

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«БРАТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра математики и физики

УТВЕРЖДАЮ:

Проректор по учебной работе

_____ Е.И. Луковникова

« _____ » _____ 201__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

СИСТЕМНОЕ И ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Б1.В.04

НАПРАВЛЕНИЕ ПОДГОТОВКИ

01.03.02 Прикладная математика и информатика

ПРОФИЛЬ ПОДГОТОВКИ

Инженерия программного обеспечения

Программа академического бакалавриата

Квалификация (степень) выпускника: бакалавр

| | |
|--|-----------|
| 1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ | 3 |
| 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ | 4 |
| 3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ | 4 |
| 3.1 Распределение объёма дисциплины по формам обучения..... | 4 |
| 3.2 Распределение объёма дисциплины по видам учебных занятий и трудоемкости | 5 |
| 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ | 5 |
| 4.1 Распределение разделов дисциплины по видам учебных занятий | 5 |
| 4.2 Содержание дисциплины, структурированное по разделам и темам | 6 |
| 4.3 Лабораторные работы..... | 7 |
| 4.4 Семинары / практические занятия | 7 |
| 4.5 Контрольные мероприятия: курсовой проект (курсовая работа), контрольная работа, РГР, реферат..... | 7 |
| 5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ | 10 |
| 6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ | 11 |
| 7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ..... | 11 |
| 8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО – ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ | 12 |
| 9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ..... | 12 |
| 9.1 Методические указания для обучающихся по выполнению лабораторных работ | 12 |
| 10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ | 27 |
| 11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ | 28 |
| Приложение 1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине..... | 29 |
| Приложение 2. Аннотация рабочей программы дисциплины | 34 |
| Приложение 3. Протокол о дополнениях и изменениях в рабочей программе | 35 |
| Приложение 4. Фонд оценочных средств для текущего контроля успеваемости по дисциплине..... | 36 |

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Вид деятельности выпускника

Дисциплина охватывает круг вопросов, относящихся к базовому виду профессиональной деятельности выпускника в соответствии с компетенциями и видами деятельности, указанными в учебном плане. Способствует формированию базовых понятий, терминологии и практических навыков работы с программным обеспечением, необходимых для практической работы по специальности и при изучения других дисциплин в сфере информатики тем или иным образом связанных с системным программным обеспечением. Поскольку любой программный продукт работает в тесной взаимосвязи с системным программным обеспечением и обязательно должен учитывать особенности его функционирования.

Цель дисциплины

Познакомить обучающихся с основными современными системными и прикладными программными продуктами, тенденциями их развития, обучить обучающихся принципам работы в различных программных пакетах, проведению анализа полученных результатов, применению современных программ в профессиональной деятельности.

Задачи дисциплины

- раскрыть значение дисциплины в общем и профессиональном образовании человека;
- приобрести навыки работы с основным программным обеспечением;
- сформировать умения и навыки самостоятельного решения прикладных задач с помощью современного прикладного и программного обеспечения.

| Код компетенции | Содержание компетенций | Перечень планируемых результатов обучения по дисциплине |
|-----------------|--|--|
| 1 | 2 | 3 |
| ОПК-3 | способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям | знать: синтаксис и семантику алгоритмических конструкций языков программирования высокого уровня и СУБД; базовые структуры данных, средства компьютерной графики и основные численные алгоритмы; уметь: разрабатывать математические и информационные модели и алгоритмы для решения прикладных задач, владеть: навыками работы с системным и прикладным обеспечением для решения задач математического моделирования в предметной области, а также современным программным обеспечением, средствами тестирования, документации ПО. |

| | | |
|-------|---|---|
| ОПК-4 | способность решать стандартные задачи профессиональной деятельности на основе информационно-библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности | <p>знать: методы сбора и обработки и хранения информации, причины нарушения компьютерной безопасности, методы разработки программного обеспечения;</p> <p>уметь: использовать ресурсы сети Интернет для разработки программного обеспечения и программной документации с учетом требований информационной безопасности;</p> <p>владеть: навыками системного и объектно-ориентированного программирования для решения стандартных прикладных задач в профессиональной деятельности; базовыми знаниями по защите информации на рабочем месте, в корпоративных сетях при входе в глобальные сети.</p> |
| ПК-2 | способность понимать, совершенствовать и применять современный математический аппарат | <p>знать: основные понятия дисциплины, её методы, место и роль в решении научно-практических задач с использованием современного математического аппарата;</p> <p>уметь: применять и совершенствовать современный математический аппарат при решении научно-практических задач прикладной математики и информатики;</p> <p>владеть: инструментарием для решения математических задач в области прикладной математики и информатики.</p> |

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина Системное и прикладное программное обеспечение относится к базовому циклу, вариативной части (код Б1.В.04) и является обязательной для изучения.

Представляет основу для изучения дисциплин: языки и методы программирования, базы данных, операционные системы, системное программирование, параллельное программирование, программные средства защиты информации.

Такое системное междисциплинарное изучение направлено на достижение требуемого ФГОС уровня подготовки по квалификации бакалавр.

3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ

3.1. Распределение объема дисциплины по формам обучения

| Форма обучения | Курс | Семестр | Трудоемкость дисциплины в часах | | | | | | контрольная работа | Вид промежуточной аттестации |
|-------------------------------|------|---------|---------------------------------|------------------|--------|---------------------|-------------------------------|------------------------|--------------------|------------------------------|
| | | | Всего часов (с экз.) | Аудиторных часов | Лекции | Лабораторные работы | Семинары Практические занятия | Самостоятельная работа | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Очная | 1 | 2 | 162 | 72 | 18 | 54 | - | 90 | кр | экзамен |
| Заочная | - | - | - | - | - | - | - | - | - | - |
| Заочная (ускоренное обучение) | - | - | - | - | - | - | - | - | - | - |
| Очно-заочная | - | - | - | - | - | - | - | - | - | - |

3.2. Распределение объема дисциплины по видам учебных занятий и трудоемкости

| Вид учебных занятий | Трудо- ем- кость (час.) | в т.ч. в интерактивной, активной, инновационной формах, (час.) | Распределение по семестрам, час |
|--|----------------------------------|--|------------------------------------|
| | | | 2 |
| 1 | 2 | 3 | 4 |
| I. Контактная работа обучающихся с преподавателем (всего) | 72 | 32 | 72 |
| Лекции (Лк) | 18 | 6 | 18 |
| Лабораторные работы (ЛР) | 54 | 26 | 54 |
| Контрольная работа | + | + | + |
| Групповые (индивидуальные) консультации | + | + | + |
| II. Самостоятельная работа обучающихся (СР) | 90 | - | 90 |
| Подготовка к лабораторным работам | 50 | - | 50 |
| Подготовка к экзамену в течение семестра | 20 | - | 20 |
| Выполнение контрольной работы | 20 | - | 20 |
| III. Промежуточная аттестация экзамен | 54 | - | 54 |
| Общая трудоемкость дисциплины час. зач. ед. | 216 | - | 216 |
| | 6 | - | 6 |

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Распределение разделов дисциплины по видам учебных занятий

- для очной формы обучения:

| № раз- дела и темы | Наименование раздела и тема дисциплины | Трудо- ем- кость, (час.) | Виды учебных занятий, включая само- стоятельную работу обучающихся и трудоемкость; (час.) | | |
|--------------------------|--|-----------------------------------|---|-----------------------------|--|
| | | | учебные занятия | | самостоя- тельная ра- бота обу- чающихся* |
| | | | лекции | лабора- торные работы | |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1. | Системное программное обеспе- чение. | 64 | 8 | 16 | 40 |
| 1.1. | Основные понятия. Определение операци- онной системы. | 16 | 2 | 4 | 10 |
| 1.2. | Управление задачами и памятью. | 12 | 2 | - | 10 |
| 1.3. | Управление вводом/выводом и файловые системы. | 12 | 2 | - | 10 |
| 1.4. | Архитектура ОС и интерфейсы приклад- ного программирования. | 24 | 2 | 12 | 10 |
| 2. | Прикладное программное обес- печение и системы. | 98 | 10 | 38 | 50 |
| 2.1. | Понятия технологии программирования. | 20 | 1 | 14 | 5 |
| 2.2. | Требования и спецификации программы. | 6 | 1 | - | 5 |
| 2.3. | Структура программы. | 12 | 2 | - | 10 |
| 2.4. | Испытания программы. | 12 | 2 | - | 10 |
| 2.5. | Внедрение и сопровождение программы. | 12 | 2 | - | 10 |
| 2.6. | Современные методы и средства проекти- рования программ. | 12 | 2 | 24 | 10 |
| | ИТОГО | 162 | 18 | 54 | 90 |

4.2. Содержание дисциплины, структурированное по разделам и темам

| <i>№ раздела и темы</i> | <i>Наименование раздела и темы дисциплины</i> | <i>Содержание лекционных занятий</i> | <i>Вид занятия в интерактивной, активной, инновационной формах, (час.)</i> |
|-------------------------|---|---|--|
| 1 | 2 | 3 | 4 |
| 1. | Системное программное обеспечение. | | |
| 1.1. | Основные понятия. Определение операционной системы. | Понятие вычислительного процесса и ресурса: диаграмма состояний процесса, реализация понятия последовательного процесса в ОС. Прерывания. Основные виды ресурсов. Классификация операционных систем: однозадачные мониторы, пакетные мониторы, мультипрограммные пакетные ОС, диалоговые многопользовательские ОС, ОС реального времени. | Лекция-беседа (2 часа) |
| 1.2. | Управление задачами и памятью. | Планирование и диспетчеризация процессов и задач: стратегии планирования, дисциплины диспетчеризации, диспетчеризация задач с использованием динамических приоритетов. Память и отображения, виртуальное адресное пространство. Распределение памяти разделами, сегментная, страничная и сегментно-страничная организация памяти. Распределение оперативной памяти в современных ОС. | |
| 1.3. | Управление вводом/выводом и файловые системы. | Основные понятия и концепции ввода/вывода в ОС. Основные системные таблицы ввода/вывода. Синхронный и асинхронный ввод/вывод. Функции файловой системы ОС. Файловая система FAT. Файловая система NTFS. Основные отличия FAT и NTFS. | |
| 1.4. | Архитектура ОС и интерфейсы прикладного программирования. | Основные принципы построения ОС. Принципы построения интерфейсов ОС. Интерфейс прикладного программирования. | Разбор конкретных ситуаций (2 часа) |
| 2. | Прикладное программное обеспечение и системы. | | |
| 2.1 | Понятия технологии программирования. | Программа. Компонент программной системы. Программная система. Программный продукт. Программное изделие. Классификация программных систем по сложности. Основные характеристики и критерии качества программ и программных систем. Модели жизненного цикла программных систем. Стадии и этапы разработки программных систем и программной документации. | Проблемная Лекция (1 час) |
| 2.2. | Требования и спецификации программы. | Понятия требований к программе. Функциональные и нефункциональные требования. Стандартизация описания требований. Техническое задание на разработку программы. Понятие спецификаций программы. Роль адекватных, корректных и полных спецификаций на всех этапах проектирования программы. Функциональные спецификации. Методы описания спецификаций. Интерфейсные спецификации. Проверка правильности спецификаций. | |
| 2.3. | Структура программы. | Разработка структуры программы. Оценка структуры программы. Методы разработки структуры программы. Нисходящие и восходя- | |

| | | | |
|------|--|---|-------------------------------------|
| | | щие методы. Модуль. Внешние спецификации модуля. Способы описания спецификаций. Проектирование логики модуля. Документирование модуля. | |
| 2.4. | Испытания программы. | Верификация, тестирование и отладка программы. Основные методы верификации программы. Тестирование и отладка. Методы тестирования. План тестирования. Заглушки и драйверы. Использование инструментальных средств тестирования и отладки. | Разбор конкретных ситуаций (2 часа) |
| 2.5. | Внедрение и сопровождение программы. | Задачи сопровождения программы. Расширение возможностей, адаптация и коррекция. Средства и методы сопровождения. Организация сопровождения. Эксплуатационная документация. | Проблемная Лекция (2 часа) |
| 2.6. | Современные методы и средства проектирования программ. | Методы и средства структурного анализа и проектирования: диаграммы потоков данных, словарь данных, спецификации процессов, диаграммы «сущность – связь», спецификации управления. Средства структурного проектирования. Унифицированный язык моделирования UML: диаграммы вариантов использования, диаграммы классов, диаграммы состояний, диаграммы деятельности, диаграммы последовательности, диаграммы компонентов. | |

4.3. Лабораторные работы

| <i>№ п/п</i> | <i>Номер раздела дисциплины</i> | <i>Наименование тем лабораторных работ</i> | <i>Объем в часах</i> | <i>Вид занятия в инновационной форме</i> |
|--------------|---------------------------------|--|----------------------|--|
| 1. | 1. | Создание дистрибутива программы. | 4 | - |
| 2. | | Создание приложения с функцией защиты от несанкционированного копирования. | 6 | Работа в малых группах (4 часа) |
| 3. | | Приложение «Курс валют» с обновлением данных из интернета. | 6 | Тренинговая работа в малых группах (8 часов) |
| 4. | 2. | Использование компонент для работы с графикой. Создание программы, использующей графические компоненты для иллюстрации данных. | 6 | - |
| 5. | | Использование диаграмм и бизнес-графики. | 8 | Анализ ситуаций (6 часов) |
| 6. | | Использование динамических библиотек. | 8 | - |
| 7. | | Создание многопоточных приложений для параллельной обработки информации. | 8 | - |
| 8. | | Архивация данных по расписанию. | 8 | Работа в малых группах (6 часов) |
| ИТОГО | | | 54 | 28 |

4.4. Семинары/ практические занятия

учебным планом не предусмотрено.

4.5. Контрольные мероприятия: контрольная работа

Контрольная работа по дисциплине «Системное и прикладное программное обеспечение» является самостоятельной творческой работой обучающегося и предполагает выполнение следующих заданий практических заданий с последующей демонстрацией их на ПК.

Работа выполняется по индивидуальным вариантам.

Цель: проверка знаний, умений и навыков работы с прикладным программным обеспечением.

Структура: контрольная работа содержит задания, которые выполняются в программных приложениях. Выполняется индивидуально, отчет должен содержать подробное описание алгоритма действий с рисунками.

Номер варианта для всех заданий определяется по порядковому номеру обучающегося в списке группы.

Основная тематика:

Вопрос №1 Работа в MS Word Задание к вопросу. Создайте гипертекстовый документ на заданную тему. Тема должна быть раскрыта полностью.

Вариант 1 Реализация информационных технологий в промышленности, в управлении и в обучении.

Вариант 2 Средства передачи данных.

Вариант 3 Сетевые технологии.

Вариант 4 Операционные системы.

Вариант 5 Технологии накопления и хранения информации.

Вариант 6 Базы данных. Базы знаний. Хранилища данных.

Вариант 7 Объектно-ориентированные среды.

Вариант 8 Функциональное программирование

Вариант 9 Логическое программирование.

Вариант 10 Способы обработки информации.

Вопрос №2 Работа в MS Excel Задание к вопросу. Создайте СТИПЕНДИАЛЬН ВЕДОМОСТЬ ФАКУЛЬТЕТА. Представьте, что на факультете 5 курсов, на каждом курсе 2 группы, в группах по 25 человек. В таблице используйте данные: ФИО обучающегося, успеваемость (средний балл за сессию), сумма, надбавки за отличную и хорошую учебу. Стипендия обучающимся, имеющим балл ниже 3,5, не начисляется (в соответствующей графе указать 0). Подготовьте отчеты по указанным в вариантах заданиям.

Вариант 1 Сформируйте сводную ведомость обучающихся с отличной учебой. Выдайте диаграмму с долей таких обучающихся. Создайте отчеты по каждому курсу с графическим отображением. Оформите диаграммы распределения отличников по группам курса и по курсам.

Вариант 2 Сформируйте сводную ведомость обучающихся со средней успеваемостью. Выдайте диаграмму с долей таких обучающихся. Создайте отчеты по каждому курсу с графическим отображением. Оформите диаграммы распределения обучающихся со средней успеваемостью по группам курса и по курсам.

Вариант 3 Сформируйте сводную ведомость неуспевающих обучающихся. Выдайте диаграмму с долей таких обучающихся. Создайте отчеты по каждому курсу с графическим отображением. Оформите диаграммы распределения неуспевающих обучающихся по группам курса и по курсам.

Вариант 4 Пусть первоначально составленная ведомость определяет фиксированный стипендиальный фонд факультета. Отмените выдачу стипендии для обучающихся, имеющих средний балл успеваемости ниже 4. Перераспределите экономию стипендиального фонда для каждой группы и для каждого курса в зависимости от доли отличников. Выдайте соответствующие отчеты.

Вариант 5 Пусть первоначально составленная ведомость определяет фиксированный стипендиальный фонд факультета. Отмените надбавки за отличную учебу. Перераспределите экономию стипендиального фонда для каждой группы и для каждого курса в зависимости от доли обучающихся, получающих стипендию. Выдайте соответствующие отчеты.

Вариант 6 Пусть первоначально составленная ведомость определяет фиксированный стипендиальный фонд факультета. Отмените выдачу стипендии для обучающихся, имеющих средний балл успеваемости ниже 4,5. Экономию стипендиального фонда перераспределите

всем обучающимся пропорционально их успеваемости. Составьте диаграммы роста размера стипендии для успевающих обучающихся (балл выше 4,5).

Вариант 7 Пусть первоначально составленная ведомость определяет фиксированный стипендиальный фонд факультета. Перераспределите заданный фонд всем (без исключения) обучающимся. Составьте диаграммы изменения размера стипендии для трех категорий обучающихся (отличники, успевающие, неуспевающие).

Вариант 8 Отчислите из каждой группы произвольным образом по три обучающегося (не только неуспевающих!). Выполните задание варианта 1.

Вариант 9 Отчислите из каждой группы произвольным образом по несколько обучающихся, не менее 5 в каждой группе (не только неуспевающих!). Выполните задание варианта 2.

Вариант 10 Пусть первоначально составленная ведомость определяет фиксированный стипендиальный фонд факультета. Внесите в каждую группу дополнительно до трех обучающихся с разной успеваемостью. Перераспределите стипендиальный фонд в основном за счет лишения стипендии обучающихся с низким баллом успеваемости. Составьте отчет по изменению числа обучающихся, получающих стипендию, по курсам.

Вопрос №3 Работа в MS PowerPoint Задание к вопросу. Создайте интерактивный кроссворд на заданную тему.

Вариант 1 Архитектура компьютерной техники.

Вариант 2 История развития вычислительной техники.

Вариант 3 Программное обеспечение вычислительной техники.

Вариант 4 Коммуникационные технологии.

Вариант 5 Компьютерная безопасность.

Вариант 6 Алгоритмизация и программирование.

Вариант 7 Сетевые технологии.

Вариант 8 Операционные системы.

Вариант 9 Периферийное оборудование.

Вариант 10 Информация.

Рекомендуемый объем: 25- 30 машинописных листов.

Выдача, задания, прием кр проводится в соответствии с календарным учебным графиком.

| Оценка | Критерии оценки контрольной работы |
|-------------------|--|
| Зачтено | «Зачтено» ставится при условии правильного выполнения 6 заданий и более. |
| Не зачтено | Если не выполнено два и более задания, то студент получает оценку «Не зачтено» и не допускается к промежуточному контролю: экзамену. |

5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

| <i>Компетенции</i> <i>№, наименование разделов дисциплины</i> | <i>Кол-во часов</i> | <i>Компетенции</i> | | | <i>Σ комп.</i> | <i>t_{ср}, час</i> | <i>Вид учебных занятий</i> | <i>Оценка результатов</i> |
|--|---------------------|--------------------|------------|-----------|----------------|----------------------------|----------------------------|---------------------------|
| | | <i>ОПК</i> | <i>ОПК</i> | <i>ПК</i> | | | | |
| | | <i>3</i> | <i>4</i> | <i>2</i> | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1. Системное программное обеспечение. | 64 | + | + | + | 3 | 21,3 | Лк, ЛР, СР | кр, экзамен |
| 2. Прикладное программное обеспечение и системы. | 98 | + | + | + | 3 | 32,7 | Лк, ЛР, СР | кр, экзамен |
| <i>всего часов</i> | 162 | 54 | 54 | 54 | 2 | 54 | | |

6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Гусева Е.Н. Информатика:[электронный ресурс]: учебное пособие /Е.Н. Гусева, Ю.Н. Ефимова, Р.И. Коробкова, И.Н. Мовчан, Л.А. Савельева. – 3-е изд. стереотип. - М.: -Флинта. – 2011. – 260 с. http://biblioclub.ru/index.php?page=book_view_red&book_id=83542

7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

| № | Наименование издания (автор, заглавие, выходные данные) | Вид заня- тия (Лк, ЛР, кр) | Количество экземпляров в библиоте- ке, шт. | Обеспечен- ность, (экз./ чел.) |
|----------------------------------|---|--|--|--------------------------------------|
| 1 | 2 | 3 | 4 | 5 |
| Основная литература | | | | |
| 1. | Царев Р.Ю. Программные и аппаратные средства информатики: учеб./Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков.-Красноярск: Сиб. федер.ун-т.-2015.-160с. http://biblioclub.ru/index.php?page=book_view_red&book_id=435670 | Лк, ЛР, кр | 1 (ЭУ) | 1 |
| 2. | Гусева Е.Н. Информатика [электронный ресурс] учебное пособие/ Е.Н. Гусева, И.Ю. Ефимова, К.В. Коробкова, И.Н. Мовчан, Л.А. Савельева.-3- е изд., стереотип.-М.: ФЛИНТА.-2011.-260 с. http://biblioclub.ru/index.php?page=book_view_red&book_id=83542 | Лк, ЛР, кр | 1 (ЭУ) | 1 |
| 3. | Информатика 1: учебное пособие/ И.Л. Артемов, А.В. Гураков, О.И. Мещерякова, П.С. Мещеряков, Д.С. Шульц.- Томск: ФДО ТУСУР.-2015.-234 с. http://biblioclub.ru/index.php?page=book_view_red&book_id=480593 | Лк, ЛР, к | 1 (ЭУ) | 1 |
| Дополнительная литература | | | | |
| 4. | Кремень Е.В. Основы работы в Windows: учебный справочник/Е.В. Кремень, Ю.А. Кремень.-Минск: ТетраСистемс, 2011.- 176 с. – (Электронный гражданин) http://biblioclub.ru/index.php?page=book_view_red&book_id=78522 | Лк, ЛР, кр | 1 (ЭУ) | 1 |
| 5. | Абрамова Л.В. Инструментальные средства информационных систем: учебное пособие/Л.В. Абрамова; Сев. (Арктич.) федер. ун-т им. М.В. Ломоносова.- Архангельск: ИПЦ САФУ.-2013.-118 с. http://biblioclub.ru/index.php?page=book_view_red&book_id=436131 | Лк, ЛР, кр | 1 (ЭУ) | 1 |

8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Электронный каталог библиотеки БрГУ http://irbis.brstu.ru/CGI/irbis64r_15/cgiirbis_64.exe?LNG=&C21COM=F&I21DBN=BOOK&P21DBN=BOOK&S21CNR=&Z21ID=.
2. Электронная библиотека БрГУ <http://ecat.brstu.ru/catalog>.
3. Электронно-библиотечная система «Университетская библиотека online» <http://biblioclub.ru>.
4. Электронно-библиотечная система «Издательство «Лань» <http://e.lanbook.com>.
5. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru>.
6. Научная электронная библиотека eLIBRARY.RU <http://elibrary.ru>.
7. Университетская информационная система РОССИЯ (УИС РОССИЯ) <https://uisrussia.msu.ru/>.
8. Национальная электронная библиотека НЭБ <http://xn--90ax2c.xn--p1ai/how-to-search/>.

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Успешное освоение дисциплины предполагает активную работу обучающегося на лекционных занятиях, лабораторных работах, выполнение контрольных мероприятий, самостоятельную работу.

В процессе подготовки к контрольной работе необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы, поиск дополнительных материалов.

Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса.

Обязательным дополнением лекционного курса являются лабораторные работы. Они помогают получить навыки работы на персональном компьютере в наиболее распространенных программных продуктах – операционной системе, текстовом и табличном процессоре, графических редакторах и др. в каждой лабораторной работе сначала излагается изучаемый материал, объясняются примеры, выполняются упражнения, а потом предлагаются задания для самостоятельного выполнения.

Требования к выполнению лабораторных работ:

1. внимательно прочитать задания,
2. изучить вспомогательный материал,
3. усвоить порядок выполнения и следовать инструкции,
4. выполнить задания для самостоятельного решения.

При выполнении приведенных выше рекомендаций подготовка к экзамену сведется к повторению изученного и совершенствованию навыков применения теоретических положений и различных методов к стандартным и нестандартным заданиям.

9.1. Методические указания для обучающихся по выполнению лабораторных работ/семинаров /практических работ

Лабораторная работа №1 Создание дистрибутива программы.

Цель работы: Изучить назначение и особенности дистрибутивов программ. Получить навыки создания дистрибутива с помощью программы InnoSetup.

Задание:

Создать и протестировать дистрибутив программы с помощью freeware программы InnoSetup. В качестве устанавливаемой программы можно использовать notepad.exe из

системного каталога Windows, или любую пользовательскую программу. Дистрибутив должен предлагать каталог для установки по умолчанию, возможность выбора каталога, обеспечивать вывод информации об устанавливаемой программе, должен присутствовать деинсталлятор.

Варианты заданий:

1. Установка по заданному паролю.
2. Два языковых пакета при установке.
3. Записать ключ в системный реестр.
4. Перезагрузить компьютер после установки.
5. Произвести установку в ускоренном режиме.

Порядок выполнения:

1. Изучить назначение и характер работы программ-дистрибутивов.
2. Изучить прикладную программу для создания установочных приложений на примере InnoSetup.
3. Изучить возможности и базовые понятия скриптовых языков на примере встроенного языка InnoSetup.
4. Ответить на контрольные вопросы.
5. Получить у преподавателя номер варианта задания.
6. Разработать и отладить программу.

Форма отчетности:

1. Распечатка исходного текста программы с комментариями.
2. Демонстрация работы программы.

Задания для самостоятельной работы:

Проанализировать дистрибутивы, которые могут предлагать каталог для установки по умолчанию, возможность выбора каталога, обеспечивать вывод информации об устанавливаемой программе, должен присутствовать деинсталлятор.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Способы создания и редактирования скриптов

Скрипт можно создавать двумя способами: вручную или с помощью специализированных инструментов типа ISTools (также достаточно широко применяется InnoSetup Script Generator), которые сильно облегчают рутинные действия, кроме того обладают очень полезной функцией – возможностью импорта файлов реестра.

Пример создания простого скрипта

В качестве примера создадим скрипт установки программы ATNotes, который будет устанавливать приложение в выбранную пользователем папку (или папку, заданную по умолчанию), создавать ярлыки на Рабочем столе, в меню Пуск и в панели Быстрого запуска.

Прежде чем начать написание скрипта, нужно подготовить необходимые для установщика файлы. В нашем случае это непосредственно сами файлы программы. Для удобства перепишем их в папку «files» (пустой файл скрипта [назовём его atnotes.iss] и папку files предварительно нужно создать). Файл скрипта, как и говорилось ранее, можно создавать с помощью мастера в программе ISTools, заполнив необходимые поля или вручную.

При написании скрипта вручную он получается чуть более коротким, т. к. мастер автоматически проставляет все параметры, которые часто необязательны.

Скрипт состоит из секций [Setup], [Files] и [Icons], из них первая секция является обязательной, без неё невозможно скомпилировать (создать) дистрибутив.

Секций [Files] и [Icons] может и не быть, если они не нужны. Всего же может быть 18 различных секций с разным назначением и параметрами.

Кроме того, каждая из строк в секции может содержать несколько флагов, которые укажут компилятору как себя вести при распаковке данных файлов. В нашем случае использован флаг «ignoreversion», указывающий установщику перезаписывать файлы независимо от их версий (даже если существующий файл новее) и «dontcopy» означающий, что файл не будет распакован в папку назначения при установке (удобно хранить исходный файл скрипта в самом дистрибутиве).

При установке в ускоренном режиме окно мастера и фоновое окно мастера не отображаются, отображается только окно прогресса. При сверхускоренной установке окно

прогресса тоже не отображается. Остальная установка проходит как обычно – отображаются сообщения об ошибках в ходе установки, страница приветствия (если ее отображение не отменено директивой DisableStartupPrompt или параметром '/SP-' командной строки). Если после установки необходима перезагрузка, команды '/NORESTART' не дано и установка проходит в ускоренном режиме, появится сообщение «Перезагрузить компьютер? ».

Если режим установки сверхускоренный, система перезагрузится, не спрашивая. /LOG – указывает инсталлятору создать log-файл в папке TEMP для подробного отчета о действиях и событиях в секции [Run], предпринятых в ходе установки. Применяется при отслеживании ошибок.

Например, если вы подозреваете, что файл не был удален, хотя должен был, log-файл сообщит, что файл был действительно пропущен и почему. Log-файлу присваивается уникальное имя на базе текущей даты. (Существующие файлы не перезаписываются и не дополняются). Информация, содержащаяся в log-файле, техническая и поэтому не может быть понята пользователем. /LOG="filename" – параметр похож на /LOG, но в отличие от него позволяет указать фиксированный путь/имя файла для log-файла. Если файл с указанным именем уже существует, он будет заменен. Если файл не может быть создан, установка прервется и выдаст сообщение об ошибке. /NOCANCEL – не дает пользователю прервать установку, отключая кнопку Отменить и игнорируя клики на кнопке Закреть.

Основная литература

1 Царев Р.Ю. Программные и аппаратные средства информатики: учеб./Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков.-Красноярск: Сиб. федер.ун-т.-2015.-160с.

2 Гусева Е.Н. Информатика [электронный ресурс] учебное пособие/ Е.Н. Гусева, И.Ю. Ефимова, К.В. Коробкова, И.Н. Мовчан, Л.А. Савельева.-3- е изд., стереотип.-М.: ФЛИНТА.-2011.-260 с.

3 Информатика 1: учебное пособие/ И.Л. Артемов, А.В. Гураков, О.И. Мещерякова, П.С. Мещеряков, Д.С. Шульц.- Томск: ФДО ТУСУР.-2015.-234 с.

Дополнительная литература

1 Кремень Е.В. Основы работы в Windows: учебный справочник/Е.В. Кремень, Ю.А. Кремень.-Минск: ТетраСистемс, 2011.- 176 с. – (Электронный гражданин)

2 Абрамова Л.В. Инструментальные средства информационных систем: учебное пособие/Л.В. Абрамова; Сев. (Арктич.) федер. ун-т им. М.В. Ломоносова.-Архангельск: ИПЦ СА-ФУ.-2013.-118 с.

Контрольные вопросы для самопроверки

1. Для чего предназначаются дистрибутивы программ?
2. Отметьте основные возможности и варианты, предлагаемые пользователю при установке программ с помощью дистрибутива.
3. Для чего предназначен встроенный скриптовый язык прикладных программ для создания дистрибутивов?

Лабораторная работа №2 Создание приложения с функцией защиты от несанкционированного копирования.

Цель работы: Изучить способы защиты программных продуктов от несанкционированного копирования. Получить навыки создания программ с защитой от копирования.

Задание:

Создать и протестировать программу, содержащую, по крайней мере, два алгоритма защиты от несанкционированного копирования. Алгоритмы выбираются из вариантов заданий. Для создания программы используется среда программирования Turbo Explorer либо Turbo C++. Варианты заданий:

1. Пароль на запуск программы.
2. Ключ, генерируемый по аппаратным характеристикам компьютера пользователя (код жесткого диска, процессора и т.п.).
3. Ключ, генерируемый по программным характеристикам компьютера (метка жесткого диска, имя компьютера и т.п.).

4. Ограничение количества запусков программы.

Порядок выполнения:

1. Изучить виды защиты программ от копирования.
2. Изучить способы обхода стандартных мер защиты от копирования.
3. Ответить на контрольные вопросы.
4. Получить у преподавателя номер варианта задания.
5. Разработать и отладить программу.

Форма отчетности:

1. Распечатка исходного текста программы с комментариями.
2. Демонстрация работы программы.

Задания для самостоятельной работы:

Изучить способы обхода стандартных мер защиты от копирования.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Пароль на программу. В этом случае, конечно, программу можно скопировать, но работать она будет лишь у того пользователя, который знает пароль. Понятно, что такой метод имеет смысл лишь тогда, когда пользователь программы не заинтересован в ее тиражировании. В связи с этим, данный метод применяется ограниченно.

Чаще всего – при изготовлении некоей специализированной программы по индивидуальному заказу. Разновидность данного метода часто используется в дистрибутивах продукции «Microsoft» и других западных производителей. Для инсталляции программы необходимо ввести индивидуальный код продукта (пользователя).

Естественно, к нелегальным (контрафактным) версиям продукта прилагается такой код, взятый с того экземпляра, который служил оригиналом. Часто можно ввести и какой-либо другой взятый наугад номер. Не с первой, так со второй-третьей попытки удаётся найти приемлемый

Метод преодоления Проще всего узнать пароль. Если нет такой возможности, то надо прибегнуть к методам взлома. Обычно проверка пароля реализована в виде под- программы (процедуры, функции), которая возвращает значение типа «верный-неверный».

Следует найти вызов этой процедуры и отключить его, т. е. вместо команды перехода по условию вставить безусловный переход. Иногда проверка пароля несколько более сложная, чем просто переход по условию.

В таком случае взлом может не дать результата. Такой метод – нахождение процедуры и отмена перехода по условию – применим и к другим методам защиты. Разработчики, зная это, иногда пытаются застраховаться от подобного взлома. Например, вставляют проверку пароля в нескольких местах, с помощью разных процедур, которые трудно разыскать.

Есть и более действенный метод защиты от обхода проверки пароля. Процедура проверки пароля возвращает не бинарный результат проверки («верный-неверный»), а некое число, показывающее правильность пароля. При верном пароле оно равно величине, известной разработчикам, но не известной хакерам. Далее это число не проверяется на соответствие контрольному значению, а используется в работе программы, например, в команде перехода по адресу или в вычислительной процедуре.

При неверном пароле получается неверное значение этого числа. Программа при этом будет работать, но с непредсказуемым результатом – может и зависнуть, и испортить жёсткий диск. Если вы сами разрабатываете защиту программного продукта, возьмите на вооружение этот метод.

Только не забудьте предусмотреть страховку от ошибочного ввода пароля. Это можно сделать, например, при помощи хэш-функции.

Другой метод преодоления данного вида защиты – попробовать подобрать пароль, используя методы криптоанализа. Однако самое простое всё же – узнать пароль у законного пользователя программы.

Ограничение срока действия программы. Используется, как правило, в демо-версиях программ, но встречается и в лицензионных копиях, когда срок лицензии ограничен. С некоторой натяжкой данный метод можно отнести к методам защиты от копирования.

Суть метода понятна из названия. Реализуется это ограничение, как правило, через

проверку текущей даты по таймеру компьютера. Иногда правильность показаний таймера может контролироваться другими способами. Бывает, что ограничивается не календарный срок действия, а количество запусков программы. Тогда где-то в программе есть соответствующий счётчик. Метод преодоления Простейший метод преодоления ясен. Надо перевести часы назад.

В программе могут быть и дополнительные проверки. Как известно, во всех файловых системах имеется запись о дате создания файла. Программа может проверить заголовки некоторых файлов. И если найдёт среди них даты позже текущей, то заподозрит обман. Как можно справиться с такой проверкой, подумайте сами. Здесь также применимы методы взлома. Можно найти в программе место проверки даты и поставить обход.

Можно найти контрольную дату – дату окончания срока действия и изменить её. Но такие усилия редко оправданы. Проще менять дату или по истечении срока действия инсталлировать про- грамму заново. В последнем случае вы можете наткнуться на ещё один метод защиты, описанный ниже. Ограничение функций программы. Демо-версии программ очень широко используются в настоящее время.

Основная литература

1 Царев Р.Ю. Программные и аппаратные средства информатики: учеб./Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков.-Красноярск: Сиб. федер.ун-т.-2015.-160с.

2 Гусева Е.Н. Информатика [электронный ресурс] учебное пособие/ Е.Н. Гусева, И.Ю. Ефимова, К.В. Коробкова, И.Н. Мовчан, Л.А. Савельева.-3- е изд., стереотип.-М.: ФЛИНТА.-2011.-260 с.

3 Информатика 1: учебное пособие/ И.Л. Артемов, А.В. Гураков, О.И. Мещерякова, П.С. Мещеряков, Д.С. Шульц.- Томск: ФДО ТУСУР.-2015.-234 с.

Дополнительная литература

1 Кремень Е.В. Основы работы в Windows: учебный справочник/Е.В. Кремень, Ю.А. Кремень.-Минск: ТетраСистемс, 2011.- 176 с. – (Электронный гражданин)

2 Абрамова Л.В. Инструментальные средства информационных систем: учебное пособие/Л.В. Абрамова; Сев. (Арктич.) федер. ун-т им. М.В. Ломоносова.-Архангельск: ИПЦ СА-ФУ.-2013.-118 с.

Контрольные вопросы для самопроверки

1. Для чего необходимо защищать программу от несанкционированного копирования?
2. Перечислить меры защиты от копирования.
3. Перечислить программные способы защиты от копирования.
4. Перечислить программно-аппаратные способы защиты от копирования.
5. Перечислить методы взлома систем защиты от копирования, рассказать, как можно противодействовать этим методам?

Лабораторная работа №3 Приложение «Курс валют» с обновлением данных из интернета.

Цель работы: Изучить способы получения информации из интернета программным путем. Получить навыки создания программ, получающих данные из интернета.

Задание:

Создать и протестировать программу, реализующую вывод информации о курсе валюты на выбранную пользователем дату. Для получения информации о курсе валюты на любую дату использовать любой доступный webресурс, например, <http://conv.rbc.ru/convert.shtml>. Вид валюты выбирается из 38 вариантов заданий. Для создания программы используется среда программирования Turbo Explorer либо Turbo C++.

Варианты заданий:

1. Доллар США.
2. ЕВРО.
3. Австралийский доллар.
4. Английский фунт стерлингов.

Порядок выполнения:

1. Изучить компоненты и методы для программного доступа в интернет.
2. Изучить процедуры и функции для доступа в интернет.
3. Ответить на контрольные вопросы.
4. Получить у преподавателя номер варианта задания.
5. Разработать и отладить программу.

Форма отчетности:

1. Распечатка исходного текста программы с комментариями.
2. Демонстрация работы программы.

Задания для самостоятельной работы:

Изучить процедуры и функции для доступа в интернет.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Создать web -браузер с использованием TWebBrowser довольно легко, и займет не более одной минуты.

Открываем новый проект, центрируем форму, в нижней части формы создаем панель, на которой располагаем ComboBox для ввода URL, слева от нее соответствующий Label, справа кнопку «Go».

Разместим также главное действующее лицо TWebBrowser над панелью. Дадим имена нашим компонентам (например, ComboBox можно назвать «selUrl»), проставим соответствующие опции выравнивания.

Создадим обработчик нажатия на кнопку «Go»:

```

procedure TFormSimpleWB.btGoClick(Sender: TObject);
var
    _URL, Flags, TargetFrameName, PostData, Headers: Olevariant;
begin
    _URL := selUrl.Text;
    Flags := 0;
    TargetFrameName := 0;
    Postdata := 0;
    Headers := 0;
    WebBrowser1.Navigate2(_URL, Flags, TargetFrameName, PostData,
Headers);
end;

```

Этот метод дает команду WebBrowser начать навигацию к указанному ресурсу. Синтаксис этого метода следующий:

```

procedure Navigate(const URL: WideString; var Flags: OleVariant;
var TargetFrameName: OleVariant; var PostData: OleVariant;
var Headers: OleVariant); overload;

```

Здесь **URL** – адрес навигации.

Для значения **Flags** определены такие константы:

navOpenInNewWindow (1) – открывает URL в новом окне браузера по умолчанию. Как открыть URL в новом экземпляре нашего браузера будет описано ниже при описании обработчика OnNewWindow

navNoHistory (2) – не заносит адрес в список History.

navNoReadFromCache (4) – не использует сохраненную в кеше страницу, а загружает

с сервера.

navNoWriteToCache (8) – не записывает страницу в дисковый кеш.

navAllowAutosearch (16) – если браузер не может найти указанный домен, он передает его в поисковый механизм.

TargetFrameName – определяет целевой фрейм по имени.

Если присвоить system.NULL (или имя несуществующего фрейма) страница загрузится в текущий браузер.

PostData – определяет данные для передачи на сервер.

Headers – определяет HTTP-хидер для передачи на сервер.

property Busy – если это свойство активно (равно True), то это свидетельствует о том, что браузер еще не закончил загрузки страницы или выполняет некоторую команду. И он проигнорирует новую команду. Так что в этом случае лучше подождать, когда это свойство станет равным false (или, когда идет загрузка, то остановить ее можно с помощью метода Stop).

Ниже приведены примеры использования компонента для доступа к информации, размещенной на web-странице.

Как получить текст HTML документа из TWebBrowser?

```
uses
    ..., MSHTML;
procedure TForm1.FormCreate(Sender: TObject);
begin
    WebBrowser1.Navigate('http://decoding.narod.ru');
end;
procedure TForm1.Button1Click(Sender: TObject);
var
    Document: IHTMLDocument2;
begin
    Document := WebBrowser1.Document as IHTMLDocument2;
```

36

```
if Document <> nil then
    Memo1.Text :=
        (Document.all.Item(NULL, 0) as IHTMLElement).OuterHTML;
end;
```

Основная литература

1 Царев Р.Ю. Программные и аппаратные средства информатики: учеб./Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков.-Красноярск: Сиб. федер.ун-т.-2015.-160с.

2 Гусева Е.Н. Информатика [электронный ресурс] учебное пособие/ Е.Н. Гусева, И.Ю. Ефимова, К.В. Коробкова, И.Н. Мовчан, Л.А. Савельева.-3- е изд., стереотип.-М.: ФЛИНТА.-2011.-260 с.

3 Информатика 1: учебное пособие/ И.Л. Артемов, А.В. Гураков, О.И. Мещерякова, П.С. Мещеряков, Д.С. Шульц.- Томск: ФДО ТУСУР.-2015.-234 с.

Дополнительная литература

1 Кремень Е.В. Основы работы в Windows: учебный справочник/Е.В. Кремень, Ю.А. Кремень.-Минск: ТетраСистемс, 2011.- 176 с. – (Электронный гражданин)

2 Абрамова Л.В. Инструментальные средства информационных систем: учебное пособие/Л.В. Абрамова; Сев. (Арктич.) федер. ун-т им. М.В. Ломоносова.-Архангельск: ИПЦ СА-ФУ.-2013.-118 с.

Контрольные вопросы для самопроверки

1. Для чего программам нужен доступ в интернет?
2. Перечислить компоненты и методы, используемые для доступа в интернет в выбранной вами среде разработки.
3. Перечислить возможные ошибки, которые могут появиться при попытке получения информации из интернета.
4. Перечислить протоколы, используемые в интернете. Каким протоколом вы пользовались в программе для получения необходимой информации?

Лабораторная работа №4 Использование компонент для работы с графикой. Создание программы, использующей графические компоненты для иллюстрации данных.

Цель работы: Изучить компоненты для работы с графикой. Изучить методы формирования графических примитивов. Получить навыки создания программ, использующих графические компоненты.

Задание:

Создать и протестировать программу

Варианты заданий:

1. Разработать программу, реализующую игру «Тетрис».
2. Разработать программу, реализующую игру «Ксоникс».
3. Разработать программу, реализующую игру «Арки».

Порядок выполнения:

1. Изучить компоненты для работы с графикой.
2. Изучить методы формирования графических примитивов.
3. Изучить процедуры и функции для изменения графической информации.
4. Изучить работу с таймером (TTimer).
5. Ответить на контрольные вопросы.
6. Получить у преподавателя номер варианта задания.
7. Разработать и отладить программу.

Форма отчетности:

1. Распечатка исходного текста программы с комментариями.
2. Демонстрация работы программы.

Задания для самостоятельной работы:

Изучить методы формирования графических примитивов.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Для рисования 2D графики Windows использует GDI (Graphics Device Interface). GDI обычно не используют для создания особых графических эффектов, для этого есть DirectX, OpenGL, или любые графические библиотеки (такие как: DelphiX, FastLib, DIBUltra, Graphics32).

Однако для создания простых эффектов с минимальными усилиями, GDI вполне подходит. Рассмотрим работу с GDI на примере Turbo Explorer.

С GDI тесно связана ещё одна аббревиатура – DC («Device Context» – контекст устройства). Это то, на чём мы рисуем. В Turbo Explorer контекст устройства представлен как TCanvas. Идея контекста устройства заключается в том, что это универсальное устройство вывода, поэтому можно использовать одинаковые функции как для экрана, так и для принтера. Все графические функции в Turbo Explorer являются надстройками над стандартными GDI функциями Windows.

Эффект перемещения точки начала рисования линии так же достигается при помощи установки свойства PenPos на Canvas.

Например, «Canvas.PenPos.x := 20;», «Canvas.PenPos.y := 50», или «Canvas.PenPos := Point(20,50);». По умолчанию, точка начала рисования установлена в (0,0), то есть, если сразу вызвать «Canvas.LineTo(100,100);» то будет нарисована линия из точки (0,0) в точку (100, 100).

Точка начала рисования автоматически переместится в (100, 100), то есть, если выполнить команду «Canvas.LineTo(200, 100);», то следующая линия будет нарисована из точки (100, 100) в (200, 100).

Поэтому, если мы хотим рисовать линии несоединённые друг с другом, то придётся воспользоваться методом MoveTo. Линия, нарисованная при помощи LineTo, использует текущее перо канваса (типа TPen). Основные свойства пера, это ширина – «Canvas.Pen.Width := 4;» (при помощи которого можно задавать различную ширину линий), и цвет «Canvas.Pen.Color := clLime;».

Рисунок сохраняется до тех пор, пока окно (форма) не было обновлено.

Перерисовка несколько отличается от понятия "рисование". Когда окну необходимо перерисоваться, то Windows посылает определённое сообщение. Это сообщение поступает в обработчик события "OnPaint". Любой код, который поместить в обработчик OnPaint будет вызван каждый раз, когда форме необходимо обновиться.

Основная литература

- 1 Царев Р.Ю. Программные и аппаратные средства информатики: учеб./Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков.-Красноярск: Сиб. федер.ун-т.-2015.-160с.
- 2 Гусева Е.Н. Информатика [электронный ресурс] учебное пособие/ Е.Н. Гусева, И.Ю. Ефимова, К.В. Коробкова, И.Н. Мовчан, Л.А. Савельева.-3- е изд., стереотип.-М.: ФЛИНТА.-2011.-260 с.
- 3 Информатика 1: учебное пособие/ И.Л. Артемов, А.В. Гураков, О.И. Мещерякова, П.С. Мещеряков, Д.С. Шульц.- Томск: ФДО ТУСУР.-2015.-234 с.

Дополнительная литература

- 1 Кремень Е.В. Основы работы в Windows: учебный справочник/Е.В. Кремень, Ю.А. Кремень.-Минск: ТетраСистемс, 2011.- 176 с. – (Электронный гражданин)
- 2 Абрамова Л.В. Инструментальные средства информационных систем: учебное пособие/Л.В. Абрамова; Сев. (Арктич.) федер. ун-т им. М.В. Ломоносова.-Архангельск: ИПЦ СА-ФУ.-2013.-118 с.

Контрольные вопросы для самопроверки

1. Перечислите компоненты, используемые для работы с графикой.
2. Что такое графические примитивы?
3. Какие функции реализуют работу с графическими примитивами?
4. Какие графические форматы файлов можно использовать в программе для вывода заранее сформированных и сохраненных изображений?
5. Для чего нужен в программе компонент «таймер»?

Лабораторная работа №5 Использование диаграмм и бизнес-графики.

Цель работы: Изучить компоненты для работы с диаграммами. Изучить способы отображения данных, виды диаграмм. Получить навыки создания программ для презентаций и демонстраций данных.

Задание:

Создать и протестировать программу

Варианты заданий:

1. Демонстрация изменения температуры воздуха.
2. График биржевой активности.
3. Демонстрация изменения активности пользователя (использование клавиатуры).
4. График изменения количества покупателей в супермаркете.
5. Демонстрация изменения активности пользователя (использование мыши).

Порядок выполнения:

1. Изучить компоненты для работы с диаграммами.
2. Изучить виды диаграмм.
3. Ответить на контрольные вопросы.
4. Получить у преподавателя номер варианта задания.
5. Разработать и отладить программу.

Форма отчетности:

1. Распечатка исходного текста программы с комментариями.

2. Демонстрация работы программы.

Задания для самостоятельной работы:

Изучить компоненты для работы с диаграммами.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Диаграмма – это графическое представление числовых данных, позволяющее быстро оценить соотношение нескольких величин. Представляет собой двухмерное геометрическое символическое изображение информации, с применением различных приёмов техники визуализации. Иногда, для оформления диаграмм используется трёхмерная визуализация, спроецированная на плоскость, что придаёт диаграмме отличительные черты или позволяет иметь общее представление области, в которой она применяется.

Например: финансовая диаграмма, связанная с денежными суммами, может представлять собой количество купюр в пачке или монет в стопке; диаграмма сравнения количества подвижного состава – различную длину изображённых поездов и т. д.

В различных процессорах графопостроения (графических редакторах и электронных таблицах, при изменении данных, на основе которых построена диаграмма, она будет автоматически перестроена с учётом внесённых изменений в таблицу исходных данных диаграммы.

Это позволяет быстро сравнивать различные показатели, статистические данные и т. д. – можно вводить новые данные и сразу видеть изменения диаграммы.

Преимущества диаграмм Преимущество диаграмм перед другими типами наглядной статистической информации заключается в том, что они позволяют быстро произвести логический вывод из большого количества полученных данных.

Результаты расчётов, выполненных с помощью систем статистических вычислений, заносятся в таблицы. Они являются основой для последующего анализа или для подготовки статистического отчёта.

Сами по себе цифры в этих таблицах не являются достаточно наглядными, а если их много, не производят достаточного впечатления. Кроме того, графическое изображение позволяет осуществить контроль достоверности полученных данных, так как на графике достаточно ярко проявляются возможные неточности, которые могут быть связаны с ошибками на каком-либо этапе проведения исследования.

В основном, все статистические пакеты, позволяют графически предоставить полученную числовую информацию в виде различных диаграмм, а затем, если это необходимо, перенести их в текстовый редактор для сборки окончательного варианта статистического отчёта .

Основные типы диаграмм Диаграммы в основном состоят из геометрических объектов (точек, линий, фигур различной формы и различных цветов) и вспомогательных элементов (осей координат, условных обозначений, заголовков и т. п.). Также диаграммы делятся на плоскостные или двухмерные, и пространственные (трёхмерные или объёмные).

Сравнение и сопоставление геометрических объектов на диаграммах может происходить по различным измерениям: по площади фигуры или её высоте, по местонахождению точек, по их густоте, по интенсивности цвета и т. д.

Кроме того, данные могут быть представлены в прямоугольной или полярной системе координат.

Пример использования в Turbo Explorer Для отображения диаграмм предназначены компоненты TChart (данные – Tseries или TLineSeries) и TDBChart (используют данные из таблицы или запроса базы данных). Данные в series добавляются с помощью процедуры AddXY(x,y)

Основная литература

1 Царев Р.Ю. Программные и аппаратные средства информатики: учеб./Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков.-Красноярск: Сиб. федер.ун-т.-2015.-160с.

2 Гусева Е.Н. Информатика [электронный ресурс] учебное пособие/ Е.Н. Гусева, И.Ю. Ефимова, К.В. Коробкова, И.Н. Мовчан, Л.А. Савельева.-3- е изд., стереотип.-М.: ФЛИНТА.-2011.-260 с.

3 Информатика 1: учебное пособие/ И.Л. Артемов, А.В. Гураков, О.И. Мещерякова, П.С. Мещеряков, Д.С. Шульц.- Томск: ФДО ТУСУР.-2015.-234 с.

Дополнительная литература

- 1 Кремень Е.В. Основы работы в Windows: учебный справочник/Е.В. Кремень, Ю.А. Кремень.-Минск: ТетраСистемс, 2011.- 176 с. – (Электронный гражданин)
- 2 Абрамова Л.В. Инструментальные средства информационных систем: учебное пособие/Л.В. Абрамова; Сев. (Арктич.) федер. ун-т им. М.В. Ломоносова.-Архангельск: ИПЦ СА-ФУ.-2013.-118 с.

Контрольные вопросы для самопроверки

1. Перечислите компоненты, используемые для работы с диаграммами.
2. Перечислите основные виды диаграмм.
3. Как в вашей программе обеспечивается динамическое изменение диаграмм при изменении исходных данных?

Лабораторная работа №6 Использование динамических библиотек.

Цель работы: Изучить назначение и методы использования динамических библиотек.

Получить навыки создания и использования динамических библиотек

Задание:

Создать и протестировать.

Варианты заданий:

1. Динамическая графическая заставка на весь экран.
2. Диаграмма по передаваемому набору значений.
3. Обновление указанного файла с сайта в интернет.
4. Помещение передаваемых данных в базу данных.
5. Проверка имени компьютера.
6. Проверка номера жесткого диска.
7. Вывод текстовой информации по запросу кодового слова.

Порядок выполнения:

1. Изучить методы создания динамических библиотек.
2. Изучить методы использования функций из состава динамических библиотек.
3. Ответить на контрольные вопросы.
4. Получить у преподавателя номер варианта задания.
5. Разработать и отладить программу.

Форма отчетности:

1. Распечатка исходного текста программы с комментариями.
2. Демонстрация работы программы.

Задания для самостоятельной работы:

Изучить методы использования функций из состава динамических библиотек.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Первоначально предполагалось, что введение DLL позволит эффективно организовать память и дисковое пространство, используя только один экземпляр библиотечного модуля для различных приложений. Это было особенно важно для ранних версий Microsoft Windows с жёсткими ограничениями по памяти.

Далее, предполагалось улучшить эффективность разработок и использования системных средств за счёт модульности. Замена DLL-программ с одной версии на другую должна была позволить независимо наращивать систему, не затрагивая приложений.

Кроме того, библиотеки DLL могли использоваться разнотипными приложениями – например, Microsoft Office, Microsoft Visual Studio и т. п. 67 Фактически, полных преимуществ от внедрения DLL получить не удалось по причине явления, называемого DLL hell («ад DLL»). DLL hell возникает, когда несколько приложений требуют одновременно различные, не полностью совместимые, версии DLL-библиотек, что приводит к сбоям в этих приложениях.

Когда система выросла до определённых размеров, количество DLL стало превышать многие тысячи, не все из них обладали полной надёжностью и совместимостью, и конфликты типа DLL hell стали возникать очень часто, резко понижая общую надёжность

системы. Поздние версии Microsoft Windows стали разрешать параллельное использование разных версий DLL, что свело на нет преимущества изначального принципа.

Примеры использования DLL в Turbo Explorer

Экспорт в DLL

```
library Example;
{Производим некие действия и возвращаем результат}
function AddNumbers(a, b: Double): Double; cdecl;
begin
  AddNumbers := a + b
end;
{Эта функция экспортируется}
exports
  AddNumbers;
{Инициализатор DLL, аналог функции main() в языках C и C++}
begin
end.
```

Подгрузка и импорт из DLL во время работы

```
program Example;
{$APPTYPE CONSOLE}
var result: Double;
    LibHandle: Hmodule;
    AddNumbers: function(a, b: Double): Double;
begin
  LibHandle := LoadLibrary('Example.dll'); // загружаем DLL
  if LibHandle <> 0 then
    begin
      @AddNumbers := GetProcAddress(LibHandle, 'AddNumbers');
    // запоминаем адрес необходимой функции
      if @AddNumbers <> nil then
        begin
          result := AddNumbers(1, 2);    // используем функцию
          Writeln('The result was: ', result);
        end;
      FreeLibrary(LibHandle);           // выгружаем DLL
    end;
end.
```

Основная литература

1 Царев Р.Ю. Программные и аппаратные средства информатики: учеб./Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков.-Красноярск: Сиб. федер.ун-т.-2015.-160с.

2 Гусева Е.Н. Информатика [электронный ресурс] учебное пособие/ Е.Н. Гусева, И.Ю. Ефимова, К.В. Коробкова, И.Н. Мовчан, Л.А. Савельева.-3- е изд., стереотип.-М.: ФЛИНТА.-2011.-260 с.

3 Информатика 1: учебное пособие/ И.Л. Артемов, А.В. Гураков, О.И. Мещерякова, П.С. Мещеряков, Д.С. Шульц.- Томск: ФДО ТУСУР.-2015.-234 с.

Дополнительная литература

1 Кремень Е.В. Основы работы в Windows: учебный справочник/Е.В. Кремень, Ю.А. Кремень.-Минск: ТетраСистемс, 2011.- 176 с. – (Электронный гражданин)

2 Абрамова Л.В. Инструментальные средства информационных систем: учебное пособие/Л.В. Абрамова; Сев. (Арктич.) федер. ун-т им. М.В. Ломоносова.-Архангельск: ИПЦ СА-ФУ.-2013.-118 с.

Контрольные вопросы для самопроверки

1. Что такое динамическая библиотека?
2. Перечислите цели создания и использования динамических библиотек.
3. Какие сложности могут возникнуть при использовании нескольких однотипных библиотек разных версий?
4. Как в вашей программе обеспечивается использование функций из состава динамической библиотеки?

Лабораторная работа №7 Создание многопоточных приложений для параллельной обработки информации.

Цель работы: Изучить назначение и особенности многопоточных приложений. Получить навыки создания и использования программ для многопоточной обработки данных.

Задание:

Создать и протестировать.

1. Фигуры, перемещающиеся в окне приложения.
2. Многопоточный поиск файлов на дисках.
3. Подбор пароля к файлу *.rar.
4. Подбор пароля к файлу *.zip.

Порядок выполнения:

1. Изучить назначение и особенности многопоточных приложений.
2. Изучить методы использования потоков при создании программ.
3. Ответить на контрольные вопросы.
4. Получить у преподавателя номер варианта задания.
5. Разработать и отладить программу.

Форма отчетности:

1. Распечатка исходного текста программы с комментариями.
2. Демонстрация работы программы.

Задания для самостоятельной работы:

Изучить методы использования потоков при создании программ.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Взаимодействие потоков В многопоточной среде часто возникают проблемы, связанные с использованием параллельно исполняемыми потоками одних и тех же данных или устройств.

Для решения подобных проблем используются такие методы взаимодействия потоков, как взаимное исключение (мьютексы), семафоры, критические секции и события. Взаимное исключение (mutex, мьютекс) – это объект синхронизации, который устанавливается в особое сигнальное состояние, когда не занят каким-либо потоком. Только один поток владеет этим объектом в любой момент времени, отсюда и название таких объектов (от английского mutually exclusive access – взаимно исключаящий доступ) – одновременный доступ к общему ресурсу исключается.

После всех необходимых действий мьютекс освобождается, предоставляя другим потокам доступ к общему ресурсу. Семафоры представляют собой доступные ресурсы, которые могут быть приобретены несколькими потоками в одно и то же время, пока пул ресурсов не опустеет. Тогда дополнительные потоки должны ждать, пока требуется количество ресурсов

не будет снова доступно. Семафоры очень эффективны, поскольку они позволяют одновременный доступ к ресурсам.

События. Объект, хранящий в себе 1 бит информации «просигнализован или нет», над которым определены операции «просигнализировать», «сбросить в непросигнализованное состояние» и «ожидать». Ожидание на просигнализованном событии есть отсутствие операции с немедленным продолжением исполнения потока. Ожидание на непросигнализованном событии приводит к приостановке исполнения потока до тех пор, пока другой поток (или же вторая фаза обработчика прерывания в ядре ОС) не просигнализует событие.

Возможно ожидание нескольких событий в режимах «любого» или «всех». Возможно также создания события, автоматически сбрасываемого в непросигнализованное состояние после пробуждения первого же – и единственного – ожидающего потока (такой объект используется как основа для реализации объекта «критическая секция»). Активно используются в MS Windows, как в режиме пользователя, так и в режиме ядра. Критические секции обеспечивают синхронизацию подобно мьютексам за исключением того, что объекты, представляющие критические секции, доступны в пределах одного процесса.

События, мьютексы и семафоры также можно использовать в однопроцессном приложении, однако, реализации критических секций в некоторых ОС (например, Windows NT), обеспечивают более быстрый и более эффективный механизм взаимно-исключающей синхронизации – операции «получить» и «освободить» на критической секции оптимизированы для случая единственного потока (отсутствия конкуренции) с целью избежать любых ведущих в ядро ОС системных вызовов.

Подобно мьютексам объект, представляющий критическую секцию, может использоваться только одним потоком в данный момент времени, что делает их крайне полезными при разграничении доступа к общим ресурсам. Условные переменные (condvars). Сходны с событиями, но не являются объектами, занимающими память, – используется только адрес переменной, понятие «содержимое переменной» не существует, в качестве условной переменной может использоваться адрес произвольного объекта. В отличие от событий, установка условной переменной в просигнализованное состояние не влечет за собой никаких последствий в случае, если на данный момент нет потоков, ожидающих на переменной.

Установка события в аналогичном случае влечет за собой запоминание состояния «просигнализовано» внутри самого события, после чего следующие потоки, желающие ожидать события, продолжают исполнение немедленно без остановки.

Для полноценного использования такого объекта необходима также операция «освободить mutex и ожидать условную переменную атомарно». Активно используются в UNIX-подобных ОС. Дискуссии о преимуществах и недостатках событий и условных переменных являются заметной частью дискуссий о преимуществах и недостатках Windows и UNIX. Порт завершения ввода-вывода (IO completion port, IOCP).

Реализованный в ядре ОС и доступный через системные вызовы объект «очередь» с операциями «поместить структуру в хвост очереди» и «взять следующую структуру с головы очереди» – последний вызов приостанавливает исполнение потока в случае, если очередь пуста, и до тех пор, пока другой поток не осуществит вызов «поместить». Главнейшей особенностью IOCP является то, что структуры в него могут помещаться не только явным системным вызовом из режима пользователя, но и неявно внутри ядра ОС как результат завершения асинхронной операции ввода-вывода на одной из дескрипторов файлов. Д

ля достижения такого эффекта необходимо использовать системный вызов «связать дескриптор файла с IOCP». В этом случае помещенная в очередь структура содержит в себе код ошибки операции ввода-вывода, а также, для случая успеха этой операции – число реально введенных или выведенных байт. Реализация порта завершения также ограничивает число потоков, исполняющихся на одном процессоре/ядре после получения структуры из очереди. Объект специфичен для MS Windows и позволяет осуществлять обработку входящих запросов соединения и порций данных в серверном программном обеспечении в архитектуре, где число потоков может быть меньше числа клиентов (нет требования создавать отдельный поток с расходами ресурсов на него для каждого нового клиента).

Основная литература

1 Царев Р.Ю. Программные и аппаратные средства информатики: учеб./Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков.-Красноярск: Сиб. федер.ун-т.-2015.-160с.

2 Гусева Е.Н. Информатика [электронный ресурс] учебное пособие/ Е.Н. Гусева, И.Ю. Ефимова, К.В. Коробкова, И.Н. Мовчан, Л.А. Савельева.-3- е изд., стереотип.-М.: ФЛИНТА.-2011.-260 с.

3 Информатика 1: учебное пособие/ И.Л. Артемов, А.В. Гураков, О.И. Мещерякова, П.С. Мещеряков, Д.С. Шульц.- Томск: ФДО ТУСУР.-2015.-234 с.

Дополнительная литература

1 Кремень Е.В. Основы работы в Windows: учебный справочник/Е.В. Кремень, Ю.А. Кремень.-Минск: ТетраСистемс, 2011.- 176 с. – (Электронный гражданин)

2 Абрамова Л.В. Инструментальные средства информационных систем: учебное пособие/Л.В. Абрамова; Сев. (Арктич.) федер. ун-т им. М.В. Ломоносова.-Архангельск: ИПЦ СА-ФУ.-2013.-118 с.

Контрольные вопросы для самопроверки

1. Что такое многопоточное приложение?
2. Перечислите плюсы использования многопоточности в программе.
3. Как взаимодействуют потоки в программе?
4. Каким образом в вашей программе обеспечивается использование многопоточности?

Лабораторная работа №8 Архивация данных по расписанию.

Цель работы: Изучить методы архивации данных, создания иконки приложения в трее.

Задание:

Создать программу, которая автоматически, с заданным пользователем интервалом, копирует указанный файл в архив zip.

Предыдущая копия архива удаляется. При запуске, не должно отображаться окно программы, лишь иконка программы в трее. При клике правой кнопкой мыши на иконке программы, должно отображаться меню, с возможностью выбора интервала архивирования.

Порядок выполнения:

1. Изучить методы работы с zip архивами.
2. Изучить компоненты и методы работы с иконкой приложения в трее.
3. Получить у преподавателя номер варианта задания.
4. Разработать и отладить программу.

Форма отчетности:

1. Распечатка исходного текста программы с комментариями.
2. Демонстрация работы программы.

Задания для самостоятельной работы:

Изучить компоненты и методы работы с иконкой приложения в трее.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Архивация данных – алгоритмическое преобразование данных, при этом водимое с целью уменьшения их объёма. Применяется для более рационального использования устройств хранения и передачи данных. Может быть использована для создания и хранения резервных копий файлов. Обратная процедура называется восстановлением данных (разархивацией). В современных средствах разработки ПО используются различные способы работы с архивами.

Существует значительное количество платных и бесплатных компонентов для работы с различными типами архивов. Компоненты устанавливаются на стандартную панель инструментов и позволяют использовать как базовые функции архивации, так и расширенные возможности.

К базовым функциям относятся: - копирование файлов в архив (архивация), - восстановление файлов из архива (разархивация), - установка степени сжатия файлов в архиве, - защита архива паролем, - создание самораспаковывающихся архивов. Кроме того, можно использовать подключающиеся библиотеки-dll, которые также позволяют работать с архивами. Наконец, можно включить в комплект программы бесплатную копию архиватора.

Как правило, все они позволяют работать с архивами с помощью параметров командной строки. Самым известным видом архивов является zip. В настоящее время этот формат считается стандартом для многих приложений, включающих функции сжатия, резервного копирования и обмена данными.

Большинство библиотек компонентов для работы с архивами используют именно этот стандарт сжатия. создание значка приложения в трее и методы его использования Для Turbo Explorer было написано немало компонентов для помещения иконки в Tray, но существует и собственный компонент TTrayIcon.

Основные операции по настройке компонента можно выполнить за считанные секунды. Стандартные функции компонента позволяют пользователю сворачивать и восстанавливать форму. Это хороший способ держать нужную программу всегда «под рукой», а не закрывать её. Для того, что бы скрыть всё приложение, достаточно применить метод Hide ко всем его формам.

Это автоматически уберёт значки приложения, как с панели переключения задач, так и с системного лотка. Дополнительно можно вызвать метод Application.Minimize, но следует отметить, что он приведет к анимированному сворачиванию формы в панель задач, а не в tray.

Важно знать, что начинать скрывать нужно с верхней (последней открытой) формы заканчивая самой первой. Это будет гарантировать, что восстановление форм произойдет в обратном порядке. Особенно этому нужно уделять внимание при работе с модальными формами и диалогами. Для того, что бы быть уверенным в правильном Z-порядке, следует использовать один простой способ.

Скрывая приложение, перебирать массив Screen.Forms от 0 до Screen.FormCount – 1, а, восстанавливая, наоборот.

Основная литература

- 1 Царев Р.Ю. Программные и аппаратные средства информатики: учеб./Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков.-Красноярск: Сиб. федер.ун-т.-2015.-160с.
- 2 Гусева Е.Н. Информатика [электронный ресурс] учебное пособие/ Е.Н. Гусева, И.Ю. Ефимова, К.В. Коробкова, И.Н. Мовчан, Л.А. Савельева.-3- е изд., стереотип.-М.: ФЛИНТА.-2011.-260 с.
- 3 Информатика 1: учебное пособие/ И.Л. Артемов, А.В. Гураков, О.И. Мещерякова, П.С. Мещеряков, Д.С. Шульц.- Томск: ФДО ТУСУР.-2015.-234 с.

Дополнительная литература

- 1 Кремень Е.В. Основы работы в Windows: учебный справочник/Е.В. Кремень, Ю.А. Кремень.-Минск: ТетраСистемс, 2011.- 176 с. – (Электронный гражданин)
- 2 Абрамова Л.В. Инструментальные средства информационных систем: учебное пособие/Л.В. Абрамова; Сев. (Арктич.) федер. ун-т им. М.В. Ломоносова.-Архангельск: ИПЦ САФУ.-2013.-118 с.

Контрольные вопросы для самопроверки

1. Перечислите основные методы работы с zip архивами.
2. Какие компоненты используются для создания значка программы в трее, какие свойства они поддерживают?
3. Какими способами можно создать приложение, которое при запуске не отображает оконную форму?

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Microsoft Imagine Premium,
ОС Windows 7 Professional,
Microsoft Office 2007 Russian Academic OPEN No Level,
Kaspersky Security.

**11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ
ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ**

| <i>Вид занятия (Лк, ЛР кр, СР...)</i> | <i>Наименование аудитории</i> | <i>Перечень основного оборудования</i> | <i>№ Лек, ЛР</i> |
|---|---|--|--------------------------------|
| 1 | 3 | 4 | 5 |
| Лк | Лаборатория технических средств защиты информации | интерактивная доска Smart Board X885ix со встроенным проектором UX60 | Лк № 1.1.-1.4 Лк № 2.1.-2.6 |
| ЛР | Лаборатория технических средств защиты информации | 16 ПК i5-2500/Н67/4Gb/500Gb (монитор TFT19 Samsung E1920NR); интерактивная доска Smart Board X885ix со встроенным проектором UX60 | ЛР№ 1-3 ЛР№ 4-8 |
| кр | Лаборатория технических средств защиты информации | 16 ПК i5-2500/Н67/4Gb/500Gb (монитор TFT19 Samsung E1920NR); | кр |
| СР | Ч31 | 10-ПК i5-2500/Н67/4Gb (монитор TFT19 Samsung); принтер HP LaserJet P2055D | - |

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

1. Описание фонда оценочных средств (паспорт)

| № компетенции | Элемент компетенции | Раздел | Тема | ФОС |
|---|--|---|--|---------------------------|
| ОПК-3 | способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям | 1. Системное программное обеспечение. | 1.1. Основные понятия. Определение операционной системы. | кр, экзаменационный билет |
| | | 2. Прикладное программное обеспечение и системы. | 2.1. Понятия технологии программирования. | кр, экзаменационный билет |
| ОПК-4 | способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности | 1. Системное программное обеспечение. | 1.2. Управление задачами и памятью. | кр, экзаменационный билет |
| | | | 1.3. Управление вводом/выводом и файловые системы. | кр, экзаменационный билет |
| | | 2. Прикладное программное обеспечение и системы. | 2.2. Требования и спецификации программы. | кр, экзаменационный билет |
| | | | 2.3. Структура программы. | кр, экзаменационный билет |
| | | | 2.4. Испытания программы. | кр, экзаменационный билет |
| 2.5. Внедрение и сопровождение программы. | кр, экзаменационный билет | | | |
| ПК-2 | способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных | 1. Системное программное обеспечение. | 1.4. Архитектура ОС и интерфейсы прикладного программирования. | кр, экзаменационный билет |
| | | | 2.6. Современные методы и средства проектирования программ. | кр, экзаменационный билет |

| | | | | |
|--|---|--|--|--|
| | сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям | | | |
|--|---|--|--|--|

2. Экзаменационные вопросы

| № п/п | Компетенции | | ЭКЗАМЕНАЦИОННЫЕ ВОПРОСЫ | № и наименование раздела | |
|-------|-------------|--|---|---------------------------------------|--|
| | Код | Определение | | | |
| 1 | 2 | 3 | 4 | 5 | |
| 1. | ОПК-3 | способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям | 1. Планирование и диспетчеризация процессов и задач: стратегии планирования, дисциплины диспетчеризации, диспетчеризация задач с использованием динамических приоритетов. | 1. Системное программное обеспечение. | |
| | | | 2. Память и отображения, виртуальное адресное пространство. | | |
| | | | 3. Распределение памяти разделами, сегментная, страничная и сегментно-страничная организация памяти. Распределение оперативной памяти в современных ОС. | | |
| | | | 4. Интерфейс прикладного программирования.. | | |
| | | | 5. Модели жизненного цикла программных систем. Стадии и этапы разработки программных систем и программной документации. | | 2. Прикладное программное обеспечение и системы. |
| | | | 6. Понятия требований к программе. Функциональные и нефункциональные требования. Стандартизация описания требований. | | |
| | | | 7. Техническое задание на разработку программы. Понятие спецификаций программы Проектирование логики модуля. Документирование модуля. | | |
| 2. | ОПК-4 | способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности | 9. Прерывания. Основные виды ресурсов. Классификация операционных систем: однозадачные мониторы, пакетные мониторы, мультипрограммные пакетные ОС, диалоговые многопользовательские ОС, ОС реального времени. | 1. Системное программное обеспечение. | |
| | | | 10. Основные понятия и концепции ввода/вывода в ОС. Основные системные таблицы ввода/вывода. | | |
| | | | 11. Основные принципы построения ОС. Принципы построения интерфейсов ОС. | | |

| | | | | |
|----|------|---|---|--|
| | | | <p>12.Программа. Компонент программной системы. Программная система. Программный продукт. Программное изделие.</p> <p>13.Классификация программных систем по сложности. Основные характеристики и критерии качества программ и программных систем.</p> <p>14.Модуль. Внешние спецификации модуля. Способы описания спецификаций.</p> <p>15.Верификация, тестирование и отладка программы. Основные методы верификации программы.</p> <p>16.Тестирование и отладка. Методы тестирования. План тестирования.</p> <p>17.Средства структурного проектирования.</p> <p>18.Унифицированный язык моделирования UML: диаграммы вариантов использования, диаграммы классов, диаграммы состояний, диаграммы деятельности, диаграммы последовательности, диаграммы компонентов.</p> <p>20. Методы и средства структурного анализа и проектирования: диаграммы потоков данных, словарь данных, спецификации процессов, диаграммы «сущность – связь», спецификации управления.</p> | 2. Прикладное программное обеспечение и системы. |
| 3. | ПК-2 | способность понимать, совершенствовать и применять современный математический аппарат | <p>8.Понятие вычислительного процесса и ресурса: диаграмма состояний процесса, реализация понятия последовательного процесса в ОС.</p> | 1. Системное программное обеспечение. |
| | | | <p>19. Задачи сопровождения программы. Расширение возможностей, адаптация и коррекция. Средства и методы сопровождения. Организация сопровождения. Эксплуатационная документация.</p> | 2. Прикладное программное обеспечение и системы. |

3. Описание показателей и критериев оценивания компетенций

| Показатели | Оценка | Критерии |
|--|----------------------------|---|
| <p>Знать (ОПК-3) синтаксис и семантику алгоритмических конструкций языков программирования высокого уровня и СУБД; базовые структуры данных, средства компьютерной графики и основные численные алгоритмы;</p> <p>(ОПК-4) -методы сбора и обработки и хранения информации, причины нарушения компьютерной безопасности, методы разработки программного обеспечения</p> | Отлично | Демонстрирует все показатели на высоком уровне. Обучающийся всесторонне и глубоко владеет знаниями, сложными навыками, способен уверенно ориентироваться в практических ситуациях. Достигнут высокий уровень формирования компетенций. |
| <p>(ПК-2) основные понятия дисциплины, её методы, место и роль в решении научно практических задач с использованием современного математического аппарата;</p> <p>Уметь (ОПК-3) -разрабатывать математические и информационные модели и алгоритмы для решения прикладных задач,</p> <p>(ОПК-4) -использовать ресурсы сети Интернет для разработки программного обеспечения и программной документации с учетом требований информационной безопасности;</p> <p>(ПК-2) применять и совершенствовать современный математический аппарат при решении научно-практических задач прикладной математики и информатики;</p> <p>Владеть (ОПК-3) -навыками работы с системным и прикладным обеспечением для решения задач математического моделирования в предметной области, а также современным программным обеспечением, средствами тестирования, документации ПО.</p> | Хорошо | Демонстрирует более половины показателей на достаточном и высоком уровне. Обучающийся владеет знаниями, проявляет соответствующие навыки в практических ситуациях, но имеют место некоторые неточности в демонстрации освоения материала. Достигнут повышенный уровень формирования компетенции. |
| <p>(ОПК-4) -использовать ресурсы сети Интернет для разработки программного обеспечения и программной документации с учетом требований информационной безопасности;</p> <p>(ПК-2) применять и совершенствовать современный математический аппарат при решении научно-практических задач прикладной математики и информатики;</p> | Удовлетворительно | Демонстрирует основную часть показателей на достаточном уровне. Обучающийся частично проявляет знания и навыки, входящие в состав компетенции. Пытается, стремится проявлять нужные навыки, понимает их необходимость, но у него не всегда получается. Достигнут только базовый уровень формирования компетенции. |
| <p>(ОПК-4) – навыками системного и объектно- ориентированного программирования для решения стандартных прикладных задач в профессиональной деятельности; базовыми знаниями по защите информации на рабочем месте, в корпоративных сетях при входе в глобальные сети.</p> <p>(ПК-2) - инструментарием для решения математических задач в области прикладной математики и информатики.</p> | Неудовлетворительно | Демонстрирует большинство показателей на недостаточном и крайне низком уровне. Обучающийся не владеет необходимыми знаниями и навыками и не старается их применять. Не достигнут базовый уровень формирования компетенции. |

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности

Место курса «Системное и прикладное программное обеспечение» в системе подготовки бакалавра определяется тем значением которое занимают ее методы в проектировании высоконадежных вычислительных устройств. Изучение дисциплины базируется на знаниях и умениях, приобретенных обучающимися в результате освоения ими следующих дисциплин: математика, дискретная математика, информатика и программирование, теория вероятностей и математическая статистика, вычислительные системы, сети и телекоммуникации. Учебная дисциплина «Теория информации и кодирования» формирует базовый уровень знаний по специальности и является основой для изучения следующих курсов: информационная безопасность, проектирование информационных систем, корпоративные информационные системы и сети.

Изучение дисциплины предусматривает:

- лекции,
- лабораторные работы;
- контрольную работу;
- самостоятельную работу обучающегося
- экзамен.

Овладение ключевыми понятиями является основой усвоения учебного материала по дисциплине.

В процессе проведения лабораторных работ происходит закрепление знаний, формирование умений и навыков.

В процессе консультации с преподавателем обучающийся выясняет наличие пробелов в знаниях и способах решения разных ситуаций.

Работа с литературой является важнейшим элементом в получении знаний по дисциплине. Прежде всего, необходимо воспользоваться списком рекомендуемой по данной дисциплине литературой. Дополнительные сведения по изучаемым темам можно найти в периодической печати и Интернете.

АННОТАЦИЯ

рабочей программы дисциплины

Системное и прикладное программное обеспечение

1. Цель и задачи дисциплины

Целью изучения дисциплины является: Знакомство обучающихся с основными современными системными и прикладными программными продуктами, тенденциями их развития, обучить обучающихся принципам работы в различных программных пакетах, проведению анализа полученных результатов, применению современных программ в профессиональной деятельности.

Задачи дисциплины

- раскрыть значение дисциплины в общем и профессиональном образовании человека;
- приобрести навыки работы с основным программным обеспечением;
- сформировать умения и навыки самостоятельного решения прикладных задач с помощью современного прикладного и программного обеспечения.

2. Структура дисциплины

2.1 Распределение трудоемкости по отдельным видам учебных занятий, включая самостоятельную работу: Лк.-18 час., ЛР-54 час.; СР-90 час.

Общая трудоемкость дисциплины составляет 216 часов, 4 зачетные единицы.

2.2 Основные разделы дисциплины:

1. Системное программное обеспечение.
2. Прикладное программное обеспечение и системы.

3. Планируемые результаты обучения (перечень компетенций)

Процесс изучения дисциплины направлен на формирование следующих компетенций:

ОПК-3 - способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям.

ОПК-4 - способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.

ПК-2 - способность понимать, совершенствовать и применять современный математический аппарат.

4. Виды промежуточной аттестации: экзамен.

*Протокол о дополнениях и изменениях в рабочей программе
на 20__-20__ учебный год*

1. В рабочую программу по дисциплине вносятся следующие дополнения:

2. В рабочую программу по дисциплине вносятся следующие изменения:

Протокол заседания кафедры № _____ от «___» _____ 20__ г.,
(разработчик)

Заведующий кафедрой _____
(подпись)

(Ф.И.О.)

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ТЕКУЩЕГО
КОНТРОЛЯ УСПЕВАЕМОСТИ ПО ДИСЦИПЛИНЕ**

1. Описание фонда оценочных средств (паспорт)

| № компетенции | Элемент компетенции | Раздел | Тема | ФОС |
|----------------------|--|--|--|---|
| ОПК-3 | способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям | 1. Системное программное обеспечение. | 1.1. Основные понятия. Определение операционной системы. | ЛР, кр, тест |
| | | 2. Прикладное программное обеспечение и системы. | 2.1. Понятия технологии программирования. | ЛР, кр, тест |
| ОПК-4 | способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности | 1. Системное программное обеспечение. | 1.2. Управление задачами и памятью. | ЛР, кр, тест |
| | | | 1.3. Управление вводом/выводом и файловые системы. | ЛР, кр, тест |
| | | 2. Прикладное программное обеспечение и системы. | 2.2. Требования и спецификации программы. | ЛР, кр, тест |
| | | | 2.3. Структура программы. | ЛР, кр, тест |
| | | | 2.4. Испытания программы. | ЛР, кр, тест |
| | | | 2.5. Внедрение и сопровождение программы. | ЛР, кр, тест |
| ПК-2 | способность понимать, совершенствовать и применять современный математический аппарат | 1. Системное программное обеспечение. | 1.4. Архитектура ОС и интерфейсы прикладного программирования. | ЛР, кр, тест |
| | | | 2. Прикладное программное обеспечение и системы. | 2.6. Современные методы и средства проектирования программ. |

2. Описание показателей и критериев оценивания компетенций

Так как текущий контроль проводится в форме тестирования и предназначен для проверки знаний самими обучающимися, тест может быть зачтен или не зачтен. В дальнейшем обучающиеся могут повторить попытки выполнить тест по той теме, где были обнаружены пробелы в их знаниях.

| Показатели | Оценка | Критерии |
|--|----------------------------|--|
| <p>Знать (ОПК-3) синтаксис и семантику алгоритмических конструкций языков программирования высокого уровня и СУБД; базовые структуры данных, средства компьютерной графики и основные численные алгоритмы; (ОПК-4) -методы сбора и обработки и хранения информации, причины нарушения компьютерной безопасности, методы разработки программного обеспечения (ПК-2)</p> | Отлично | Демонстрирует все показатели на высоком уровне. Обучающийся всесторонне и глубоко владеет знаниями, сложными навыками, способен уверенно ориентироваться в практических ситуациях. Достигнут высокий уровень формирования компетенций. |
| <p>основные понятия дисциплины, её методы, место и роль в решении научно практических задач с использованием современного математического аппарата; Уметь (ОПК-3) -разрабатывать математические и информационные модели и алгоритмы для решения прикладных задач, (ОПК-4) -использовать ресурсы сети Интернет для разработки программного обеспечения и программной документации с учетом требований информационной безопасности;</p> | Хорошо | Демонстрирует более половины показателей на достаточном и высоком уровне. Обучающийся владеет знаниями, проявляет соответствующие навыки в практических ситуациях, но имеют место некоторые неточности в демонстрации освоения материала. Достигнут повышенный уровень формирования компетенции. |
| <p>(ПК-2) применять и совершенствовать современный математический аппарат при решении научно-практических задач прикладной математики и информатики; Владеть (ОПК-3) -навыками работы с системным и прикладным обеспечением для решения задач математического моделирования в предметной области, а также современным программным обеспечением, средствами тестирования, документации ПО. (ОПК-4)</p> | Удовлетворительно | Демонстрирует основную часть показателей на достаточном уровне. Обучающийся частично проявляет знания и навыки, входящие в состав компетенции. Пытается, стремится проявлять нужные навыки, понимает их необходимость, но у него не всегда получается. Достигнут только базовый уровень формирования компетенции. |
| <p>навыками системного и объектно- ориентированного программирования для решения стандартных прикладных задач в профессиональной деятельности; базовыми знаниями по защите информации на рабочем месте, в корпоративных сетях при входе в глобальные сети. (ПК-2) - инструментарием для решения математических задач в области прикладной математики и информатики.</p> | Неудовлетворительно | Демонстрирует большинство показателей на недостаточном и крайне низком уровне. Обучающийся не владеет необходимыми знаниями и навыками и не старается их применять. Не достигнут базовый уровень формирования компетенции. |

Фонд тестовых заданий

по дисциплине

Б1.В.04 Системное и прикладное программное обеспечение

ТЕМАТИЧЕСКАЯ СТРУКТУРА ТЕСТОВ

| № раздела | Наименование раздела | № задания | Тема задания |
|-----------|---|----------------|---|
| 1. | Системное программное обеспечение. | 1-4 | Основные понятия. Определение операционной системы. |
| | | 7-8 | Управление задачами и памятью. |
| | | 13-15 | Управление вводом/выводом и файловые системы. |
| | | 10-12 21-23 | Архитектура ОС и интерфейсы прикладного программирования. |
| 2. | Прикладное программное обеспечение и системы. | 5-6 | Понятия технологии программирования. |
| | | 16-19 | Требования и спецификации программы. |
| | | 20 | Структура программы. |
| | | 33-35 | Испытания программы. |
| | | 24-27 | Внедрение и сопровождение программы. |
| | | 28-32 | Современные методы и средства проектирования программ. |

Тестовые задания

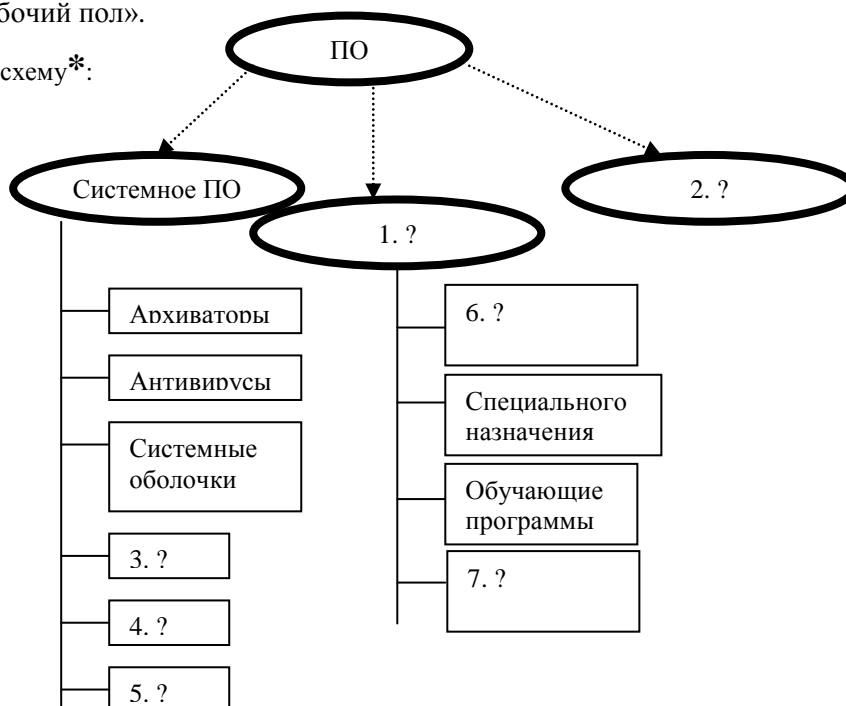
- Программа должна обладать следующими свойствами:
 - упорядоченная последовательность;
 - системность, дискретность;
 - массовость, понятность;
 - однозначность, точность.
- Какие приложения относятся к прикладным программам общего назначения?
 - текстовый редактор;
 - бухгалтерские программы;
 - игры;
 - ОС.
- Системный диск необходим для:
 - загрузки ОС;
 - хранения важных файлов;
 - систематизации файлов;
 - лечение компьютерных вирусов.
- Программное обеспечение (ПО) – это:
 - совокупность программ, позволяющих организовать решение задач на компьютере;
 - возможность обновления программ за счет бюджетных средств;
 - список имеющихся в кабинете программ, заверен администрацией школы;
 - программы по настройке ПК.
- Система программирования – это:
 - процесс составления программ;
 - процесс, облегчающий работу программиста;
 - процесс, обучающий начальным шагам программиста;
 - процесс, обеспечивающий функционирование устройств.
- Прикладное программное обеспечение:
 - программы для обеспечения работы других программ;
 - программы для решения конкретных задач обработки информации;
 - программы, обеспечивающие качество работы печатающих устройств;
 - программы для загрузки компьютера.
- Операционные системы:
 - DOS, Windows, Unix;
 - Word, Excel, Power Point;

- c) Basic, Access, Pascal;
 - d) Winrar, Norton Commander, command.com.
8. Системное программное обеспечение:
- a) программы для организации совместной работы устройств компьютера как единой системы;
 - b) программы для организации удобной системы размещения программ на диске;
 - c) набор программ для работы устройства системного блока компьютера;
 - d) программы для работы периферийных устройств.
9. Что такое утилиты?
- a) это программы для обнаружения компьютерных вирусов;
 - b) это устройства преобразования сигналов;
 - c) это программы – переводчики;
 - d) это программы, обеспечивающие удобства при обслуживании компьютера.
10. Каково назначение программы Текстовый редактор?
- a) создание компьютерных программ;
 - b) создание и редактирование текстовых документов;
 - c) создание и редактирование графических изображений;
 - d) создание компьютерных вирусов.
11. Каково назначение Систем управления базами данных
- a) выполнение расчётов;
 - b) создание и обработка базы данных;
 - c) создание текстовых документов;
 - d) поиск компьютерных вирусов.
12. В каком формате размещается текстовая информация в сети Internet?
- a) текстовые документы *.doc;
 - b) текстовые файлы *.txt;
 - c) файлы баз данных *.mdb;
 - d) web-страницы *.htm.
13. К какому классу программного обеспечения относятся драйверы?
- a) к прикладному;
 - b) к системному;
 - c) к системам программирования;
 - d) к офисному.
14. Программный продукт Windows представляет собой ...
- a) прикладную программу;
 - b) систему программирования;
 - c) только оболочку операционной системы;
 - d) операционную систему с оболочкой;
15. Архивный файл представляет собой:
- a) файл, защищенный от копирования;
 - b) файл, защищенный от несанкционированного доступа;
 - c) файл, сжатый с помощью архиватора;
 - d) файл, зараженный компьютерным вирусом.
16. Назовите «лишнюю» программу:
- a) Excel;
 - b) Access;
 - c) Word;
 - d) Turbo-Pascal.
17. Какое из названных действий можно произвести архивным файлом:
- a) переформатировать;
 - b) распаковать;
 - c) запустить на выполнение;
 - d) отредактировать.
18. Файл Tetris.com находится на диске C в папке Games, которая является подпапкой папки Day. Выбрать полное имя файла.
- a) C:\Tetris.com\Games\Day;

- b) C:\Games\Tetris.com;
- c) C:\Day\Games\Tetris.com;
- d) C:\Games\Day\Tetris.com.

19. После выполнения команды «удалить» в Windows файлы и папки обычно попадают
- a) в буфер обмена;
 - b) на панель задач;
 - c) в «корзину»;
 - d) на «рабочий пол».

20. Дополните схему*:



21. Какие программы входят в системное ПО компьютера?
- a) ОС, Tota Commander, бухгалтерские программы;
 - b) Игры, электронные учебники, антивирусник;
 - c) ОС, Tota Commander, антивирусные программы;
 - d) электронные таблицы, текстовый редактор, презентации.
22. В процессе загрузки операционной системы происходит?
- e) копирование файлов ОС с дискеты на жесткий диск;
 - f) копирование файлов ОС с CD на жесткий диск;
 - g) последовательная загрузка файлов ОС в оперативную память;
 - h) Копирование оперативной памяти на жесткий диск.
23. Операционная система:
- a) система программ, которая обеспечивает совместную работу всех устройств компьютера по обработке информации;
 - b) система математических операций для решения отдельных задач;
 - c) система планового ремонта и технического обслуживания компьютерной техники;
 - d) система взаимодействия всех программ компьютера.
24. Прикладное программное обеспечение – это:
- a) справочное приложение к программам;
 - b) текстовый и графический редакторы, обучающие и тестирующие программы, игры;
 - c) набор игровых программ;
 - d) электронные справочники.
25. Сервисные (обслуживающие) программы:
- a) программы сервисных организаций по бухгалтерскому учету;
 - b) программы обслуживающих организаций по ведению делопроизводства;
 - c) системные оболочки, утилиты, драйвера устройств, антивирусные и сетевые программы;
 - d) текстовые редакторы, базы данных, графические редакторы.

26. Системные оболочки – это:
- специальная кассета для удобного размещения дискет с операционной системой;
 - специальная программа, упрощающая диалог пользователь – компьютер, выполняет команды операционной системы;
 - система приемов и способов работы конкретной программы при загрузке программ и завершении работы;
 - специальная область на диске для размещения ОС.
27. Что такое драйвер?
- это устройство, управляющее работой какого-либо внешнего устройства;
 - это программа, управляющая работой какого-либо внешнего устройства;
 - это программа для тестирования устройств компьютера;
 - это программа для обслуживания магнитных дисков.
28. Какие возможности предоставляют пользователю программы - графические редакторы?
- создание и редактирование графических изображений;
 - создание и редактирование текстовых документов;
 - создание программ – приложений;
 - создание баз данных.
29. Какое основное назначение имеют программы Электронные таблицы
- создание текстовых документов;
 - выполнение расчётов с табличными данными;
 - создание электронного текста;
 - создание графических изображений.
30. В каком формате размещается текстовая информация в сети Internet?
- текстовые документы *.doc;
 - текстовые файлы *.txt;
 - файлы баз данных *.mdb;
 - web-страницы *.htm.
31. К какому классу программного обеспечения относятся компьютерные игры?
- к системам программирования;
 - к системному;
 - к прикладному;
 - к развивающему;
32. Какая из перечисленных программ относится к офисному программному обеспечению
- Paint;
 - Visual Basic;
 - WORD;
 - Windows.
33. Операционные системы входят в состав:
- системы управления базами данных;
 - систем программирования;
 - прикладного программного обеспечения;
 - системного программного обеспечения.
34. Назовите «лишнюю» программу:
- Excel;
 - Access;
 - Word;
 - Turbo-Pascal.
35. Файл Tetris.com находится на диске C в папке Games, которая является подпапкой папки Day. Выбрать полное имя файла.
- C:\Tetris.com\Games\Day;
 - C:\Games\Tetris.com;
 - C:\Day\Games\Tetris.com;
 - C:\Games\Day\Tetris.com.

Программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 01.03.02 Прикладная математика и информатика от «12» марта 2015г. № 228 и

для набора 2015 года: учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «13» июля 2015 г. № 475

для набора 2016 года: учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «06» июня 2016г. № 429

для набора 2017 года учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «6» марта 2017г. № 125

Программу составил (и):

Шашок О.В., доцент кафедры МиФ к.т.н. _____

Медведева О.И., доцент кафедры МиФ, к.т.н. _____

Рабочая программа рассмотрена и утверждена на заседании кафедры МиФ

от «__» _____ 20 __ г., протокол № _____

И.о. заведующего кафедрой МиФ _____ О.И. Медведева

СОГЛАСОВАНО:

И.о. заведующего выпускающей кафедрой МиФ _____ О.И. Медведева

Директор библиотеки _____ Т.Ф. Сотник

Рабочая программа одобрена методической комиссией Естественнонаучного факультета

от «__» _____ 20 __ г., протокол № _____

Председатель методической комиссии факультета _____ М.А. Варданян

СОГЛАСОВАНО:

Начальник
учебно-методического управления _____ Г.П. Нежевец

Регистрационный № _____

(методический отдел)