

**МЕЛІТОПОЛЬСЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО**

ФАКУЛЬТЕТ ІНФОРМАТИКИ, МАТЕМАТИКИ ТА ЕКОНОМІКИ

Кафедра інформатики і кібернетики

Назва курсу <i>Нормативний/вибірковий</i>	ПП-02 Програмування <i>Нормативний</i>
Ступінь освіти	Бакалавр
Освітня програма	Професійна освіта. Комп'ютерні технології
Рік викладання/ Семестр/ Курс (рік навчання)	2019-2020/ IV семестр / II курс
Викладач	Єремєєв В.С., Ібрагімова Л.А.
Профайл викладача	http://inf.mdpu.org.ua/2017/10/25/ieremieiev-volodimir-sergijovich/ http://inf.mdpu.org.ua/2016/01/16/ibragimova-ljudmila-anatolivna/
Контактний тел.	+380972310982
E-mail:	ibragimova@mdpu.org.ua
Сторінка курсу в ЦОДТ МДПУ ім. Б.Хмельницького	https://dfn.mdpu.org.ua/course/view.php?id=888
Консультації	Очні консультації: згідно графіку роботи кафедри інформатики і кібернетики. Онлайн-консультації: через систему ЦОДТ МДПУ ім. Б. Хмельницького.

1. АНОТАЦІЯ

Навчальна дисципліна «ПП-02 Програмування» з циклу нормативних компонентів здобувачів вищої освіти за освітньою програмою «Професійна освіта. Комп'ютерні технології» передбачає формування у здобувачів поглиблених знань, умінь і навичок в програмуванні.

Основою вивчення дисципліни є отримання здобувачами знань з теоретичних та технологічних аспектів створення сучасних мов програмування, вивчення структури мови, опанування її технологій з метою подальшого проектування та програмування сучасних програмних комплексів з використанням Java технологій.

Здобувачі мають можливість ознайомитись з основами теорії побудови масштабованих рішень: основні класи Java EE, методи розробки компонент програмних систем, інструментальні засоби та передові технології мови програмування Java.

2. МЕТА ТА ЗАВДАННЯ КУРСУ

Основна мета засвоєння курсу полягає у формуванні студентами знань, вмінь та навичок з проектування, розробки додатків, використання сучасних мов програмування, методів програмування, які застосовуються в процесі розробки додатків; а також підвищення рівня теоретичних знань про технології програмування.

Завданнями курсу є набуття компетентностей, знань, умінь та навиків на рівні новітніх досягнень у програмуванні з використанням технологій Java.

3. ПЕРЕЛІК КОМПЕТЕНТНОСТЕЙ, ЯКІ НАБУВАЮТЬСЯ ПІД ЧАС ОПАНУВАННЯ ОСВІТНІМ КОМПОНЕНТОМ

Загальні компетентності

ЗК 3. Здатність застосовувати знання у практичних ситуаціях.

ЗК 7. Здатність працювати в команді.

ЗК 9. Здатність оцінювати та забезпечувати якість виконуваних робіт.

Спеціальні (фахові) компетентності

ФК 18. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних методів та алгоритмів обчислень, структур даних, парадигм програмування на основі побудови відповідних моделей.

ФК 19. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.

ФК 20. Здатність застосовувати технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем відповідно до вимог замовника.

ФК 22. Здатність до розробки програмного забезпечення, що використовує комп'ютерні системи і мережі передачі даних.

ФК 23. Здатність застосовувати методи та засоби захисту даних в інформаційних системах та комп'ютерних мережах.

4. РЕЗУЛЬТАТИ НАВЧАННЯ

ПРН 19. Уміння ефективно використовувати математичний апарат у професійній діяльності для розв'язання задач теоретичного та прикладного характеру у процесі аналізу, синтезу та проектування інформаційних систем за галузями.

ПРН 20. Знання базових понять теорії алгоритмів, формальних моделей алгоритмів, питань обчислюваності, розв'язності та нерозв'язності масових проблем, понять складності алгоритмів.

ПРН 21. Знання структур даних та фундаментальних алгоритмів, методології та інструментальних засобів об'єктно-орієнтованого аналізу та проектування, особливостей різних парадигм програмування, принципів, моделей, методів і технологій проектування і розроблення програмних продуктів різного призначення.

ПРН 24. Знання та вміння використовувати стандарти, методи, технології і засоби управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій.

ПРН 25. Знання принципів, інструментальних засобів, мов програмування та технологій створення веб-ресурсів, баз даних, розподілених застосувань, інтелектуальних інформаційних систем тощо.

ПРН 26. Уміння використовувати методи, технології та інструментальні засоби для проектування і розробки веб-застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти запити до них, створювати розподілені бази даних, інтелектуальні інформаційні системи.

ПРН 29. Знання концепції інформаційної безпеки, принципів попередження погроз, безпеки комп'ютерних мереж та інформаційних систем.

ПРН 31. Уміння здійснювати підбір і підготовку інформації та задач проектній команді, ставити цілі і формулювати завдання для реалізації проектів у галузях професійної освіти та інформаційних технологій.

ПРН 32. Уміння використовувати сучасну комп'ютерну техніку, пакети прикладних програм і програмувати на рівні, достатньому для розв'язування фізичних та математичних задач, комп'ютерного моделювання явищ і процесів, виконання обчислювальних експериментів.

5. ОБСЯГ КУРСУ

Вид заняття	Загальна кількість	Лекції	Практичні/ лабораторні заняття	Самостійна робота
Кількість годин	210 годин	40 годин	54 години	60 годин

6. ПОЛІТИКА КУРСУ

Політика навчання через дослідження:

➤ Курс є складовою освітньо-професійної програми, тому усі його складові розглядаються у контексті відповідності наукових інтересів бакалаврів.

Політика академічної поведінки та етики:

➤ Не пропускати та не запізнюватися на заняття за розкладом;
➤ Вчасно виконувати завдання семінарів та питань самостійної роботи;
➤ Вчасно та самостійно виконувати контрольні-модульні завдання.
➤ Дотримуватись Кодексу академічної доброчесності, прийнятого у МДПУ імені Богдана Хмельницького https://mdpu.org.ua/wp-content/uploads/2020/11/Kodeks-akadem-dobrochnosti_2020.pdf та Положення про Академічну доброчесність https://mdpu.org.ua/wp-content/uploads/2020/11/akademichna-dobrochnist_2020.pdf. Здобувачі освіти мають самостійно виконувати навчальні завдання, завдання поточного та періодичного контролю, самостійні завдання, посилаючись на джерела інформації у разі запозичень ідей, тверджень, відомостей; дотримуватись норм законодавства про авторське право.

Політика щодо дедлайнів та перекладання: роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Перекладання модулів відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

Політика щодо академічної доброчесності: усі письмові роботи перевіряються на наявність плагіату і допускаються до захисту із коректними текстовими запозиченнями не більше 20%. Списування під час контрольних робіт заборонені (в т.ч. із використанням мобільних девайсів). Політика щодо відвідування: Відвідування занять є обов'язковим. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу.

Поточний контроль: усне опитування в ході лекцій та практичних занять, перевірка завдань (у тому числі самостійної роботи), оцінювання правильності вирішення тестових та практичних завдань на семінарських заняттях.

7. СТРУКТУРА КУРСУ

Кількість годин	Тема	Форма діяльності (заняття, кількість годин)	Література	Завдання	Вага оцінки	Термін виконання
Блок 1. Реалізація головних концепцій об'єктно-орієнтованого програмування у мові С#						
18	Тема 1. Графічні можливості мови програмування С#.	Лекція (6 год.), практичне заняття (8 год.), самостійна робота (4 год.)	Основна: 1-4, Допоміжна: 1	Опитування Перевірка виконаних завдань	5	впродовж четвертого навчального семестру (перший періодичний контроль)
11	Тема 2. Використання індексаторів та властивостей.	Лекція (4 год.), практичне заняття (2 год.), самостійна робота (5 год.)	Основна: 1-4, Допоміжна: 1	Опитування Перевірка виконаних завдань	5	впродовж четвертого навчального семестру (перший періодичний контроль)
10	Тема 3. Обробка виключень.	Лекція (2 год.), практичне заняття (4 год.), самостійна робота (4 год.)	Основна: 1-4, Допоміжна: 1	Опитування Перевірка виконаних завдань	5	впродовж четвертого навчального семестру (перший періодичний контроль)
11	Тема 4. Колекції.	Лекція (2 год.), практичне заняття (4 год.), самостійна робота (5 год.)	Основна: 1-4, Допоміжна: 1	Опитування Перевірка виконаних завдань	5	впродовж четвертого навчального семестру (перший періодичний контроль)
10	Тема 5. Рядки та регулярні вирази	Лекція (2 год.), практичне заняття (4 год.), самостійна робота (4 год.)	Основна: 1-4, Допоміжна: 1	Опитування Перевірка виконаних завдань	5	впродовж четвертого навчального семестру (перший періодичний контроль)

15	Тема 6. Використання атрибутів.	Лекція (4 год.), практичне заняття (6 год.), самостійна робота (5 год.)	Основна: 1-4, Допоміжна: 1	Опитування Перевірка виконаних завдань	5	впродовж четвертого навчального семестру (перший періодичний контроль)
Блок 2. Введення в патерни проектування						
12	Тема 7. Уніфікована мова моделювання (UML)	Лекція (4 год.), практичне заняття (2 год.), самостійна робота (6 год.)	Основна: 1-4, Допоміжна: 1	Опитування Перевірка виконаних завдань	5	впродовж четвертого навчального семестру (другий періодичний контроль)
12	Тема 8. Принципи проектування програмного забезпечення на основі аналізу предметної області.	Лекція (4 год.), практичне заняття (2 год.), самостійна робота (6 год.)	Основна: 1-4, Допоміжна: 1	Опитування Перевірка виконаних завдань	5	впродовж четвертого навчального семестру (другий періодичний контроль)
9	Тема 9. Введення в шаблони проектування.	Лекція (2 год.), практичне заняття (2 год.), самостійна робота (5 год.)	Основна: 1-4, Допоміжна: 1	Опитування Перевірка виконаних завдань	5	впродовж четвертого навчального семестру (другий періодичний контроль)
12	Тема 10. Породжувальні патерни.	Лекція (2 год.), практичне заняття (6 год.), самостійна робота (4 год.)	Основна: 1-4, Допоміжна: 1	Опитування Перевірка виконаних завдань	5	впродовж четвертого навчального семестру (другий періодичний контроль)
14	Тема 11. Вступ до структурних патернів.	Лекція (4 год.), практичне заняття (6 год.), самостійна робота (4 год.)	Основна: 1-4, Допоміжна: 1	Опитування Перевірка виконаних завдань	5	впродовж четвертого навчального семестру (другий періодичний контроль)
16	Тема 12. Застосування	Лекція (4 год.),	Основна: 1-4,	Опитування	5	впродовж четвертого

	патернів проектування.	практичне заняття (8 год.), самостійна робота (4 год.)	Допоміжна: 1	Перевірка виконаних завдань		навчального семестру (другий періодичний контроль)
Блок 3. Навчальна практика						
60	Навчальна практика	Практичне заняття (60 год.)	Основна: 1-4, Допоміжна: 1	Захист звіту, перевірка виконаних завдань	5	впродовж четвертого навчального семестру

7.2. СТРУКТУРА КУРСУ (ЛЕКЦІЙНИЙ БЛОК)

Теми лекцій та питання, що вивчаються
<p>Графічні можливості мови програмування C#.</p> <ol style="list-style-type: none"> Особливості графічного виведення даних. Логічна система координат. Простори імен GDI+. Обробка повідомлення перемальовування. Програмне генерування повідомлення перемальовування. Графічні об'єкти GDI+. Використання пензлів, пер та шрифтів. Робота з графічними зображеннями.
<p>Використання індексаторів та властивостей.</p> <ol style="list-style-type: none"> Реалізація принципу поліморфізму у мові C#. Віртуальні методи. Абстрактні класи та методи. Реалізація поліморфної поведінки на базі абстрактного класу. Правила застосування абстрактних класів. Інтерфейси. Реалізація поліморфної поведінки на базі інтерфейсу. Правила застосування інтерфейсів. Принципи переважання операцій. Особливості використання функції operator. Індексатори. Властивості
<p>Обробка виключень.</p> <ol style="list-style-type: none"> Види помилок у програмах. Проблеми традиційного підходу до обробки помилок. Механізм обробки виключень. Класи виключень стандартної бібліотеки .NET. Синтаксис обробки виключень. Перевірка на арифметичне переповнення.
<p>Колекції.</p> <ol style="list-style-type: none"> Основні класи та інтерфейси колекцій. Класи колекцій загального призначення. Застосування колекцій при написанні програмного засобу.

<p>Рядки та регулярні вирази</p> <ol style="list-style-type: none"> 1. Основні методи класу Char. 2. Основні методи класу String. 3. Основні елементи класу StringBuilder. 4. Метасимволи в регулярних виразах. 5. Уточнюючі метасимволи в регулярних виразах. Пошук в тексті за шаблоном.
<p>Використання атрибутів.</p> <ol style="list-style-type: none"> 1. Використання атрибутів. 2. Призначені для користувача атрибути. 3. Визначення атрибутивного класу. 4. Приклад настоюваного атрибута. Написання програм з використання атрибутів.
<p>Уніфікована мова моделювання (UML)</p> <ol style="list-style-type: none"> 1. Призначення уніфікованої мови програмування. 2. Види діаграм та їх призначення. 3. Особливості побудови діаграм.
<p>Принципи проектування програмного забезпечення на основі аналізу предметної області</p> <ol style="list-style-type: none"> 1. Поняття проектування програмного забезпечення на основі аналізу предметної області. 2. Визначення аналізу предметної області, еквівалентність понять сутність предметної області та класу об'єктів. Загальні принципи проектування. 3. Типові причин перепроєктування. 4. Якісні характеристики коду: гнучкість, переносність, розширюваність. 5. Механізми повторного використання коду: спадкування, композиція, делегування обов'язків. 6. Плюси і мінуси описаних механізмів, випадки найефективнішого використання
<p>Введення в шаблони проектування.</p> <ol style="list-style-type: none"> 1. Передумови виникнення шаблонів проектування. 2. Забезпечення багаторазового використання коду. 3. Класифікація патернів проектування: патерни проектування класів/об'єктів (GoF), архітектурні системні патерни, патерни інтеграції корпоративних інформаційних систем. 4. Класифікація патернів проектування класів/об'єктів: породжуючі, структурні та поведінкові патерни. 5. Шаблон Абстрактна Фабрика (Abstract Factory): визначення, мета використання, переваги та недоліки використання, застосовність, діаграма класів, учасники шаблону, програмна реалізація.
<p>Породжувальні патерни.</p> <ol style="list-style-type: none"> 1. Породжувальні патерни – загальне визначення, призначення, способи використання, випадки застосування: 2. Прототип (Prototype), Одинак (Singleton), Фабричний Метод (Factory Method): визначення, мета використання, переваги та недоліки використання, застосовність, діаграма класів, учасники шаблонів, програмна реалізація. 3. Загальні висновки стосовно областей та способів використання породжуючих патернів, порівняльна характеристика. 4. Приклади практичного застосування породжувальних патернів.
<p>Вступ до структурних патернів.</p> <ol style="list-style-type: none"> 1. Загальний опис структурних патернів. 2. Структурні патерни рівня класів та об'єктів. Програмні механізми, що використовуються при застосуванні структурних патернів. Спільність, різниця, способи реалізації. Випадки та приклади застосування. 3. Шаблони Міст (Bridge) та Адаптер (Adapter) Визначення патерну Міст. Призначення, синонімічні назви, мотивація використання, застосовність, переваги та недоліки

<p>використання, діаграма класів, учасники шаблону, відношення учасників, результати використання, програмна реалізація, відомі застосування. Визначення патерну Адаптер.</p> <p>4. Призначення, синонімічні назви, мотивація використання, застосовність, переваги та недоліки використання, діаграма класів, учасники шаблону, відношення учасників, результати використання, програмна реалізація, відомі застосування.</p>
<p>Застосування патернів проектування процесі написання програмних засобів</p> <ol style="list-style-type: none"> 1. Введення до патернів поведінки. 2. Загальні висновки стосовно патернів проектування класів/об'єктів (GoF).

7.3. СТРУКТУРА КУРСУ (ПРАКТИЧНЕ ЗАНЯТТЯ)

Теми практичних занять
Створення найпростіших графічних зображень.
Створення простих анімацій в мові програмування C #.
Створення графіку функцій за допомогою компонента Chart та за допомогою класу Graphics.
Використання властивостей та індексаторів при розробці програм
Розробка програм з реалізації виключень. Перевірка значень, які вводяться з клавіатури та їх обробка.
Використання регулярних виразів при розробці додатків мовою C#
Використання стандартних колекцій під час розробки програм.
Використання атрибутів в процесі написання програм.
Використання основних бібліотек платформи MICROSOFT .NET при розробці додатків. Інтерфейси.
Моделювання програмного засобу засобами UML
Приклади використання породжувального патерну Фабричний Метод (Factory Method)
Породжувальний патерн Абстрактна Фабрика (Abstract Factory)
Розв'язання задач із використанням структурного патерну Адаптер (Adapter)
Побудова програмних додатків із використанням структурного патерну Фасад (Façade)
Загальні принципи побудови програмних додатків із використанням шаблонів поведінки. Патерн Ітератор (Iterator)
Розробка програмних додатків із використанням патерну поведінки Спостерігач (Observer)

7.4 СТРУКТУРА КУРСУ (ТЕМИ ДЛЯ САМОСТІЙНОГО ОПРАЦЮВАННЯ)

Тема для самостійного опрацювання
Відношення агрегації. Реалізація агрегації у мові C#.
Відношення спадкування. Ієрархія класів. Синтаксис спадкування у мові C#.
Управління механізмом доступу до членів базового класу. Механізми використання захищених членів класу. Використання специфікатора доступу <code>protected</code> для надання членам класу статусу захищеності.
Поняття про віртуальні базові класи. Перевизначення методів. Заборона спадкування.
Рядкове представлення об'єкта.
Показчики на похідні типи – підтримка динамічного поліморфізму. Механізми реалізації віртуальних функцій.
Поняття про віртуальні функції. Успадкування віртуальних функцій.
Поняття про суто віртуальні функції та абстрактні класи. Класові ієрархії та абстрактні класи.
Раннє та пізнє зв'язування.
Контейнер. Ієрархія об'єктів.
Організація файлового введення-виведення даних. Відкриття та закриття файлу. Зчитування та запис текстових файлів.
Неформатне введення-виведення даних у двійковому режимі. Зчитування та записування у файл блоків даних. Використання функції <code>eof()</code> для виявлення кінця файлу.

8. ФОРМИ КОНТРОЛЮ І МЕТОДИ НАВЧАННЯ

Оцінювання результатів навчання здобувачів вищої освіти здійснюється відповідно до «Положення про бально-накопичувальну систему оцінювання результатів навчання здобувачів вищої освіти у МДПУ імені Богдана Хмельницького» (протокол №5 від 24.10.2019) <https://drive.google.com/file/d/1BDRNtAJupqmHkldtICJTkvL-LNTIjWRX/view>.

Усне та письмове опитування, перевірка виконання практичних завдань, презентація результатів проєктної діяльності, тестування, захист лабораторних робіт, звіт про результати навчальної практики, поточний контроль, підсумковий семестровий контроль

Методи навчання. Студентсько-центроване навчання. Професійно-орієнтоване навчання, індивідуально-творчий підхід. Очний (*offline*) у вигляді лекційних та семінарських занять. Змішаний (*blended*) через систему Центру освітніх дистанційних технологій МДПУ імені Б.Хмельницького, Zoom, Інтернет. Усі складові курсу розглядаються у контексті відповідності наукових інтересів бакалаврів. Словесні методи (бесіди та дискусії, розповідь, пояснення, лекція); наочні методи (ілюстрування, демонстрування та самостійне спостереження); лабораторні роботи; проблемно-пошукові методи; методи стимулювання та мотивації навчально-

пізнавальної діяльності; інтерактивні методи (частково пошукові методи, дослідні методи (проект), мозковий штурм, робота в мікрогрупах), самостійна робота студентів, навчальна практика.

9. СИСТЕМА ОЦІНЮВАННЯ ТА ВИМОГИ

Методи контролю результатів навчання	Максимальна кількість балів та критерії до їх зарахування
<p>Поточний контроль (робота на лабораторних заняттях)</p>	<p>Середньозважена оцінка: 20 балів Максимальна оцінка за 1 заняття – 5 балів</p> <ul style="list-style-type: none"> – 5 балів – завдання виконане вірно, реалізовано найбільш ефективний алгоритм реалізації, супроводжується необхідним поясненням і обґрунтуванням, можуть бути допущені окремі помилки, які є наслідком неухважності, але не призводять до помилок у виконанні програми і не демонструють відсутність знань з дисципліни. – 4 бали – завдання виконане вірно, супроводжується неповним поясненням, наявні помилки, які є наслідком неухважності, але не призводять до помилок у виконанні програми. – 3 бали – завдання виконане, але результат не повністю відповідає вимогам, пояснення неповне, пропущені логічні кроки, наявні помилки, які свідчать про пробіли у знаннях з дисципліни. – 2 бали – завдання виконане частково, результат не відповідає вимогам, пояснення відсутнє, наявні помилки, які свідчать про великі пробіли у знаннях з дисципліни. – 1 бал – завдання не виконане, є спроби обрати спосіб вирішення, великі пробіли у знаннях з дисципліни.
<p>Періодичний контроль</p>	<p>Разом 30 балів:</p> <p>Теоретичне питання (10 балів)</p> <ul style="list-style-type: none"> – 1 бал – фрагментарні знання при незначному загальному їх обсязі за відсутності сформованих умінь та навичок; відповідь часткова, приклади відсутні; – 2 бали – початковий рівень знань теоретичного матеріалу; елементарні, нестійкі навички роботи з інформаційними технологіями; відповідь фрагментарна, приклади відсутні; – 3 бали – відтворення матеріалу за аналогією з елементами логічних зв'язків; стійкі навички виконання елементарних дій з опрацювання інформації на комп'ютері; відповідь фрагментарна, приклади відсутні або такі, що недостатньо розкривають питання; – 4 бали – студент уміє застосовувати вивчений матеріал у стандартних ситуаціях; може пояснити основні процеси; відповідь фрагментарна, приклади відсутні або такі, що недостатньо розкривають питання; – 5 балів – студент вільно володіє навчальним матеріалом; вміє узагальнювати і систематизувати навчальну інформацію; самостійно знаходить і виправляє допущені помилки; може аргументовано обрати раціональний спосіб виконання навчального завдання; – 6 балів – відповідь повна, наведені обґрунтування та частково сформульовані основні принципи та закономірності, проте в доведенні порушено правильний ланцюг міркувань, наведені приклади в достатній мірі пояснюють матеріал теоретичного питання;

- 7 балів – відповідь повна, наведені обґрунтування та сформульовані основні принципи та закономірності, проте в доведенні порушено правильний ланцюг міркувань, наведені приклади частково пояснюють матеріал теоретичного питання;
- 8 балів – студент володіє узагальненими знаннями; відповідь неповна, достатньо обґрунтована, сформульовані основні принципи та закономірності, але в аргументації порушено правильний ланцюг міркувань, наведені приклади в достатній мірі пояснюють матеріал теоретичного питання;
- 9 балів – студент володіє узагальненими знаннями, вміє використовувати інформацію відповідно до мети і завдань пізнавальної діяльності; відповідь повна, достатньо обґрунтована, сформульовані основні принципи та закономірності, наведені приклади в достатній мірі пояснюють матеріал теоретичного питання;
- 10 балів – студент має стійкі системні знання та продуктивно їх використовує, стійкі навички керування інформаційною системою в нестандартних ситуаціях; відповідь повна, розгорнута з обґрунтованими твердженнями, сформульовані основні принципи та закономірності, вказані причинно-наслідкові зв'язки, наведено приклади застосування матеріалу даного теоретичного питання з необхідним поясненням.

Практичне завдання (10 балів)

- 1 бал – завдання не виконано;
- 2 бали – завдання виконано частково, наявні суттєві помилки;
- 3 бали – завдання виконано частково, наявні незначні помилки;
- 4 бали – завдання виконано повністю, пояснень до тексту програми немає;
- 5 балів – завдання виконано повністю, пояснення до тексту програми неповні, з помилками;
- 6 балів – завдання виконано повністю, пояснення до тексту програми неповні;
- 7 балів – завдання виконано повністю, пояснення до тексту програми неповні;
- 8 балів – завдання виконано повністю, пояснення до тексту програми неповні;
- 9 балів – завдання виконано повністю, пояснення до тексту програми повні;
- 10 балів – завдання виконано повністю з використанням найбільш оптимальних засобів, пояснення до тексту програми повні.

Тести (10 балів)

1 бал – відповідь вірна, 0 балів – відповідь невірна

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання

0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни
------	----------	--	---

9. РЕКОМЕНДОВАНА ЛІТЕРАТУРА ТА ІНФОРМАЦІЙНІ РЕСУРСИ

ОСНОВНА ЛІТЕРАТУРА

1. Булгакова О.С. Інформатика: візуальне програмування: навч.-метод. посіб. / О.С. Булгакова, В.В. Зосімов, Н.А. Броницька, Н.В. Танкова; за ред. О.С. Булгакової, - Стеретипне вид. Херсон: Олді –Плюс. 2017. 312 с.
2. Голуб Б.М. С#. Концепція та синтаксис. Навч. посібник / Б.М. Голуб, – Львів: Видавничий центр ЛНУ імені Івана Франка, 2006. – 136 с.
3. Ковалюк Т.В. Алгоритмізація та програмування: підручник для студ-тів вищих навч. закладів/ Т.В. Ковалюк; за заг. ред. В.В. Пасічника Л.: Магнолія-2006. 2013. 399с.
4. Шилдт Г. С# 4.0: полное руководство / Г. Шилдт. – М.: ООО “И.Д. Вильямс”, 2011. – 1056 с.

Допоміжна:

1. Глибовець А. М. Практикум з мови програмування: навч. посіб./ А. М. Глибовець, М. М. Глибовець, В. С. Проценко. К.: ВД "Києво-Могилянська академія". 2010. 209 с.
2. Павловская Т.А. С#. Программирование на языке высокого уровня. Учебник для вузов / Т.А. Павловская – СПб.: Питер, 2007. – 432 с.

Інформаційні ресурси:

1. <http://www.intuit.ru/department/se/csharpdevtools/>
2. <http://www.intuit.ru/department/pl/phlcsharp/>
3. <http://www.intuit.ru/department/se/csharpvs2005>
4. <http://www.intuit.ru/department/se/msadonet/>
5. <http://www.ksuonline.ksu.ks.ua>
6. <http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>
7. www.microsoft.com
8. www.c-sharpcorner.com
9. www.functionx.com