



СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ



«ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ»

Шифр та назва спеціальності	122 – Комп'ютерні науки	Факультет / Інститут	Комп'ютерних наук і програмної інженерії
Назва освітньо-наукової програми	Комп'ютерні науки	Кафедра	Системного аналізу та інформаційно-аналітичних технологій

ВИКЛАДАЧ



Прокопенков Володимир Пилипович, Volodymyr.Prokopenkov@khp.edu.ua

Старший викладач кафедри Системного аналізу та інформаційно-аналітичних технологій НТУ «ХПІ». Досвід роботи – 30 років. Автор 45 наукових та навчально-методичних праць. Провідний лектор з дисципліни: «Технології програмування»

ЗАГАЛЬНА ІНФОРМАЦІЯ ПРО ДИСЦИПЛІНУ

Анотація	Дисципліна спрямована на освоєння сучасних технологій, методів та засобів програмування на основі системи автоматизації програмування Visual Studio.Net
Мета та цілі	Виробити у студентів теоретичні та практичні навички використання сучасних технологій, методів та засобів програмування мовою C# в середовищі Visual Studio.Net для розробки програмних застосунків для операційної системи Windows
Формат	Лекції, лабораторні роботи, консультації. Підсумковий контроль - залік
Результати навчання	Теоретичні та практичні навички розробки програмних застосунків мовою C# в середовищі Visual Studio.Net
Обсяг	Загальний обсяг дисципліни 120 год.: лекції – 32 год., лабораторні роботи – 32 год., самостійна робота – 56 год.
Пререквізити	«Програмне забезпечення обчислюваних систем», «Програмування та алгоритмічні мови», «Архітектура обчислювальних систем», «Об'єктно-орієнтоване програмування», «Алгоритми та структури даних»
Вимоги викладача	Студент зобов'язаний відвідувати заняття згідно розкладу, не спізнюватися. Дотримуватися етики поведінки. Для проходження дисципліни необхідно мати: бажання, можливість, терпіння, сумління та персональний комп'ютер. Працювати з навчальної та додатковою літературою як на електронних носіях так і в Інтернеті. Студент повинен регулярно готуватись до занять.

СТРУКТУРА ДИСЦИПЛІНИ

Лекція 1	Вивчення середовища розробки та виконання застосунків Microsoft Visual Studio .Net та його інструментів.	Лабораторна робота 1	Середовище MS Visual Studio .Net та його інструменти.	Самостійна робота	Дізасемблер проміжного коду ILDASM та його використання. Застосування WinCV.exe і його використання для вивчення типів. Web-застосування ClassViewer.
Лекція 2	Розробка консольного застосунка з використанням процедурної техніки.	Лабораторна робота 2	Розробка консольного застосунка засобами компілятора командної строки		Створення застосунків засобами компілятора командної строки.
Лекція 3	Розробка застосунка обробки даних з використанням вбудованих структурних та типів та типів посилань та управляючих структур.	Лабораторна робота 3	Розробка консольного застосунка засобами середовища MS Visual Studio. NET.		Розробка форми застосунка і використання елементів управління.
Лекція 4	Обробка даних з використанням типу string.	Лабораторна робота 4	Використання типу string.		
Лекція 5	Використання класу Console для введення та виведення даних застосунка на консоль.	Лабораторна робота 5	Введення та виведення даних.		
Лекція 6	Розробка застосунка обробки масивів даних.	Лабораторна робота 6	Обробка масивів даних.		
Лекція 7	Обробка виключних ситуацій при виконанні застосунка.	Лабораторна робота 7	Обробка виключних ситуацій.		
Лекція 8	Розробка застосунка пов'язаного з зчитуванням та записом даних у файли на диску.	Лабораторна робота 8	Зчитування та запис даних у файли на диску.		
Лекція 9	Розробка і використання типів делегат в застосунку C#.	Лабораторна робота 9	Тип делегат в застосунку C#.		
Лекція 10	Розробка застосунків з використанням подій.	Лабораторна робота 10	Використання подій.		
Лекція 11	Розробка класу для вирішення задачі обробки даних з використанням об'єктної технології.	Лабораторна робота 11	Розробка класу для вирішення задачі обробки даних.		Створення MDI – застосунків.
Лекція 12	Розробка ієрархії класів для вирішення задач обробки даних.	Лабораторна робота 12	Розробка ієрархії класів для вирішення задач обробки даних.		
Лекція 13	Розробка контейнерного класу, що наслідує задані інтерфейси.	Лабораторна робота 13	Розробка контейнерного класу.		
Лекція 14	Розробка і використання шаблонних функцій та класів.	Лабораторна робота 14	Розробка шаблонних функцій та класів.		
Лекція 15	Розробка Windows-застосунка.	Лабораторна робота 15	Розробка Windows застосунка.		Створення і виконання другорядних потоків застосунка.
Лекція 16	Розробка застосунка з паралельною обробкою даних.	Лабораторна робота 16	Використання потоків.		

ЛІТЕРАТУРА ТА НАВЧАЛЬНІ МАТЕРІАЛИ

1. Рейли Д. Создание приложений Microsoft ASP.Net/ Пер.с англ.- М.: Изд.-торговый дом Русская редакция, 2002.- 480с. : ил.
2. Петцольд Ч. Программирование для Microsoft Windows на C#. В 2-х томах. Том 1./Пер. с англ. — М.: Издательско-торговый дом Русская Редакция, 2002.- 576 с.: ил.
3. Петцольд Ч. Программирование для Microsoft Windows на C#. В 2-х томах. Том 2./Пер. с англ. — М.: Издательско-торговый дом Русская Редакция, 2002.— 624 с.: ил.
4. Лабор В. В. Си Шарп: Создание приложений для Windows/ В. В. Лабор.— Мн.: Харвест, 2003. -384 с.
5. Рихтер Дж. Программирование на платформе Microsoft .NET Framework /Пер. с англ. — 2-е изд., испр. — М.: Издательско-торговый дом Русская Редакция, 2003.— 512 стр.: ил.
6. Разработка Windows-приложений на Microsoft Visual Basic .NET и Microsoft Visual C# -NET. Учебный курс MCAD/MCSD/Пер. с англ. — М.: Издательско-торговый дом Русская Редакция, 2003. — 512 стр.: ил.
7. Троелсен Э. C# и платформа NET. Библиотека программиста. — Спб.: Питер, 2003.-800с.:ил.
8. Анализ требований и создание архитектуры решений на основе Microsoft .NET. Учебный курс MCSD/Пер. с англ. — М.: Издательско-торговый дом Русская Редакция, 2004.— 416 стр.: ил.
9. Шилдг, Герберт. Полный справочник по C#. : Пер. с англ. — М. : Издательский дом "Вильямс", 2004. — 752 с. : ил.
10. Бишоп Дж. C# в кратком изложении/ Дж. Бишоп, Н.Хорспул; Пер. с англ.=М.: Бином, Лаборатория знаний, 2005.- 472с., ил.
11. Методичні вказівки до лабораторних робіт "Основи програмування на мові C#" для студентів напрямів 6.040302 "Інформатика", 6.040303 "Системний аналіз" з курсу "Технологія програмування" /Упоряд. Ю. Н.Кожин, О. Н.Малих, В. Ф. Прокопенков—Харків: НТУ "ХПІ", 2011. — 122 с.
12. Методичні вказівки до лабораторних робіт "Робота з двійковими файлами в мові C#" з дисципліни "Технологія програмування" для студентів напрямів 6.040302 "Інформатика", 6.040303 "Системний аналіз" /упоряд. Ю. Н. Кожин, О. Н. Малих, В. Ф. Прокопенков—Х.: НТУ "ХПІ", 2016.— 32 с.
13. Методичні вказівки до лабораторних робіт "Обробка виняткових ситуацій у мові C#" з дисципліни "Технологія програмування" для студентів напрямів 6.040302 "Інформатика", 6.040303 "Системний аналіз" /упоряд. Ю. Н. Кожин, О. Н. Малих, В. Ф. Прокопенков—Х.: НТУ "ХПІ", 2013.— 36 с.

1. Науково-технічна бібліотека НТУ «ХПІ» [Електронний ресурс]. Режим доступу: <http://library.kpi.kharkov.ua/>
2. Русский MSDN [Електронний ресурс].-Режим доступу: <https://msdn.microsoft.com/ru-ru>
3. Англомовний MSDN [Електронний ресурс].- Режим доступу: <https://msdn.microsoft.com/en-us/>
4. Язык программирования C# 5.0 и платформа .NET 4.5, Эндрю Троелсен, 6-е издание, 1312 стр., ISBN 978-5-8459-1814-7, «ВИЛЬЯМС», март 2013
5. C# 5.0. Справочник. Полное описание языка, Джозеф Албахари, Бен Албахари, 5-е издание, 1054 стр., ISBN 978-5-8459-1819-2, «ВИЛЬЯМС», 2013

ПЕРЕЛІК ЗАПИТАНЬ ДЛЯ ПІДГОТОВКИ ДО ЗАЛІКУ

Порівняння засобів програмування для Windows. Концепція швидкої розробки програм. Компоненти, властивості, засоби і події. Переваги .Net над існуючими засобами програмування під Windows. Побудові блоки .Net (CLR, CTS та CLS, бібліотека базових класів). Мови для програмування в .Net та збірники. Мова C# та її переваги. Загальні схеми виконання програм в сучасних системах. Трансляція та інтерпретація програм. Структура середовища CLR .Net. Схема виконання програм в середовищі .Net. Проміжна міжплатформна мова .Net (IL –intermediate language) та її переваги. Виконання програми на вхідній мові середовища .Net в CLR. Організація типів даних в .Net. Стандартна система типів (CTS - Common Type System). Система правил специфікації типів (CLS - Common Language Specification) та їх зміст. Загальні типи даних CLS. Організація бібліотеки базових класів .Net. та простори імен. Використання просторів імен у кодї застосування. Важливі простори імен. Використання зовнішніх збірок. Отримання інформації про простори імен та типи. Використання CTS при створенні різноманітних типів. Організація середовища та робота в ньому. Структура застосування на мові C#. Клас як базовий елемент застосування. Створення застосування ConsoleApplication. Клас Program. Статичний метод Main() та його прототипи. Параметри командної строки. та їх передача в метод Main() застосування. Змінні даних. Перетворення даних. Введення та виведення даних застосування. Правильна організація застосування на мові C#. Процедурна техніка програмування з використанням статичних методів. Приклад простішого застосування на мові C #. Компілятор командної строки мови C# csc.exe та його параметри. Створення слідування з використанням середовища розробки VisualStudio.NET. Проект застосування та його контроль засобами середовища. Базові типи та типи користувача. Класи та структури в мові C#. Структурні типи та типи посилань. Визначення та характеристика структурних типів. Визначення та характеристика типів посилань. Порівняльна характеристика структурних типів та типів посилань. Створення та ініціалізація екземплярів типу. Ієрархія базових типів даних. Класи Object і ValueType. Цілочисельні типи. Дійсні типи. Логічний тип. Символьний тип. Тип string. Змінні та константи. Тип перетворення даних. Перетворення типу. Явне і неявне перетворення типу. Деякі корисні методи базових структурних типів даних. Оператори та вирази. Оператор присвоювання. Арифметичні оператори. Оператори порівняння. Логічні оператори. Побітові оператори. Комбінація оператору присвоювання і перетворення даних. Вирази перетворення даних та пріоритет операторів. Умове перетворення даних. Функція перетворення даних. Визначення методу класу як функції перетворення даних. Виклик функції перетворення. Пустий оператор. Складовий оператор. Функційний клас Math. Порядок виконання операторів програми. Оператор слідування. Оператор переходу goto. Оператори переходу if-goto. Оператори розгалуження if та if-else. Оператор багато альтернативного вибору switch. Мітка case. Оператор break. Оператор default. Оператор циклу while. Оператор циклу do-while. Оператор циклу з параметром for. Оператори переходу в циклі break і continue. Структурна і не структурна техніка програмування. Функціональний клас Console. Можливості та призначення класу Console. Властивості та методи класу Console. Настоювання вікна консолі. Виведення даних на консоль. Формати виведення даних на консоль та управляючі послідовності. Введення символьних даних з консолі. Введення числових даних з консолі. Метод Parse(string). Клас Convert для перетворення даних. Постановка задачі розробки програми. Розробка методу вирішення задачі. Розробка структури даних і алгоритму для обраного методу вирішення задачі. Форми представлення алгоритму: опис алгоритму, блок-схема алгоритму, текст програми. Програмна реалізація. Компіляція та виконання програми. Тестування програми. Налаштування програми. Опис програми. Блок-схема алгоритму. Призначення масивів і наборів даних. Визначення і використання одномірних масивів даних. Багатомірні прямокутні масиви даних. Багатомірні не вирівняні масиви даних. Клас System.Array. Клас System.Collections.ArrayList. Цикл foreach для перебору елементів набору даних. Поняття виключної ситуації. Механізм структурної обробки виключеної ситуації. Генерація виключення оператором throw. Відновлення роботи застосування після виключеної ситуації. Розробка власних класів виключення. Класи простору System.IO для роботи з файловою системою: Directory, File, FileSystemInfo, DirectoryInfo, FileInfo, FileAttributes, FileMode, FileAccess, FileShare. Потоки для роботи з текстовими файлами. Класи TextReader, TextWriter, StreamReader, StreamWriter, StringReader, StringWriter. Потоки для роботи з двійковими файлами. Класи Stream, FileStream, MemoryStream, BufferedStream. Класи для роботи з даними в двійковому форматі: BinaryReader, BinaryWriter. Сепіалізація даних. Класи форматерів

BinaryFormatter, SoapFormat-ter. Поняття делегата. Клас System.MulticastDelegate та його призначення. Синхронний і асинхронний режим виконання делегата. Визначення класу делегата. Інтерфейс IAsyncResult і його використання для контролю виконання делегату. Засоби синхронізації делегата з основним потоком. Поняття події. Зв'язок делегата та події. Використання подій для розробки застосунків. Використання подій для передачі повідомлень між об'єктами програми. Об'єктна технологія і поняття класу як типу. Типи класів: класи модулі і класи типи. Синтаксис опису класу як типу даних. Члени класу: елементи даних, методи, конструктори, деструктори, вкладені класи, події. Модифікатори опису класу та елементів класу: static, public, private, protected, internal. Визначення елементів даних. Визначення конструкторів класу. Посилання this. Схема створення об'єкту класу. Визначення методу класу. Поняття прототипу (сигнатури) методу. Тип значення, що повертає метод, формальні параметри методу. Використання класу як типу – створення екземплярів (об'єктів) класу. Доступ до елементів даних, виклик методів об'єкту. Клас System.Object як батьківський клас для усіх створюваних класів, заміщення його методів у класі, що наслідується від нього. Принципи об'єктно орієнтованого програмування. Інкапсуляція на рівнях методу і об'єкту. Поняття поля. Поле об'єкта. Статичне поле класу. Форма доступу до поля класу та об'єкта. Забезпечення інкапсуляції – обмеження доступу до поля. Специфікатор readonly. Забезпечення інкапсуляції – закриті дані та відкриті методи доступу до них. Забезпечення інкапсуляції – визначення властивостей з методами get і set та з обмеженням доступу. Вкладені класи як особлива форма інкапсуляції. Наслідування. Модель класичного наслідування – is a. Модель наслідування включення-делегування – has a. Вкладені класи як особлива форма інкапсуляції. Просте наслідування. Заміщення методу базового класу. Специфікатор new. Поліморфна поведінка об'єктів. Поліморфізм для конкретного випадку. Класичний поліморфізм. Реалізація класичного поліморфізму. Специфікатори virtual і override. Приведення для типів посилань на об'єкти класів. Визначення абстрактного методу. Визначення абстрактного класу та його призначення. Абстрактні методи і класи. Створення ієрархій класів. Використання технології для моделювання предметної області вирішуваної задачі. Тип інтерфейсу і його визначення. Посилання на інтерфейс. Інтерфейси як параметри методів. Явна реалізація методу інтерфейсу. Створення ієрархії інтерфейсів. Стандартні інтерфейси. Типи простору імен System.Collections. Механізм прибирання сміття. Управління прибиранням сміття в програмі використанням деструктора. Об'єкти, що фіналізуються. Клас System.GC і його можливості. Управління прибиранням сміття в програмі використанням інтерфейсу IDisposable. Комбінування засобів управління прибиранням сміття. Проблема універсалізації коду. Клас з універсальними методами. Універсальний клас. Конкретизація універсального методу та класу. Універсальність як доповнення ООП. Обмеження універсальності. Технологія візуальної розробки застосунків. Історія розвитку засобів та можливості. Базові елементи побудови застосунків. Структура застосування для Windows. Подієве виконання застосування в ОС Windows. Форми та їх властивості. Менеджер рішень. Панель інструментів. Робота в режимах дизайну та коду. Властивості проекту. Компіляція проекту. Форматування елементів управління форми. Приклад застосування. Створення іконки для вікна застосування. Огляд елементів управління. Група командних об'єктів. Група текстових об'єктів. Група перемикачів. Група контейнерів. Група графічних елементів. Діалогові вікна. Меню та створення меню. Контекстні меню. Стандартні вікна діалогів. Наслідування форм. Організація багато поточного застосування.

ПЕРЕЛІК ОБЛАДНАННЯ

Лабораторний практикум укомплектовано наступним устаткуванням: сучасні ПЕОМ з багатоядерними процесорами з операційною системою Windows 10, середовищем розробки програмних застосунків Visual Studio.Net 2017

СИСТЕМА ОЦІНЮВАННЯ

Розподіл балів для оцінювання успішності студента	Оцінка ECTS		Оцінка за національною шкалою		Нарахування балів
	90-100	A	відмінно		
	82-89	B	добре		
	74-81	C			
	64-73	D	задовільно		
	60-63	E			
	35-59	FX	незадовільно з можливістю повторного складання		
	0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни		

Підсумкову залікову оцінку студент може отримати як рейтингову – інтегровану оцінку з накопичених оцінок під час семестрового навчання.
Якщо студент не отримав залік за результатами поточного контролю, то він виставляється за результатами виконання ним залікової контрольної роботи.

НОРМИ АКАДЕМІЧНОЇ ЕТИКИ

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при нерозв'язності конфлікту доводиться до завідуючого кафедри, декану.

Силабус за змістом повністю відповідає робочій програмі навчальної дисципліни