что такое условные точки останова?
2. Перечислите и дайте краткое описание основным окнам среды *Microsoft VisualStudio*, используемым при отладке?

Раздел (тема) дисциплины: **Средства визуализации**

Задание выполняется на лабораторной работе №4. **Выполнение приложений в виртуальной среде.**

Пороговый уровень:

1. Создать Виртуальную машину
2. Установить на созданную виртуальную машину операционную систему
3. Перенести на виртуальную машину приложение и запустить его
4. Убедиться, что приложение на виртуальной машине не имеет доступа к ресурсам реального компьютера

Примерные теоретические вопросы для защиты лабораторной работы:

1. Опишите этапы создания ВМ.
2. Опишите этапы подключения жесткого диска.

Раздел (тема) дисциплины: **Защита данных**

Задание выполняется на лабораторной работе №5. **Работа с файловым менеджером.**

Пороговый уровень

1. Ознакомиться с программой *TotalCommander*
2. Скопировать, переместить и удалить файлы в *TotalCommander*
3. Создать группу файлов и произвести групповое переименование

Примерные теоретические вопросы для защиты лабораторной работы:

1. Менеджер файлов *Total Commander*: общая характеристика, функциональные возможности.
2. Создание каталогов в *Total Commander*. Объединение файлов и каталогов в группу в *Total Commander*. Переименование и удаление файлов и каталогов.

Критерии оценки лабораторных работ:

- 5 баллов выставляется студенту, если он полностью ответил на контрольные вопросы, правильно все практические задачи лабораторной работы и подготовил отчет, полностью удовлетворяющий требованиям;

- 4 балла выставляется студенту, если он частично ответил на контрольные вопросы и правильно все практические задачи лабораторной работы, а также подготовил отчет, полностью удовлетворяющий требованиям;

- 3 балла выставляется студенту, если он не ответил на меньшую часть контрольных вопросов и решил одну из задач лабораторной работы с ошибкой, при условии получения конечного результата выполнения лабораторной работы, а также подготовил отчет, полностью удовлетворяющий требованиям;

- 2 балла выставляется студенту, если он ответил на все контрольные вопросы, частично выполнил лабораторную работу, но не получил конечного результата и оформил отчет, удовлетворяющий требованиям;

- 1 балл выставляется студенту, если он не ответил на большую часть контрольных вопросов, частично выполнил лабораторную работу, но не получил конечного результата и оформил отчет, удовлетворяющий требованиям;

- 0 баллов выставляется студенту, не приступившему к выполнению лабораторной работы.

Требования предъявляемые к отчету:

1. Четкое формулирование поставленной цели лабораторной работы и задач, решение которых необходимо для достижения поставленной цели.

2. Описание в виде пунктов, тех действий, которые требуются для решения поставленных задач. Все рисунки и таблицы последовательно нумеруются и описываются.

5.2 Методические материалы, определяющие процедуры оценивания результатов обучения (знаний, умений, владений), характеризующих этапы формирования компетенций

Приводится методика проведения процедур оценивания конкретных результатов обучения (знаний, умений, владений) формируемого этапа компетенции. То есть для каждого образовательного результата определяются показатели и критерии

сформированности компетенций на различных этапах их формирования, приводятся шкалы и процедуры оценивания.

Компетенция, ее этап и уровень формирования	Заявленный образовательный результат	Типовое задание из ФОС, позволяющее проверить сформированность образовательного результата	Процедура оценивания образовательного результата	Критерии и оценки
ПК-15, пороговый уровень, 1 этап	Знать: Архитектуру и принципы работы основных типов информационных систем.	Ответ на вопросы для контроля № 1-15, 20-26 (см. выше).	В конце изучения каждой темы на практическом занятии студенту дается 1 вопрос на выбор. Либо бакалавр отвечает на теоретические вопросы при сдаче зачета.	Критерий оценки (см. выше)
	Уметь: Определять и модифицировать конфигурацию информационных систем	Защита лабораторной работы № 2-3	Лабораторная работа проводится в соответствии с расписанием проведения занятий. В рамках данной лабораторной работы бакалавры в малых группах организуют научный семинар. Результатом защиты работы является организованный семинар на лабораторной работе. Методические указания по выполнению лабораторных работ (см. выше)	Критерий оценки (см. выше)
	Владеть: Методами отладки и тестирования программного обеспечения. Технологиями сохранения и восстановления	Защита лабораторных работ № 3-5	Лабораторные работы проводятся в соответствии с расписанием проведения занятий. Отчет по лабораторной работе студенты защищают	Критерий оценки (см. выше)

	конфигурации информационных систем.		в конце/начале практического занятия или на специально выделенных консультациях, время защиты – 5 минут Методические указания по выполнению лабораторных работ (см. выше)	
--	-------------------------------------	--	--	--

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

6.1 Основная литература

1. Павловская, Т. А. С#. Программирование на языке высокого уровня : [учебник для вузов] / Т.А.Павловская. - Санкт-Петербург: Питер, 2012.- 432 с.
2. Васильков, Ю. В. Компьютерные технологии вычислений в математическом моделировании : учебное пособие для вузов / Ю. В. Васильков, Н. Н. Василькова .— М. : Финансы и статистика, 2004 .— 256 с.

6.2 Дополнительная литература

1. Платт, Д. С. Платформа Microsoft 2005 = The Microsoft Platform AHEAD : пер. с англ. / Д. С. Платт .— М. : Русская редакция, 2004 .— 245 с.
2. Гагарина, Л. Г. Технология разработки программного обеспечения : [учебное пособие] / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул ; под ред. Л. Г. Гагариной .— М. : ФОРУМ : ИНФРА-М, 2008 .— 399 с.

6.3 Программное обеспечение современных информационно-коммуникационных технологий

- Visual Studio
- Total Commander
- Virtual Box

6.4 Методические указания к лабораторным занятиям

Занятие 1. Использование средств автозапуска приложений.

1. Цель и задачи работы

Цель работы: Изучение инструментов автозапуска программ в ОС *WindowsXP*.

Задачи работы:

- настроить автозапуск через папку автозапуска;
- настроить автозапуск через планировщик.

2. Краткие теоретические сведения

Автозагрузка программ в *Windows* — это инструмент для запуска конкретного программного обеспечения одновременно с включением компьютера.

3. Краткое описание используемого программного обеспечения

Для выполнения лабораторной работы используется персональный компьютер под управлением операционной системы *MSWindowsXP*.

4. Меры безопасности

При выполнении лабораторной работы не отключать и не подключать компоненты оборудования и вычислительной техники друг к другу и к сети питания без разрешения преподавателя, или лаборанта; не помещать посторонние предметы в корпус компьютера и периферийных устройств; не повреждать кабели и другие элементы оборудования.

5. Задание

1) Установить приложение «Блокнот» на автозапуск.

6. Методика выполнения работы

1. Найти папку «Автозагрузка».

C:\Users\ИмяПользователя\AppData\Roaming\Microsoft\Windows\StartMenu\Programs\Start up

1. Скопировать в нее ярлык программы
2. Перезапустить систему и проверить запуск приложения.
3. Запустить планировщик *Windows*.
4. Добавить событие в систему и настроить в нем запуск приложения.
5. Перезапустить систему и проверить запуск приложения.

7. Контрольные вопросы

1. Какие еще есть методы добавления приложения в автозагрузку?
2. Как отключить автозагрузку приложения?

8. Требования к содержанию и оформлению отчета

Отчет должен содержать:

- задание лабораторной работы;
- описание выполнения работы с экранными формами и пояснениями того, решение каких задач на них показано;
- выводы по работе.

9. Критерии результативности лабораторного практикума

Критерием результативности лабораторной работы является полное выполнение студентом задания, формирование у студента базовых умений и навыков настройки автозапуска приложений.

Занятие 2. Создание пользовательских интерфейсов в среде *VisualStudio*.

1. Цель и задачи работы:

Цель работы: Изучение средств визуальной разработки графического пользовательского интерфейса

Задачи работы:

- создать приложение
- создать графический интерфейс программы
- ознакомиться с автоматически сформированным кодом, генерирующим графический интерфейс программы

2. Краткие теоретические сведения

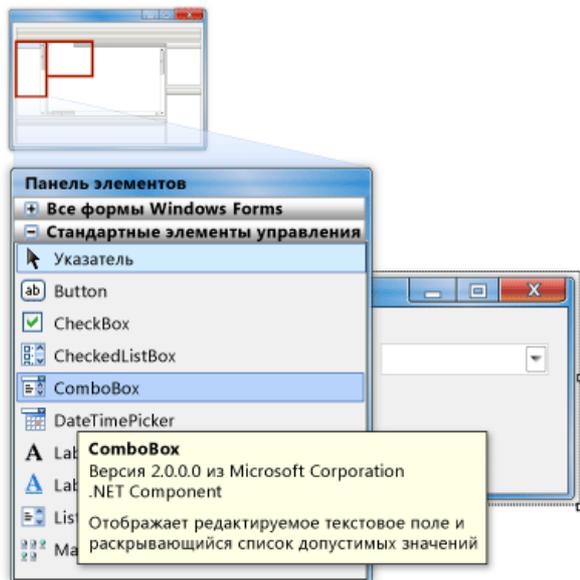
В создании пользовательских интерфейсов для приложений *Windows Forms* имеются три основных этапа:

- Добавление элементов управления на поверхность разработки.
- Установка начальных свойств для элементов управления.
- Написание обработчиков для заданных событий.

Хотя пользовательский интерфейс можно создать, написав собственный код, с помощью конструкторов это можно сделать намного быстрее.

Добавление элементов управления

Элементы управления, такие как кнопки и текстовые поля, можно перетащить мышью на поверхность разработки, представляющую форму. На рисунке ниже показано поле со списком, которое при помощи перетаскивания из панели элементов было добавлено в форму в конструкторе *Windows Forms*.



При работе в визуальном режиме конструктор преобразует выполняемые действия в исходный код *C#* и записывает их в файл проекта с именем `имя.designer.cs`, где имя — имя, назначенное форме. Когда приложение будет выполнено, исходный код разместит элементы пользовательского интерфейса и скорректирует их размер таким образом, как они отображаются на поверхности разработки. Дополнительные сведения см. в разделе **Конструктор Windows Forms**.

Задание свойств

После добавления элемента управления в форму в окне **Свойства** можно задать его свойства, такие как цвет фона и текст по умолчанию. Значения, задаваемые в окне **Свойства**, являются начальными значениями, которые будут назначены соответствующему свойству при создании элемента управления во время выполнения. Во многих случаях доступ к значениям и их изменение возможен программными средствами во время выполнения путем получения или установки свойств в экземпляре класса элемента управления в приложении. Окно **Свойства** может оказаться полезным во время выполнения, так как с его помощью можно просматривать все свойства, события и методы, поддерживаемые элементом управления. Дополнительные сведения см. в разделе **Окно "Свойства"**.

Обработка событий

Программы с графическим интерфейсом пользователя главным образом основаны на событиях. Такие программы ожидают действий пользователя, например ввода текста в текстовое поле, нажатия кнопки или изменения выбора в поле со списком. При выполнении действия элемент управления, который всего лишь представляет собой экземпляр класса *.NET Framework*, отправляет событие в приложение. Для обработки события можно написать специальный метод в приложении, который будет вызван при получении события.

В окне **Свойства** можно указать события, которые должны обрабатываться в коде. Для просмотра событий элемента управления следует выбрать его в конструкторе и нажать кнопку **События** с изображением молнии в окне **Свойства**. Следующая схема отображает кнопку событий. ⚡

При добавлении обработчика событий с использованием окна **Свойства** конструктор автоматически напишет основную часть пустого метода, а пользователь должен написать код, на основе которого метод будет выполнять необходимые действия. Большинство элементов управления создают много событий, но приложению часто придется обрабатывать лишь часть из них или даже только одно. Например, возможно потребуется обработать событие `Click` для кнопки, но не ее

событие Paint, если только нет необходимости изменить ее внешний вид каким-либо дополнительным способом. Для каждого элемента управления существует обработчик событий, определенный по умолчанию. Чтобы создать обработчик событий по умолчанию, дважды щелкните элемент управления в форме. Будет создан обработчик событий и откроется редактор кода для написания события для обработки события.

3. Краткое описание используемого программного обеспечения

Для выполнения лабораторной работы используется персональный компьютер под управлением операционной системы *MSWindowsXP* с установленной средой разработки *VisualStudio*.

4. Меры безопасности

При выполнении лабораторной работы не отключать и не подключать компоненты оборудования и вычислительной техники друг к другу и к сети питания без разрешения преподавателя, или лаборанта; не помещать посторонние предметы в корпус компьютера и периферийных устройств; не повреждать кабели и другие элементы оборудования.

5. Задание

1. Создать графический интерфейс приложения *Windows*.

6. Методика выполнения работы

1. Создать в *VisualStudio* проект «Приложение *Windows Forms*».
2. Создать графический интерфейс, используя стандартные элементы управления.

3. Реализовать алгоритм работы приложения, взаимодействующий с графическим интерфейсом.

4. Отладить разработанную программу.

5. Ознакомиться с автоматически сформированным кодом, генерирующим графический интерфейс программы.

7. Контрольные вопросы.

1. Что такое окно свойств *Properties*?
2. Элементы управления режима дизайна и кода.

8. Требования к содержанию и оформлению отчета

Отчет должен содержать:

- задание лабораторной работы;
- описание выполнения работы с экранными формами и пояснениями того, решение каких задач на них показано;
- выводы по работе.

Критерии результативности лабораторного практикума

Критерием результативности лабораторной работы является полное выполнение студентом задания, формирование у студента базовых умений и навыков по освоению навыков визуальной разработки графического пользовательского интерфейса

Занятие 3. Отладка и тестирование.

1. Цель и задачи работы

Цель работы: Изучение отладку и тестирование приложений в среде разработки *MicrosoftVisualStudio*.

Задачи работы:

- изучить режим пошагового выполнения
- изучить механизм контрольных точек
- изучить средства просмотра переменных

2. Краткие теоретические сведения

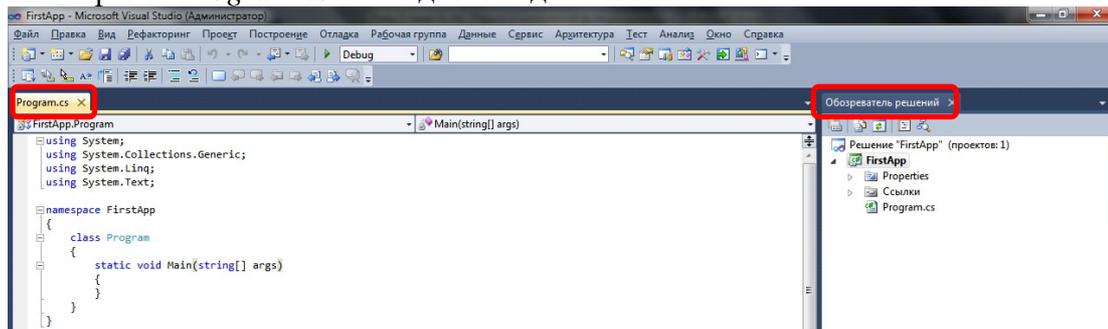
Создание нового проекта – консольного приложения.

Выберите в меню *Файл \ создать \ проект*. Откроется меню:

Выбор целевой версии *.NETFramework*

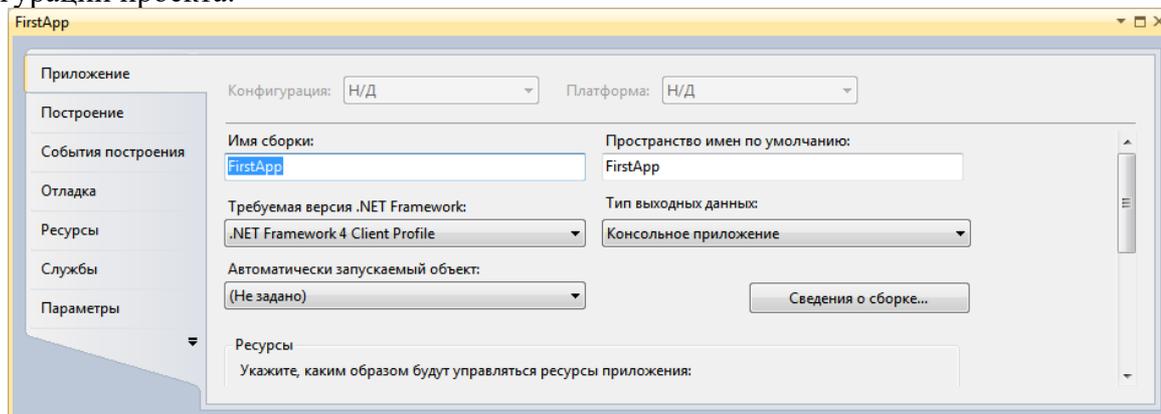
Слева отображаются шаблоны проектов, выберите ветвь *VisualC# \ Windows* (Если вместо ветви *VisualC#* отображается *VisualC++* – откройте ветвь «Другие языки»). По центру отображается содержимое выбранного набора шаблонов. Выберите *Консольное приложение*. Кроме всего прочего, можно выбрать версию *.NET*, которая будет использоваться при создании приложений. В завершении укажите имя проекта *FirstApp* и нажмите кнопку *ОК*.

По-умолчанию, открывается 2 вкладки: *Обозреватель решений (Solutionexplorer)* и содержимое файла *Program.cs* – исходный код:



Если обозреватель решений отсутствует, откройте его через меню *Вид \ Обозреватель решений*. Эта утилита позволяет просматривать набор всех файлов с содержимым и ссылаемых сборок, которые входят в состав текущего проекта.

Обозреватель решений имеет вид дерева папок и файлов. Имя корневой папки – имя проекта, например, *FirstApp*. Внутри отображаются свойства проекта (*Properties*), если дважды щелкнуть левой кнопкой мыши по этой папке, откроется редактор конфигурации проекта.

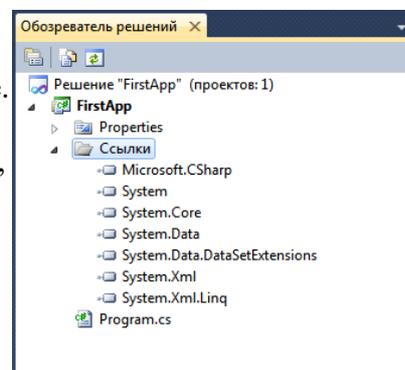


В этом окне можно устанавливать различные параметры безопасности, назначать сборке имя, разворачивать приложение, выставлять необходимые для приложения ресурсы и конфигурировать события, которые должны происходить перед и после компиляции сборки.

Если раскрыть папку свойств (нажав на треугольник слева от имени папки), получим доступ к файлу – «*AssemblyInfo.cs*», в котором содержится информация о сборке.

Кроме того, в проекте присутствует папка «Ссылки» (*References*), внутри которой отображается список всех сборок, на которые в проекте были добавлены ссылки. В зависимости от типа проекта и целевой версии *.NetFramework*, этот список выглядит по-разному.

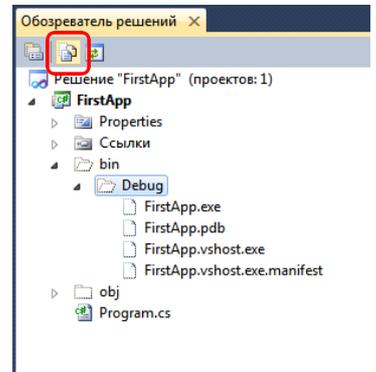
Ну и последний элемент – файл с исходным кодом приложения «*Program.cs*». Если по какой-либо причине вкладка (или окно) с исходным кодом не отображается, просто дважды щелкните мышкой по этому файлу.



Построение решения.

Есть несколько вариантов построения:

1. Построение решения. Для этого, на панели инструментов VisualStudio выберите конфигурацию построения – «*Debug*» (Отладка), затем в меню «*Построение*» выберите «*Построить решение*». В случае удачного построения (компилятор не обнаружит ошибок в исходном коде), будет создан исполняемый файл (*.exe) с именем вашего проекта (*FirstApp*) и файлы, куда записывается информация для отладки приложения. Они создаются в папке, где сохранен сам проект, в директории *bin\debug*. Их можно увидеть и в обозревателе решений, если нажать на кнопку «*Показать все файлы*».
2. Запуск отладки приложения. Аналогично предыдущему варианту, но после построения приложения, оно запускается. То есть можно сразу посмотреть результаты своего творчества. Для запуска отладки нужно выбрать конфигурацию построения – «*Debug*» и нажать кнопку «*Начать отладку*» или клавишу *F5*.



Начать отладку

Выбор конфигурации
построения



3. Построение финальной версии приложения. Используется, когда приложение отлажено (то есть ошибки отсутствуют).

Для этого сначала в меню «*Построение*» выберите «*Очистить решение*» (*Cleansolution*) – среда удалит промежуточные файлы и выходные файлы, созданные во время предыдущих построений. Затем выберите конфигурацию построения – «*Release*» (Выпуск), после чего либо постройте решение (если не хотите запускать), либо запустите отладку, либо запустите без отладки (комбинация клавиш *ctrl + F5*). В этом случае, решение будет создано также в папке *bin*, но в подкаталоге *Release* (а не *Debug*, как в предыдущих двух случаях).

Ошибки в коде

Для каждой ошибки отображается описание, имя проекта, имя файла, номер строки и столбца, где встретилась эта ошибка в указанном файле. Если дважды щелкнуть левой кнопкой мыши по ошибке, то курсор в файле с исходным кодом будет помещен в указанную строку и столбец.

3. Краткое описание используемого программного обеспечения

Для выполнения лабораторной работы используется персональный компьютер под управлением операционной системы *MSWindowsXP* установленной средой разработки *VisualStudio*.

4. Меры безопасности

При выполнении лабораторной работы не отключать и не подключать компоненты оборудования и вычислительной техники друг к другу и к сети питания без разрешения преподавателя, или лаборанта; не помещать посторонние предметы в корпус компьютера и периферийных устройств; не повреждать кабели и другие элементы оборудования.

5. Задание

1. Создать новый проект.
2. Изучить средства отладки.

6. Методика выполнения работы

1. Запустить среду разработки *VisualStudio*.
2. Создать новый проект – консольное приложение.
3. Запрограммировать алгоритм по своему выбору.
4. Выставить контрольные точки.
5. Запустить приложение.
6. Изучить режимы пошагового выполнения программы после его остановки на контрольной точке.

7. Изучить средства просмотра переменных.

7. Контрольные вопросы

1. Назовите основные составляющие среды *VisualStudio* и их назначение.
2. Каково назначение Панели элементов?
3. Назовите состав и назначение пунктов главного меню системы.

8. Требования к содержанию и оформлению отчета

Отчет должен содержать:

- задание лабораторной работы;
- описание выполнения работы с экранными формами и пояснениями того, решение каких задач на них показано;
- выводы по работе.

9. Критерии результативности лабораторного практикума

Критерием результативности лабораторной работы является полное выполнение студентом задания, формирование у студента базовых умений и навыков создания проекта, его отладка и тестирование.

Занятие 4. Выполнение приложений в виртуальной среде.

1. Цель и задачи работы:

Цель работы: Ознакомление с технологией выполнения приложений в виртуальной среде

Задачи работы:

- создать виртуальную машину (ВМ)
- установить на созданную ВМ операционную систему
- перенести на ВМ приложение и запустить его
- убедиться, что приложение на ВМ не имеет доступа к ресурсам реального компьютера

2. Краткие теоретические сведения

Технология виртуализации предназначена для осуществления возможности одновременного запуска на одном компьютере нескольких (в том числе, различных) ОС. **VirtualBox:**

- является кросс платформенным приложением виртуализации и работает на компьютерах с процессорами *Intel* или *AMD* под управлением операционных систем *Windows, Mac, Linux* или *Solaris*;
- расширяет возможности вашего компьютера тем, что позволяет работать множеству операционных систем одновременно (внутри виртуальных машин);
- позволяет установить и запустить виртуальных машин столько, сколько вам нужно, ограничивается только размером свободного дискового пространства и памяти.

В ОС, реализованной на виртуальной машине, имеется возможность выполнять различные малоизученные или потенциально опасные для нее операции, не беспокоясь о последствиях: поскольку система является виртуальной, ее крах или частичное повреждение не скажется на работе реальной ОС.

Основные преимущества такого подхода состоят в следующем:

- появляется возможность инсталляции на одном компьютере нескольких ОС без необходимости соответствующего конфигурирования физических жестких дисков;
- возможно осуществлять работу с несколькими ОС одновременно с динамическим переключением между ними без перезагрузки реальной системы;

- сокращается время изменения состава и конфигурации установленных виртуальных ОС;
- осуществляется изоляция реального оборудования от нежелательного воздействия программного обеспечения, работающего в среде виртуальной ОС;
- появляется возможность моделирования вычислительной сети на единственном автономном компьютере.

Задачи, которые можно решать без перезагрузки системы и без опасения нанести ей какой-либо ущерб:

- освоение новой, альтернативной ОС;
- запуск специализированных приложений, предназначенных для работы в среде конкретной ОС;
- тестирование одного приложения под управлением различных ОС;
- установка и удаление оценочных или демонстрационных версий новых приложений;
- тестирование потенциально опасного программного обеспечения, относительно которого имеется подозрение на вирусное заражение;
- управление правами доступа пользователей к данным и программам в пределах виртуальной среды.

3. Краткое описание используемого программного обеспечения

Для выполнения лабораторной работы используется персональный компьютер под управлением операционной системы *MSWindowsXP* с установленной на нее *VirtualBox*.

4. Меры безопасности

При выполнении лабораторной работы не отключать и не подключать компоненты оборудования и вычислительной техники друг к другу и к сети питания без разрешения преподавателя, или лаборанта; не помещать посторонние предметы в корпус компьютера и периферийных устройств; не повреждать кабели и другие элементы оборудования.

5. Задание

1. Создать виртуальную машину и установить операционную систему.
2. Настроить «Общие папки», перенести и установить программу на ВМ.
3. Убедиться, что приложение на ВМ не имеет доступа к ресурсам реального компьютера (используя стандартный диалог приложения для работы с файлами).

6. Методика выполнения работы

1. Запустить *VirtualBox*. Создать виртуальную машину.
2. Задать параметры виртуальной машины.
3. Запустить машину и выполнить установку *WindowsXP*.
4. Перейти в настройки виртуальной машины, далее в раздел «Общие папки».
5. Добавить новую общую папку.
6. Выбрать папку для взаимодействия ВМ и главного компьютера, скопировать в нее приложение.
7. Запустить операционную систему.
8. Запустить приложение.
9. Через меню работы с файлами проверить недоступность работы с файлами реального ПК.

7. Контрольные вопросы

1. Что такое виртуальная машина?
2. Какие операционные системы можно устанавливать в качестве гостевой в *OracleVirtualBox*?

3. Есть ли у виртуальной машины свой *BIOS*?

8. Требования к содержанию и оформлению отчета

Отчет должен содержать:

- задание лабораторной работы;

- описание выполнения работы с экранными формами и пояснениями того, решение каких задач на них показано;
- выводы по работе.

9. Критерии результативности лабораторного практикума

Критерием результативности лабораторной работы является полное выполнение студентом задания, формирование у студента базовых умений и навыков работы с технологиями выполнения приложений в виртуальной среде.

Занятие 5. Работа с файловым менеджером.

1. Цель и задачи работы

Цель работы: Получить основные навыки работы с программой *TotalCommander*: освоить различные способы выполнения операций с файлами и папками.

Задачи работы:

- скопировать файлы в различных режимах копирования (одиночный, групповой, групповой по шаблону)
- выполнить групповое переименование файлов с различными шаблонами

2. Краткие теоретические сведения

Total Commander - файловый менеджер, созданный швейцарцем Кристианом Гислером. Программа совмещает в себе как интерфейс панелей Norton Commander, так и программы Проводник. От первых *Total Commander* унаследовал две панели, командную строку под ними, меню и видимое расширение файлов. А от программы Проводник стандартные иконки папок и файлов, панель инструментов и контекстное меню. Присутствуют «горячие» кнопки, однако можно работать с файлами и папками привычным методом «тяни и брось» (*Drag and Drop*).

В возможности *Total Commander* входят:

- поддержка языковых модулей для более десятка языков (включая русский);
- поддержка дополнительных программных модулей (*plugins*), созданных сторонними разработчиками;
- встроенный *FTP* клиент (*File Transfer Protocol*- протокол работы с файлами в Internet);
- встроенная поддержка архивов (часто используемые – *zip* и *rar*).
- встроенная программа просмотра графических файлов.

3. Краткое описание используемого программного обеспечения

Для выполнения лабораторной работы используется персональный компьютер под управлением операционной системы *MSWindowsXP* и установленной программой *TotalCommander*.

4. Меры безопасности

При выполнении лабораторной работы не отключать и не подключать компоненты оборудования и вычислительной техники друг к другу и к сети питания без разрешения преподавателя, или лаборанта; не помещать посторонние предметы в корпус компьютера и периферийных устройств; не повреждать кабели и другие элементы оборудования

5. Задание

1. Скопировать, переместить и удалить файлы в *TotalCommander*.
2. Создать группу файлов и произвести групповое переименование

6. Методика выполнения работы

1. Создать текстовый файл и сохранить в каталоге три текстовых файла с именами Файл1, Файл2 и Файл3, Файл4, Файл5, Документ1, Документ2.
2. Скопировать Файл1 в каталог мое_Имя.
3. Скопировать Файл1 из каталога мое_Имя в каталог Фамилия под другим именем (Док1).
4. Переместить свой Файл2 в каталог Фамилия.
5. Удалить файл Документ1 из каталога мое_Имя.
6. Выделить файлы по шаблону имени, скопировать выделенные файлы.

7. Выделить произвольную группу файлов, скопировать их.

Групповое переименование:

1. Создать группу файлов с одинаковым расширением.
2. Выделить группу файлов (предназначенную для переименования)
3. Вызвать диалог группового переименования.
4. Задать параметры для переименования.
5. Проверить результат.

7. Контрольные вопросы

1. Менеджер файлов *Total Commander*: общая характеристика, функциональные возможности.
2. Создание каталогов в *Total Commander*. Объединение файлов и каталогов в группу в *Total Commander*. Переименование и удаление файлов и каталогов.

8. Требования к содержанию и оформлению отчета

Отчет должен содержать:

- задание лабораторной работы;
- описание выполнения работы с экранными формами и пояснениями того, решение каких задач на них показано;
- выводы по работе.

9. Критерии результативности лабораторного практикума

Критерием результативности лабораторной работы является полное выполнение студентом задания, формирование у студента базовых умений и навыков работы с программой *TotalCommander*.

7. Образовательные технологии

При реализации дисциплины применяются классические образовательные технологии. Дистанционные образовательные технологии, электронное обучение, а также сетевое обучение не реализуется

В частности, предусмотрено использование следующих образовательных технологий:

- лекция-визуализация – передача информации посредством схем, таблиц, рисунков, видеоматериалов, проводится по ключевым темам с комментариями;
- работа в команде – совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путем творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности,
- проблемное обучение – стимулирование студентов к самостоятельному приобретению знаний, необходимых для решения конкретной проблемы,
- обучение на основе опыта – активизация познавательной деятельности студента за счет ассоциации и собственного опыта с предметом изучения,
- опережающая самостоятельная работа – изучение студентами нового материала до его изучения в ходе аудиторных занятий,

При реализации дисциплины применяются интерактивные формы проведения практических занятий в виде проблемного обучения.

8. Методические указания по освоению дисциплины

Содержание учебной программы дисциплины «Инструментальные средства информационных систем» реализуется посредством лекций и лабораторных занятий, а также самостоятельной работы студентов.

Познавательная активность студентов на лабораторных занятиях обеспечивается рациональным сочетанием словесных, наглядных и практических методов с элементами обучения на основе опыта, работой с различными информационными источниками, решением познавательных и практик ориентированных задач.

Дисциплина состоит из 7 разделов.

Раздел 1. Введение. Цели и задачи дисциплины.

Лекции -2ч., СРС- 6 ч.

На изучение данной темы отводится двалеционных часа, которые включают в себя материал, перечисленный ниже.

Цели и задачи дисциплины. Объем дисциплины и виды учебной работы. Понятие инструментальных средств и примеры их реализации. Примеры применения инструментальных средств, при разработке и эксплуатации информационных систем.

В ходе лекционного курса проводится систематическое изложение современных научных материалов, освещение основных проблем. В тетради для конспектирования лекций необходимо иметь поля, где по ходу конспектирования студент делает необходимые пометки. Записи должны быть избирательными, полностью следует записывать только определения. В конспектах рекомендуется применять сокращения слов, что ускоряет запись.

Раздел 2. Кодирование алгоритмов.

Лекции -2ч., КСР-1 ч., СРС- 10ч.

На изучение данной темы отводится два лекционных часа, которые включают в себя материал, перечисленный ниже.

Языки программирования высокого и низкого уровня, их особенности. Структурное программирование. Объектно-ориентированное программирование. Интегральные среды разработки программного обеспечения. Специализированные редакторы программного кода.

Для достижения максимальной эффективности усвоения материала по данному разделу рекомендуется в ходе лабораторной работы №1 использовать форму работы – работа в команде. Продолжительность выполнения одной лабораторной работы 4 часа.

Раздел 3. Кодирование пользовательского интерфейса.

Лекции - 2ч., лабораторные занятия – 4ч., КСР-1 ч., СРС- 10 ч.

На изучение данной темы отводится два лекционных часа, которые включают в себя материал, перечисленный ниже.

Средства визуальной разработки пользовательского интерфейса. Совместное использование средств визуальной разработки и программного кода для формирования пользовательского интерфейса.

Для достижения максимальной эффективности усвоения материала по данному разделу рекомендуется в ходе лабораторной работы №2 использовать форму работы – работа в команде. Продолжительность выполнения одной лабораторной работы 4 часа.

Раздел 4. Отладка и тестирование.

Лекции - 2ч., лабораторные занятия – 4ч., КСР-1 ч., СРС- 10 ч.

На изучение данной темы отводится два лекционных часа, которые включают в себя материал, перечисленный ниже.

Средства отладки: отладчики и отладочные библиотеки. Инструменты отладки программных модулей: трассировка, точки прерывания, отслеживание переменных. Модификация программного кода для отладки. Отладка без средств отладки.

Тестирование как поиск логических ошибок. Стратегии тестирования. Средства автоматизированного тестирования. Организационная поддержка тестирования.

Для достижения максимальной эффективности усвоения материала по данному разделу рекомендуется в ходе лабораторной работы №3, использовать форму работы – работа в команде. Продолжительность выполнения одной лабораторной работы 4 часа.

Раздел 5. Средства визуализации.

Лекции - 2ч., лабораторные занятия – 4ч., СРС- 10 ч.

На изучение данной темы отводится два лекционных часа, которые включают в себя материал, перечисленный ниже.

Понятие виртуальной среды исполнения и виртуальной машины. Основные сценарии использования. Примеры виртуальных машин и сред исполнения. Виртуальные ресурсы и средства их формирования в операционной системе *Microsoft Windows*.

Для достижения максимальной эффективности усвоения материала по данному разделу рекомендуется в ходе лабораторной работы №4, использовать форму работы – работа в команде. Продолжительность выполнения одной лабораторной работы 4 часа.

Раздел 6. Защита данных

Лекции - 2ч., лабораторные занятия – 4ч., СРС- 10 ч.

На изучение данной темы отводится два лекционных часа, которые включают в себя материал, перечисленный ниже.

Разграничение доступа и шифрование данных. Несимметричное шифрование и электронные подписи. Разграничение доступа к данным в *Microsoft Windows*. Средства шифрования и подписывания данных.

Для достижения максимальной эффективности усвоения материала по данному разделу рекомендуется в ходе лабораторной работы №5, использовать форму работы – работа в команде. Продолжительность выполнения одной лабораторной работы 4 часа.

Раздел 7. Заключение.

Лекции - 2ч., СРС- 6 ч.

На изучение данной темы отводится три лекционных часа, которые включают в себя материал, перечисленный ниже.

Обзор изученных инструментальных средств. Направления развития инструментальных средств и технологий разработки информационных систем.

Рекомендуемые виды самостоятельных работ: конспектирование, реферирование, составление опорных схем, составление аннотированных каталогов и аналитических обзоров информационных ресурсов.

9. Материально-техническое обеспечение дисциплины

Для лабораторных занятий по дисциплине «Инструментальные средства информационных систем» используются компьютерные лаборатории 6-301 и 6-303, оснащенные персональными компьютерами с типом процессора Intel, объединенных в локальную сеть, с установленными на них операционными системами *Windows*, поддерживающими выход в глобальную сеть *Internet*, оснащенных антивирусными программными средствами.

Для проведения лекций визуализации по дисциплине «Инструментальные средства информационных систем» могут быть использованы 2 мультимедийных проектора.

10. Адаптация рабочей программы для лиц с ОВЗ

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.

Лист согласования рабочей программы дисциплины

Направление подготовки: 09.03.02 Информационные системы и технологии
код и наименование

Дисциплина: Инструментальные средства информационных систем

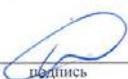
Направленность подготовки(профиль) Геоинформационные системы

Учебный год 2015/2016

РЕКОМЕНДОВАНА заседанием кафедры геоинформационных систем
наименование кафедры

протокол № 10 от "26" 06 2015 г.

Заведующий кафедрой  Христодуло О. И. 26.06.15
подпись расшифровка подписи

Исполнители:
к.т.н, доцент  Плеханов С. В.
должность подпись расшифровка подписи

СОГЛАСОВАНО:

Заведующий кафедрой¹

наименование кафедры личная подпись расшифровка подписи дата

Председатель НМС по УГСН 09.00.00 Информатика и вычислительная техника

протокол № 3 от "28" 08 2015 г.

 Фрид А. И. 28.08.15
личная подпись расшифровка подписи

Библиотека  Т.В. Р. ... 28.08.15
личная подпись расшифровка подписи дата

Декан факультета ИРТ  Юсупова Н.И. 28.08.15
личная подпись расшифровка подписи дата

Рабочая программа зарегистрирована в ООПИС и внесена в электронную базу данных

Начальник  А.Н. ... 28.08.15
личная подпись расшифровка подписи дата

¹ Согласование осуществляется с выпускающими кафедрами (для рабочих программ, подготовленных на кафедрах, обеспечивающих подготовку для других направлений и специальностей)

2789

**Дополнения и изменения в рабочей программе учебной дисциплины
на 20__/20__ уч. год**

УТВЕРЖДАЮ

Декан факультета (директор института, филиала)

ФИО

(подпись)

« ____ » _____ 20__ г.

В рабочую программу по дисциплине _____
для направления _____
направленность (программа) _____
вносятся следующие изменения:

1)

.....

...

2)

.....

...

ПЕРЕСМОТРЕНА на заседании кафедры _____
наименование кафедры

протокол № _____ от " ____ " _____ 20__ г.

Заведующий кафедрой _____
подпись расшифровка подписи

Научный руководитель магистерской
программы¹ _____
подпись расшифровка подписи

ОДОБРЕНА на заседании НМС по УГСН _____

протокол № _____ от " ____ " _____ 20__ г.

Председатель _____
личная подпись расшифровка подписи

СОГЛАСОВАНО:

Заведующий кафедрой²

наименование кафедры личная подпись расшифровка подписи дата

Библиотека³ _____
личная подпись расшифровка подписи дата

Дополнения и изменения внесены в базу данных рабочих программ дисциплин

Начальник ООПМА _____
личная подпись расшифровка подписи дата

¹ Только направлений подготовки магистров

² Согласование осуществляется с выпускающими кафедрами (для рабочих программ, подготовленных на кафедрах, обеспечивающих подготовку для других направлений подготовки научно-педагогических кадров высшей квалификации)

³ Только при внесении изменений в список литературы