

ПРИМЕРНОЕ СОДЕРЖАНИЕ КУРСОВ (И ЛЕКЦИОННЫЙ МАТЕРИАЛ, И ЛАБОРАТОРНЫЙ ПРАКТИКУМ МОГУТ ИМЕТЬ В ИТОГЕ НЕСКОЛЬКО ИНОЕ СОДЕРЖАНИЕ, НО БЛИЗКОЕ К ПРЕДСТАВЛЕННОМУ)

Для студентов 2-4 курса

1. Язык программирования «С#»

Лекции

1. Введение в технологию .NET. Краткий обзор её языков. Особенности и преимущества С#. Основы языка С#.
2. Объектно-ориентированное программирование на языке С#. Интерфейсы и коллекции.
3. Индексаторы. Перегрузка операторов. Делегаты. События. Рефлексия типов. Атрибуты.
4. Разработка графического интерфейса пользователя. WindowsForms. Обзор элементов управления WindowsForms.
5. Многопоточное программирование на языке С#. Работа с файлами и сериализация объектов.
6. Сборки. Домены приложений. Работа с унаследованным программным кодом.
7. Технология GDI+. Работа с базами данных. ADO.NET.
8. Разработка Web-приложений с использованием ASP.NET. Web-сервисы.
9. Новые возможности языка С# версий 3.0, 3.5 и 4.0. Параллельное программирование на языке С#.

Лабораторные работы (*примерные темы*)

1. Объектно-ориентированное программирование на языке С#.
2. Разработка WinForms-приложений.
3. Чтение и запись данных в текстовые, двоичные и XML-файлы. Модульное тестирование.
4. Доступ к данным с использованием технологии ADO.NET. Разработка Web-сервиса для доступа к данным.
5. Распараллеливание решения матричных задач с использованием технологии TPL.

2. ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Лекции

1. Введение в современные методы проектирования программного обеспечения.
2. Язык UML. Диаграммы прецедентов, классов, взаимодействия, состояний и деятельности.
3. Анализ требований. Виды обеспечения программных систем.
4. Парадигмы проектирования программных систем. Стадии процесса разработки.
5. Модульное (юнит) тестирование. Самодокументируемый код.
6. Рефакторинг. Стратегии оптимизации кода.
7. Шаблоны проектирования. Порождающие шаблоны. Структурные шаблоны. Шаблоны поведения.
8. Шаблоны для организации пользовательского интерфейса. MVC и MVP. Шаблоны для организации доступа к данным.
9. Методы декомпозиции программных систем.

Лабораторные работы (примерные темы)

1. Разработка UML-диаграмм.
2. Юнит-тестирование кода. Технология TDD. Рефакторинг существующего программного кода.
3. Проектирование программного приложения с использованием шаблонов. Организация доступа к данным с использованием шаблонов проектирования.
4. Организация интерфейса пользователя с использованием шаблона MVP.
5. Разработка самодокументируемого кода с использованием XML-комментариев.

Замечание – все примеры программ в данном курсе будут приводиться на языке C# ☺

3. ПРОГРАММИРОВАНИЕ ПРОГРАММНЫХ КОМПЛЕКСОВ

Получаемые знания:

1. Проектирование программных комплексов
2. Обработка ошибок
3. Стандартная библиотека.
4. Внешние библиотеки.
5. Многопоточное программирование.

Получаемые компетенции:

1. Умение решать задачи, требующей создания программ средней сложности.

(В нашем случае много внимания в этом курсе будет уделено технологиям многопоточных вычислений – OpenMP и CUDA, а также – технологии распределенных вычислений MPI, использованию вычислительных библиотек на языке C++)

4. ТЕОРИЯ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ

Получаемые знания:

1. Языки и алгоритмы.
2. Сложность.
3. Сложностные классы
4. Вероятностные сложностные классы.

Получаемые компетенции:

Умение оценивать временную емкость сложной решаемой задачи.

(ТАКЖЕ В ЭТОМ КУРСЕ – АЛГОРИТМИЗАЦИЯ, ЗАДАЧИ НА ГРАФАХ, ОБРАБОТКА СТРОК, КОМБИНАТОРИКА И ПРОЧЕЕ).

Для студентов 5-6 курсов

5. Криптография

Лекции

1. Основы теории защиты информации. Цифровое шифрование.
2. Математический аппарат криптографии. Конечные поля. Дискретный логарифм.
3. Криптография с закрытым ключом. Блочные и поточные шифры.
4. Криптография с открытым ключом.
5. Цифровая подпись.
6. Протоколы распределения ключей и секретов. Доказательства с нулевым разглашением.
7. Программное обеспечение криптосистем. Обзор криптопровайдеров.
8. Аппаратное обеспечение криптосистем.
9. Законодательные аспекты, связанные с криптографией.

Лабораторные работы

1. Решение базовых задач криптографии.
2. Организации шифрования с закрытым ключом.
3. Организации шифрования с открытым ключом.
4. Работа с современными криптопровайдерами.
5. Аппаратное обеспечение криптосистем.

6. УПРАВЛЕНИЕ ПРОЕКТАМИ

Получаемые знания:

1. Жизненный цикл проекта. Связь с жизненным циклом программ.
2. Планирование, управление и мониторинг состояния проекта.
3. Процессы производства программного обеспечения.
4. Управление рисками при производстве программного обеспечения

Получаемые компетенции:

Умение организовывать процесс производства программного обеспечения.