

Системное и прикладное программное обеспечение

Темы, определяемые образовательным стандартом

- 1) Основные этапы, методы, средства и стандарты разработки программного обеспечения.
- 2) Системы программирования (принципы организации, состав и схема работы).
- 3) Основные типы операционных систем, принципы управления ресурсами в операционной системе.
- 4) Работа с математическими пакетами (MATLAB и др.)

Методическое обеспечение

1) Материалы на сайте НГТУ на личной странице преподавателя.

2) Электронный курс в системе DiSpace на сайте НГТУ.

При противоречиях в методических указаниях приоритет имеют лекции.

План на семестр

- 1) Лабораторные работы.
- 2) Расчетно-графическая работа. Разработка приложения в рамках проекта.
- 3) Теоретические знания (тест).

Лабораторные работы

В рамках курса требуется выполнить 7 работ.

Работа в бригадах по 1–2 человека.

Не менее 2-х работ выполняется под Linux.

Невыполнение этого условия влечёт штрафные баллы.

Одна работа не может выполняться и под Windows, и под Linux.

По двум работам предоставляется отчёт, в т.ч. один отчёт по работе в Матлаб.

Лабораторные работы

Работы находятся в файле SppoLabs2016, который доступен на сайте НГТУ на личной странице преподавателя.

Первые две работы из каждого из трёх разделов являются рекомендованными.

Оставшаяся работа выбирается на усмотрение учащегося из любых работ, в т.ч. из предлагавшихся в прошлые годы (27 работ).

Рекомендуется начать с работ в Matlab.

Работы различаются по максимальному количеству баллов.

Требования к отчету

Отчет сдаётся в печатном виде.

Отчет должен начинаться с заголовка, где указывается тема, исполнители и цель работы.

Основную часть составляет раздел «Ход работы», в котором должны быть описаны проделанные действия с объяснением результатов и обязательным приведением конкретных значений (имен файлов, путей, размеров архивов, номеров кластеров и т. п.). Порядок выполнения шагов может отличаться от рекомендованного в задании.

Вывод может содержать, например, заключение о степени удобства изученного программного продукта и возможной области его применения.

Порядок защиты

Защита – выполнить «задание» и ответить на любой вопрос по теме. Возможны дополнительные задания, аналогичные основным.

Многие задания нужно выполнять не буквально, а ориентируясь на конфигурацию.

Кто сдал больше, имеет приоритет в очередности.

График защиты: по одной работе за каждое занятие, начиная с третьего.

Раздел 1. Инструментарий пользователя

1.1. Командная строка.

1.2. Издательская система TeX.

1.3. Файловые менеджеры. Архивация данных

1.4. Загрузчики. Эмуляторы.

1.5. Программный интерфейс ОС.

Раздел 2. Управление ресурсами.

- 2.1.** Файловая система (FAT: 1,0, ext2fs: 1,1 +0,5).
- 2.2.** Производительность функций чтения и записи файлов.
- 2.3.** Производительность функций управления памятью.

Раздел 3. Программные средства для математических расчётов.

- 3.1.** Матричная арифметика Матлаб.
- 3.2.** Средства визуализации в Матлаб.
- 3.3.** Взаимодействие Матлаб с приложениями
(в т.ч. на C++)
- 3.4.** Математические расчёты в R.

Приветствуется выполнение работ в Octave.

Расчетно-графическая работа

Цель. сформировать практические навыки проектирования и разработки программных продуктов.

Работа выполняется в рамках коллективной или индивидуальной разработки программного продукта по собственному проекту.

Численность бригады не ограничена.

Объем работ: 15–25 часов.

Оценка: от 5 до 20–25 баллов.

Требования к выполнению

Средства реализации должны быть доступны в терминальных классах.

Особая форма — полуреферативная работа: делается реферат по некоторой теме с иллюстрациями программным кодом.

Допускается использование готовых свободно-распространяемых кодов.

Обязательно использование UML.

Этапы выполнения

Этапы работы:

- составление проектной документации,
- написание кода,
- отладка и тестирование,
- инсталляция, составление пользовательской документации.

Данные задачи выполняются индивидуально или распределяются между членами бригады. Задачи достаточно независимы. В частности, набор тестов составляется по спецификации и не требует наличия кода. Руководство пользователя также можно написать на основе лишь проекта, не имея реализации.

Структура отчета

- 1) Аннотация проекта (описание идеи и цели).
- 2) Введение в предметную область (или реферат).
- 3) Спецификация (техническое задание на разработку).
- 4) Архитектура, технические решения (основной раздел).
- 5) Примеры кода с описанием.
- 6) Набор тестов.
- 7) Презентация продукта (описание, что получилось).
- 8) Руководство пользователя.

Отчет может быть представлен один на бригаду или персонально. В реферативном проекте обязательные пункты: 1, 2, 3, 5, 8.

Варианты заданий

Задача для программной реализации придумывается самостоятельно или запрашивается у преподавателя.

Рекомендуется брать задачу вдвое проще, чем представляется реализуемым.

Примеры.

- Интернет-пейджер.
- Анимация.
- Игра (уровня «крестики-нолики»).
- Мобильное приложение.
- Программа в Матлаб.
- На основе лабораторной работы (непрописанной).

Контроль теоретических знаний

Тест на компьютере: программа доступна, но может быть обновлена.

Устный зачёт: два вопроса по материалам лекций.

Альтернатива лабораторным работам

- придумать свои работы,
- оптимизировать существующие,
- другое.. (напр. расширенная РГР),
- работа может засчитываться за несколько.

Балльно–рейтинговая система

Ориентировочный вклад видов контроля

- | | |
|-------------------------|-------------|
| — лабораторные работы | — 50 баллов |
| — РГР | — 25 баллов |
| — теоретические вопросы | — 25 баллов |

Все задания оцениваются дифференцированно в традиционной шкале.

При выполнении всех заданий итоговая оценка примерно равна средней оценке по заданиям.

Классификация ПО

Системное ПО

управляет ресурсами, обеспечивает работу прикладных программ.

Категории:

- ОС (UNIX, DOS/Windows, специализированные),
- драйверы (обеспечивают единообразный интерфейс к устройствам),
- системные утилиты (антивирусы, дисковые, сетевые),
- системные оболочки (Win 3.11), файловые процессоры (проводник, FAR, mc),
- средства разработки (компиляторы, компоновщики, интегрированные среды).

Прикладное

служит потребностям пользователя.

Категории:

- сервисное (широкого использования),
- пользовательские программы (специализированное).