

Силабус навчальної дисципліни «Розробка програм на платформі .NET»	
1. Загальна інформація	
Освітня програма (галузь, спеціальність, рівень вищої освіти, форма навчання)	<u>Освітня програма:</u> «Інженерія програмного забезпечення» <u>Галузь знань:</u> 12 «Інформаційні технології» <u>Спеціальність:</u> 121 «Інженерія програмного забезпечення» <u>Рівень вищої освіти:</u> Перший (бакалаврський) <u>Форма навчання:</u> Денна, заочна
Тип дисципліни (нормативна/вибіркова)	Нормативна
Кількість кредитів ECTS та кількість годин денна/заочна (лекції / лабораторні / самостійна робота здобувачів) Форма контролю	Кредити – 4,0 Загальний обсяг – 120 год. Денна: 16 лекц., 32 лаб. роб., 72 сам. роб. Заочна: 6 лекц., 6 лаб. роб., 108 сам. роб. Форма контролю – залік
Викладачі (ППП, наукові ступені і звання, контактний e-mail)	Рибальченко Олена Геннадіївна, ст. викладач rybalchenko@knu.edu.ua
Посилання на матеріали дисципліни (робоча програма, методичні матеріали)	https://drive.google.com/drive/folders/1LM_SGUQshEGOU4_mKvQvZjz4wGMTyM0z?usp=sharing
Мова викладання	Українська
Інформація про розклад занять	http://asu.knu.edu.ua/time-table/chair
Кафедра: (адреса, телефон, e-mail, сайт, QR-code)	вул. Віталія Матусевича, 11, корпус 1, каб. 317 м. Кривий Ріг; тел. 056-409-06-07 mpz@knu.edu.ua http://mpz.knu.edu.ua/
2. Коротка анотація до курсу	
<p>Фокус уваги курсу «Розробка програм на платформі .NET» спрямовано на оволодіння базовими інструментами середовища .Net та особливості мови програмування C#: властивості, події, індексатори, методи розширення, використання інтерфейсів, серіалізація, життєвий цикл об'єктів, засоби рефлексії, LINQ. Вивчення дисципліни має сприяти формуванню розуміння основних відомостей про принципи побудови програмного забезпечення на основі .NET-технологій, особливості організації процесу програмування на платформі .NET Framework, вмінню студентів оцінювати ступінь правильності формування програмного коду.</p>	
3. Мета та завдання курсу	
<p>Метою викладання навчальної дисципліни «Розробка програм на платформі .NET» є формування у здобувачів теоретичного базису щодо сучасних підходів і методик розробки різного роду застосувань на платформі .NET, а також формування практичних навичок з оптимального використання .NET у подальшій фаховій практиці. Завданнями вивчення дисципліни «Розробка програм на платформі .NET» є: опанування синтаксису і семантики мови C#, вивчення теорії типів і системи типізації в .NET; поглиблення знань з об'єктно-орієнтованого програмування та математичних основ ООП; застосування концепції .NET для реалізації ООП; ознайомлення з подієво-орієнтованим програмуванням та компонентним підходом до програмування; набуття вміння створювати додатки мовою C# на основі сучасних методів об'єктно-орієнтованого та компонентного програмування у інтегрованих середовищах програмування Visual Studio та Rider.</p>	



4. Що ви будете знати	5. Що ви будете вміти
<ul style="list-style-type: none"> • об'єктно-орієнтований підхід до програмування; • основні поняття ООП; • синтаксис і семантику мови C#; • теорію типів і систему типізації в .NET; • подієво-орієнтоване програмування; • застосування концепції .NET для реалізації ООП; • компонентний підхід до програмування. 	<ul style="list-style-type: none"> • використовувати основні синтаксичні та семантичні конструкції мови C#; • обґрунтовувати необхідність використання об'єктно-орієнтованої технології або компонентного підходу для вирішення конкретної задачі; • створювати консольні прикладні застосування мовою C# на основі сучасних методів об'єктно-орієнтованого та компонентного програмування; • використовувати в професійній діяльності сучасні інтегровані середовища програмування (Visual Studio, Rider); • самостійно опановувати нові методи та технології розробки програм.
6. Матеріально-технічне / інформаційне та навчально-методичне забезпечення	
<ol style="list-style-type: none"> 1. Лекційна аудиторія з мультимедійним проектором та підключенням до мережі Інтернет. 2. Аудиторія персональних комп'ютерів з операційною системою типу Windows 7, 8 або 10 та підключенням до мережі Інтернет. 3. Програмне забезпечення: Visual Studio від 2015, IDE Rider. 4. Платформа .NET та мова програмування C# 8.0: навч. посіб. / Коноваленко І.В., Марущак П.О. Тернопіль: ФОП Паляниця В.А., 2020. 320 с. 5. Troelsen A., Japikse Ph. Pro C# 10 with .NET. Foundational Principles and Practices in Programming. Apress Berkeley, 2019. 1328 p. 6. Albahari J, Albahari B. C# 7.0 in a Nutshell: The Definitive Reference. O'Reilly Media, 2017. 1024 p 7. Price M. J. C# 7 and .NET Core: Modern Cross-Platform Development. Packt Publishing, 2017. 640 p. 8. Рибальченко О.Г., Білашенко С.В. Методичні вказівки до виконання лабораторних робіт з дисципліни «Розробка програм на платформі .NET» для студ. всіх форм навч. за спец. 121 «Інженерія програмного забезпечення». Кривий Ріг, 2023. 36 с. 9. Інструкція з C# - Частина 1-2. https://professorweb.com/my/csharp/charp_theory/ (дата звернення 20.12.2024) 10. . LINQ to Objects https://professorweb.com/my/LINQ/base/level1/linq_index.php/ (дата звернення 20.12.2024). 11. https://classroom.google.com/c/Mzg4MTQzMtI2MzMy?hl=ru&cjc=qa65wnf. 	

7. Тематика курсу

Що таке .NET, CLR та IL. Синтаксис, семантика, типізація у мові C#. Структури вводу, виводу, управління та повторення. Масиви та рядки у C#. Методи, параметри методів. Правила DRY та KISS. Конструкція try...catch. Фільтрація виключень. Операція throw. Блок finally. Використання конструкції using та інтерфейсу IDisposable для безпечної роботи з некерованими ресурсами. Узагальнення та їх обмеження. Інтерфейс IEnumerable<T>, ключове слово yield. CRUD для об'єкта, що ітерується. Колекція IList<T>, List<T>, індексація по об'єкту, динамічний масив в контексті C#, основні методи. Множина ISet<T>, HashSet<T>, основні методи та властивості. Стек Stack<T>. Черга Queue<T>. Інтерфейс IDictionary<TKey, TValue>, KeyValuePair<TKey, TValue>. Словник Dictionary<TKey, TValue>, основні методи та особливості використання. Декоратор ReadOnlyCollection<T>. Клас, структура, запис. Модифікатори доступу. Поля. Властивості. Конструктори. Ключове слово static. Абстрактні класи. Спадкування в C#. Інтерфейси. Спадкування конструкторів. Ключове слово base. Перетворення типів, поліморфізм подтипів. Віртуальні члени класу, ключові слова virtual, override, new. Інтерфейс vs абстрактний клас. Делегати Action<T>, Func<T>, лямбда-функція. Події, обробка подій, клас Timer. Призначення. Проекція колекції в іншу колекцію. Фільтрація колекції. Пошук елементів за умовою: First, FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault, Any, All, Contains. Операції з властивостями або полями об'єктів колекції: Min, Max, Sum, Average. Методи Aggregate, Take, TakeWhile, Skip, SkipWhile. Сортування за умовою. Багаторівневе сортування. Методи Concat, Distinct, Union, Except, Intersect, Group By. Stream як основний клас для роботи з потоками. FileStream для передачі даних файлів. Декоратори для роботи з потоками StreamReader та StreamWriter. Робота з віддаленими серверами за допомогою HttpClient. Отримання результатів як потоків. Принцип єдиного обов'язку. Принцип відкритості/закритості. Принцип підстановки Лісков. Принцип розділення інтерфейсу. Принцип інверсії залежностей. Породжуючі патерни GoF. Поведінкові та структурні патерни. Базові поняття про бази даних. Концепція реляційної бази даних. Сутності, первинні ключі, зв'язки між сутностями, зовнішній ключ, індекси в реляційних базах даних.

8. Система оцінювання

Використовується модульно-рейтингова система оцінювання, яка передбачає розподіл балів за виконання всіх запланованих видів робіт. При цьому максимальна кількість балів за модуль при умові його бездоганного виконання дорівнює 100. Ця сума складається з балів, що накопичив студент у ході поточного контролю, та балів за виконання індивідуального проекту.

Лабораторні роботи у модулі відображують оволодіння навичками та вміння застосовувати знання на практиці і сукупно відповідають 70-ти відсоткам ваги. При зниженні якості виконання тієї чи іншої лабораторної роботи, знижується і кількість балів, якою вона оцінюється. Оцінювання кожної лабораторної роботи ведеться за наступними показниками:

1. Своєчасність практичного виконання лабораторної роботи (у тиждень згідно із графіком робіт) (0-1 бали).

2. Теоретичний захист виконаної лабораторної роботи (у тиждень наступний за тижнем планового виконання роботи) (0-3 бали).

3. Якість знайдених студентом рішень (ефективність алгоритму, доречність використання елементів інтерфейсу, тощо) (6-11 балів).

Якість знайдених студентом рішень (ефективність алгоритму, доречність використання структур даних, елементів інтерфейсу тощо) оцінюється наступним чином:

- робота виконана без зауважень - максимальний бал;
- робота виконана достатньо повно з деякими зауваженнями –75% від максимального бала;
- робота виконана не повністю – 50% від максимального бала.

Оцінка всієї лабораторної роботи знаходиться підсумовуванням балів за кожний з показників.

Робочою програмою передбачено самостійне виконання індивідуального проекту на запропоновану тему. Оцінювання індивідуального проекту відбувається за наступними складовими частинами (наведено максимально можливий бал):

- використання інтерфейсів (1 бал);
- використання делегатів/подій (2 бали);
- використання узагальнень (2 бали);
- використання конструкторів (2 бали);
- відповідність коду конвенціям про стилізацію коду C# (4 бали);
- відповідність коду SOLID, KISS, DRY (4 бали);
- введення-виведення даних в програму оформлено у вигляді окремих модулів (2 бали);
- наявність валідації даних (1 бал);
- загальний архітектурний підхід (3 бали);
- відповідність обсягів функціоналу розробленої програми вимогам до програм такого типу (9 балів).

Загальна оцінка індивідуального проекту знаходиться підсумовуванням балів за кожен складову частину.

Семестровий контроль здійснюється у формі заліку у третьому семестрі для всіх форм навчання. Для допуску до підсумкового контролю студент повинен виконати графік навчального процесу, усі види запланованих завдань і протягом семестру отримати в сумі не менше 50 балів. Підсумкова оцінка з дисципліни розраховується як середня зважена з оцінок за лабораторні роботи та індивідуальний проект.

9. Зарахування результатів неформальної освіти

Окремі модулі курсу можуть бути зараховані за умови надання здобувачем вищої освіти сертифікату про проходження он-лайн курсів, зазначених в робочій програмі дисципліни.

10. Політика курсу

Відвідування занять. Регуляція пропусків. Відвідування усіх занять є обов'язковим. У випадку пропуску лабораторного заняття, студент має виконати та здати лабораторну роботу згідно графіку, наведеного у робочій програмі дисципліни. У випадку пропуску лекції студент опрацьовує матеріал самостійно та може задати питання на консультації.

Політика академічної доброчесності регламентується Положенням про академічну доброчесність у Криворізькому національному університеті.

Використання комп'ютерів/телефонів на занятті. Використання комп'ютерів на практичних заняттях є обов'язковим задля досягнення навчальної мети.

Розробник силябусу:

Старший викладач кафедри моделювання та програмного забезпечення
Олена РИБАЛЬЧЕНКО



Завідувач кафедри моделювання та програмного забезпечення, доцент,
канд. пед. наук
Андрій СТРЮК



Гарант ОПП, канд. пед. наук
Андрій СТРЮК

