КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

БУДІВНИЦТВА І АРХІТЕКТУРИ

**БАКАЛАВР**

Кафедра інформаційних технологій

«ЗАТВЕРДЖУЮ»

Декан факультету автоматизації і інформаційних технологій

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ І.В. Русан /

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 року

НАВЧАЛЬНА РОБОЧА ПРОГРАМА ДИСЦИПЛІНИ

«**ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ**»

(назва навчальної дисципліни)

|  |  |
| --- | --- |
| шифр | назва спеціальності |
| 123 |  «Комп’ютерна інженерія» |
|  | назва спеціальності |
| 125 |  «Кібербезпека» (Безпека інформаційних і комунікаційних систем) |

|  |  |
| --- | --- |
| Розробник: |  |
| Гончаренко Т.А., кандидат технічних наук, доцент  |  |
| (прізвище та ініціали, науковий ступінь, звання) | (підпис) |

Робоча програма затверджена на засіданні кафедри інформаційних технологій проектування та прикладної математики

Протокол № 15 від "25" травня 2020 року

Завідувач кафедри ІТ / Цюцюра С.В. /

 (підпис) (прізвище та ініціали)

Схвалено навчально-методичною радою ФАІТ

Протокол № 14 від "25" травня 2020 року

Голова НМР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_Гаркавенко О.М.)

(підпис) (прізвище та ініціали)

**ВИТЯГ З НАВЧАЛЬНОГО ПЛАНУ 2020-2021 рр.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| шифр | ОР, бакалавр | Форма навчання: **денна** | Форма контролю | Семестр | Відмітка про погодження |
| Назва спеціальності (спеціалізації) | Кредитів на сем. | Обсяг годин | Кількість індивідуальних робіт |
| Всього | аудиторних |
| Разом | у тому числі |
| Л | Лр | Пз | КП | КР | РГР | Роб |
| 123 | «Комп’ютерна інженерія» | ***4*** | ***94*** | ***50*** | ***24*** | ***26*** |  |  |  | ***1*** |  | ***Залік*** | ***3*** |  |
| 125 | «Кібербезпека»  | ***4*** | ***94*** | ***50*** | ***24*** | ***26*** |  |  |  | ***1*** |  | ***Залік*** | ***3*** |  |
| 123 | «Комп’ютерна інженерія» | ***4*** | ***94*** | ***50*** | ***24*** | ***26*** |  |  | ***1*** |  |  | ***Екз*** | ***4*** |  |
| 125 | «Кібербезпека»  | ***4*** | ***94*** | ***50*** | ***24*** | ***26*** |  |  | ***1*** |  |  | ***Екз*** | ***4*** |  |

### Мета та завдання навчальної дисципліни

**Метою навчальної дисципліни** є набуття студентами теоретичних знань та практичних навичок по об’єктно-орієнтованому програмуванню; ознайомлення з сучасними мовами об’єктно-орієнтованого програмування та сучасними засобами розробки програм; набуття первинних знань з об’єктно-орієнтованого аналізу та проектування інформаційних систем та використанню UML-нотацій для опису програмного застосування на різних етапах розробки.

**Основними завданнями дисципліни** є

• вивчення базових понять об’єктно-орієнтованого підходу до розробки програм, синтаксису та семантики сучасних об’єктно-орієнтованих мов програмування;

• знайомство з об’єктно-орієнтованою технологією розробки програмного забезпечення (об’єктно-орієнтований аналіз предметної області, об’єктно-орієнтоване проектування та програмування ПЗ);

• оволодіння сучасними інструментальними програмними засобами розробки об’єктно-орієнтованого програмного забезпечення.

1. **Компетенції студентів, що формуються в результаті засвоєння дисципліни**

|  |  |  |
| --- | --- | --- |
| Код | Зміст | Результати навчання |
| Спеціальні (фахові) компетентності. Загально-професійні |
| **ЗК1.** |

|  |
| --- |
| Здатність до абстрактного мислення, аналізу та синтезу |

 | *Знати*: основи проектування програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування,  |
| *Вміти*: проектувати, розробляти та аналізувати алгоритми розв’язання обчислювальних та логічних задач |
| **ЗК3**. | Знання та розуміння предметної області та розуміння професійної діяльності. | *Знати*: основи об'єктно-орієнтованого програмування; порядок постановки задачі як сукупності класів, що відповідають об'єктам і процесам певного предметного середовища. |
| *Вміти*: розробляти програмне забезпечення на базі сучасних інструментальних засобів розробки програм. |
| **СК3.** |

|  |
| --- |
| Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв’язності та нерозв’язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.  |

 | *Знати*: методи створення і використання технології виключень для підвищення надійності програмного забезпечення; розробки архітектури, модулі та компоненти програмних систем.  |
| *Вміти*: застосовувати методи компонентної розробки програмного забезпечення, обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення,будувати UML –діаграми для опису елементів програмного забезпечення; за побудованими UML – діаграмами розробляти елементи програмного забезпечення; використовувати стандартні класи під час розробки програмного забезпечення; |
| **СК8.** | Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об’єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління | *Знати*:; порядок створення класів та їх будову; організацію безпечного доступу до інформації завдяки застосуванню засобів інкапсуляції; методи повторного використання коду та інформаційних компонентів через механізми спадкування класів та поліморфізму |
| *Вміти*: описувати абстрактні типи даних – класи та утворювати ієрархію класів;використовувати принци поліморфізму під час розробки ієрархії класів; використовувати принцип інкапсуляції при розробці класів; |

### Програма навчальної дисципліни

Модуль 1. Об’єктно-орієнтований підхід до створення програмного забезпечення

Змістовий модуль 1. Технологія об’єктно-орієнтованого програмування

### Тема 1. Основи об’єктно-орієнтованого аналізу та проектування програмного забезпечення.

Еволюція технології розробки програмного забезпечення. Структурний та об’єктно-орієнтований підходи. Визначення об’єктно-орієнтованого аналізу, проектування та програмування.

Поняття об’єктно-орієнтованого аналізу предметного середовища. Загальні принципи побудови об’єктної модель предметного середовища.

Поняття об’єктно-орієнтованого проектування програмного забезпечення. Поняття класу та об’єкту (екземпляру класу). Поля та методи класу.

Взаємовідносини між класами та об’єктами. Поняття успадкування. Композиція та колекція об’єктів.

Основи об’єктно-орієнтованого проектування програмного забезпечення мовою UML. Основні діаграми UML. Правила побудови UML-діаграми класів.

### Тема 2. Об’єктно-орієнтоване програмування. Сучасні інструментальні засоби візуального програмування.

Сучасні середовища візуального програмування. Основні компоненти графічного інтерфейсу користувача. Використання бібліотеки візуальних компонентів (меню, панелі інструментів, шаблони діалогових вікон тощо). Побудова та опис діаграм прецедентів.

### Тема 3. Основи програмування керованого подіями.

Архітектура програм, орієнтованих на події. Обробники подій від миші, клавіатури, команд меню, елементів управління тощо. Побудова та опис діаграм взаємодії та компонентів.

Змістовий модуль 2. Основи об’єктно-орієнованої мови програмування С#

### Тема 1. Розробка класів та об’єктів мовою С#.

Опис класу. Конструктори, деструктори класів. Створення екземпляру класу. Масиви об’єктів.

Статичні, константні члени класів.

### Тема 2. Інкапсуляція.

Поняття інкапсуляції. Директиви видимості класів С# за замовчанням. Організація доступу до захищених полів класу. Дружні функції та класи.

### Тема 3. Спадковість, просте та множинне успадкування.

Поняття спадковості. Опис класів-нащадків в С# при одиничній спадковості. Директиви видимості при спадкуванні в С#. Явний виклик конструкторів-предків. Послідовність викликів конструкторів/ деструкторів для ієрархії класів. Множинна спадковість в С#. Проблеми ромбовидної спадковості, віртуальні предки.

### Тема 4. Покажчики на об’єкти. Передача об’єктів як параметрів функцій.

Операції динамічного виділення пам’яті new та delete мови С#.

Особливості передачі параметрів-об’єктів за значенням мови С#. Використання покажчиків та посилань на об’єкти. Використання конструктору копії об’єкту. Повернення об’єкту в якості результату роботи функції.

### Тема 5. Перевантаження функцій.

Неоднозначність при перевантаженні (сумісність типів параметрів, параметри-посилання, параметри за замовчанням). Перевантаження методів класу та конструкторів.

### Тема 6. Перевантаження операцій.

Перевантаження бінарних та унарних операції. Перевантаження операції індексування. Перевантаження операції присвоєння. Особливості присвоєння об’єктів мови С#. Перевантаження операцій потокового уведення-виведення.

### Тема 7. Поліморфізм.

Поняття раннього та пізнього зв’язування. Поняття перекриття та заміщення методів. Механізм виклику заміщуваних методів, таблиці віртуальних методів. Використання покажчиків на класи, як спосіб застосування поліморфізму. Виклик заміщуваних методів нащадків з класів предків. Абстрактні методи.

### Тема 8. Шаблони функцій і класів.

Поняття шаблону функції. Опис шаблону. Створення екземпляру шаблону функції. Відмінності перевантаження функції та шаблонів функції.

Опис шаблону-класу. Створення екземпляру шаблону класу.

Змістовий модуль 3. Оброблення виключень та бібліотеки класів

### Тема 1. Бібліотека стандартних шаблонів STL: основні поняття.

Складові бібліотеки STL: контейнери, ітератори, алгоритми. Базові класи-контейнери, їх загальна характеристика. Особливості роботи з елементами контейнера небазових типів. Ітератори як покажчики на елементи контейнерів, типи ітераторів. Загальне поняття алгоритму.

### Тема 2. Використання класів-контейнерів бібліотеки STL: vector, list, map.

Клас-контейнер vector. Способи створення. Методи класу вектор. Ітератори на елементи вектора.

Клас-контейнер list. Способи створення. Методи класу списка. Ітератори на елементи списка.

Клас-контейнер map. Способи створення. Методи класу асоціативний список. Ітератори на елементи.

### Тема 3. Застосування алгоритмів бібліотеки STL.

Базові алгоритми бібліотеки STL. Поняття предикату, унарний та бінарний предикат. Приклади використання алгоритмів.

### Тема 4. Класи потоків уведення та виведення.

Перевантаження операцій потокового уведення/виведення. Файлове уведення/виведення. Управління потоковим уведенням/виведенням.

### Тема 5. Оброблення виняткових ситуацій в С#.

Використання оператору try-catch. Повторне збудження виключної ситуації throw.

### Тема 6. Розроблення DLL-бібліотек.

Модуль 2. Курсова робота. Розробка об’єктно-орієнтованої програми

Змістовий модуль 1. Об’єктно-орієнтований аналіз та проектування програми

Тема 1. Дослідження та опис предметної області, розробка функцій програми, розробка прецедентів.

Тема 2. Розробка концептуальної моделі предметної області та діаграми класів.

Тема 3. Розробка діаграми компонентів.

Змістовий модуль 2. Розробка об’єктно-орієнтованої програми

Тема 1. Розробка математичних постановок задач.

Тема 2. Розробка ієрархії класів програмних об’єктів.

Тема 3. Розробка програмного інтерфейсу користувача.

Тема 4. Реалізація методів класів програмних об’єктів.

Тема 5. Перевірка працездатності програмного забезпечення: тестування та налагодження програми.

**Змістовий модуль 3. Оформлення пояснювальної записки до курсової роботи**

Тема 1. Побудова UML-діаграм: прецедентів, класів, послідовностей, компонентів.

Тема 2. Опис UML-діаграм, програмного інтерфейсу користувача, тестових прикладів роботи програми.

**6. Теми лабораторних занять**

|  |  |  |
| --- | --- | --- |
| № | Назва теми | Кількість годин |
| денна | заочна |
|  | Сучасні інструментальні засоби візуального програмування. Основи програмування керованого подіями. | 2 |  |
|  | Основи програмування керованого подіями. | 2 |  |
|  | Створення багатовіконних застосувань | 2 |  |
|  | Розробка класів та об’єктів мовою С#.  | 2 |  |
|  | Інкапсуляція. | 2 |  |
|  | Спадковість, просте успадкування.  | 2 |  |
|  | Спадковість, множинне успадкування.  | 2 |  |
|  | Перевантаження функцій.  | 2 |  |
|  | Перевантаження бінарних та унарних операцій.  | 2 |  |
|  | Поліморфізм.  | 2 |  |
|  | Використання класів-контейнерів бібліотеки STL: vector, list, map. | 2 |  |
|  | Застосування алгоритмів бібліотеки STL. | 2 |  |
|  | Класи потоків уведення та виведення.  | 2 |  |
|  | Оброблення виняткових ситуацій в С#. | 2 |  |
|  | Розроблення DLL-бібліотек. | 2 |  |
|  | ***Разом*** | ***30*** |  |

**7. Самостійна робота**

|  |  |  |
| --- | --- | --- |
| № | Назва теми | Кількість годин |
| денна | заочна |
|  | Основи об’єктно-орієнтованого аналізу та проектування програмного забезпечення. | 4 |  |
|  | Об’єктно-орієнтоване програмування. Сучасні інструментальні засоби візуального програмування.  | 4 |  |
|  | Основи програмування керованого подіями.  | 4 |  |
|  | Розробка класів та об’єктів мовою С#. Інкапсуляція. | 4 |  |
|  | Розробка класів та об’єктів мовою С#. Спадковість, просте та множинне успадкування.  | 4 |  |
|  | Перевантаження функцій. Покажчики на об’єкти. Передача об’єктів як параметрів функцій.  | 4 |  |
|  | Розробка класів та об’єктів мовою С#. Перевантаження операцій.  | 4 |  |
|  | Розробка класів та об’єктів мовою С#. Поліморфізм.  | 4 |  |
|  | Шаблони функцій і класів. | 4 |  |
|  | Бібліотека стандартних шаблонів STL: основні поняття. | 4 |  |
|  | Використання класів-контейнерів бібліотеки STL: vector, list, map та застосування алгоритмів бібліотеки STL. | 5 |  |
|  | Розроблення DLL-бібліотек. | 5 |  |
|  |  **Всього** | **50** |  |
|  |  |  |  |
|  | **Індивідуальне завдання. РГР. Реалізація спадковості.** | **24** |  |
|  | **Індивідуальне завдання. Модуль 2. Курсова робота.**  | **56** |  |
|  |  |  |  |
|  | **Підготовка до іспиту** |  |  |
|  | Повторення теоретичного матеріалу змістовного модуля 1 | 4 |  |
|  | Розв’язок практичних завдань до ЗМ1. Побудова UML - діаграм прецедентів, класів, послідовностей та компонентів | 4 |  |
|  | Повторення теоретичного матеріалу змістовного модуля 2 | 4 |  |
|  | Розв’язок практичних завдань до ЗМ2. Опис ієрархії класів, інкапсуляція, поліморфізм | 4 |  |
|  | Розв’язок практичних завдань до ЗМ2. Перевантаження функцій та операцій. | 4 |  |
|  | Розв’язок практичних завдань до ЗМ2.  | 4 |  |
|  | Повторення теоретичного матеріалу змістовного модуля 3 | 4 |  |
|  | Розв’язок практичних завдань до ЗМ3. Використання класів-шаблонів бібліотеки STL. | 4 |  |
|  | Розв’язок практичних завдань до ЗМ3. Управління потоковим введенням-виведенням | 3 |  |
|  |  **Всього** | **35** |  |
|  | ***Разом***  | ***165*** |  |

**8. Індивідуальні завдання**

|  |  |  |
| --- | --- | --- |
| № | Назва теми | Кількість годин |
| денна | заочна |
|  | **Індивідуальне завдання. РГР. Реалізація спадковості.**  |  |  |
|  | Розробка батьківського класу та класу-нащадку | 4 |  |
|  | Розробка програмного інтерфейсу з користувачем | 4 |  |
|  | Тестування програмного забезпечення | 4 |  |
|  | Побудова діаграми прецедентів, опис прецеденту | 4 |  |
|  | Побудова та опис діаграми класів | 4 |  |
|  | Побудова та опис діаграми компонентів | 4 |  |
|  |  **Всього** | **24** |  |
|  |  |  |  |
|  | **Індивідуальне завдання. Модуль 2. Курсова робота.**  |  |  |
|  | Дослідження та опис предметної області, розробка функцій програми, розробка прецедентів. | 4 |  |
|  | Розробка концептуальної моделі предметної області та діаграми класів. | 4 |  |
|  | Розробка діаграми компонентів. | 4 |  |
|  | Розробка математичних постановок задач. | 4 |  |
|  | Розробка ієрархії класів програмних об’єктів. | 8 |  |
|  | Розробка програмного інтерфейсу користувача. | 8 |  |
|  | Реалізація методів класів програмних об’єктів. | 8 |  |
|  | Перевірка працездатності програмного забезпечення: тестування та налагодження програми. | 6 |  |
|  | Побудова UML-діаграм: прецедентів, класів, послідовностей, компонентів. | 6 |  |
|  | Опис UML-діаграм, програмного інтерфейсу користувача, тестових прикладів роботи програми. | 4 |  |
|  |  **Всього** | **56** |  |
|  | ***Разом***  | ***80*** |  |

Теми курсових робіт:

1. Розробка програми для обліку посібників з навчальних дисциплінах.
2. Розробка програми калькуляції вартості страв на основі даних про кулінарні рецепти.
3. Розробка програми продажу квитків на поточні авіарейси.
4. Розробка програми роботи з відомостями про маршрути громадського транспорту.
5. Розробка програма – утилізатора файлів.
6. Розробка програми для підтримки процесу курсового проектування.
7. Розробка програми ведення ділового щоденника.
8. Розробка програми ведення обліку земельних ділянок населених пунктів.
9. Розробка програми ведення обліку учбового навантаження викладачів кафедри (“Вчений секретар ”).
10. Розробка програми ведення розрахунків за комунальні платежі ЖЕКа.
11. Розробка програми графічного редактора зі збереженням зображення, списку об’єктів та їх параметрів у файлі.
12. Розробка програми для ведення довідника про випускників університету.
13. Розробка програми для контролю виконання студентами програми певної дисципліни.
14. Розробка програми для обліку пацієнтів ветеринарної клініки.
15. Розробка програми каталогізації носіїв інформації.
16. Розробка програми обліку відвідування занять студентами.
17. Розробка програми обліку зайнятості авто в агенції по прокату автомобілів.
18. Розробка програми обліку замовлень в агенції таксі.
19. Розробка програми обліку ліків в аптеці.
20. Розробка програми обліку міжміських та міжнародних телефонних переговорів.
21. Розробка програми обліку постояльців в готелі.
22. Розробка програми обліку спортивних змагань.
23. Розробка програми обліку товарів, що зберігаються на складах.
24. Розробка програми продажу путівок туристичної агенції.
25. Розробка програми роботи з об’єктами вулично-дорожньої мережі міста.
26. Розробка програми формування меню ресторану.
27. Розробка програми формування програми телепередач на каналах.
28. Розробка програми формування сітки мовлення телевізійного каналу.

**9. МЕТОДИ НАВЧАННЯ**

При викладанні навчальної дисципліни використовуються словесний, інформаційно-ілюстративний, наочний та практичний, проблемний та пошуковий методи навчання із застосуванням лекцій, задач, ситуаційних завдань, моделювання конкретних ситуацій, комплексних розрахункових завдань, реферативних оглядів, провокаційних вправ і запитань, ділових ігор, мозкових атак.

**10. МЕТОДИ контролю**

Контрольні заходи передбачають проведення вхідного (за необхідності), поточного, модульного та семестрового контролю.

Вхідний, поточний, модульний контроль здійснюється під час проведення практичних та індивідуальних занять з викладачем.

Семестровий контроль виконується за окремим графіком, складеним деканатом факультету.

Засоби контролю засвоєння матеріалу курсових робіт – представлення та захист роботи чи проекту.

**10. Розподіл балів, які отримують студенти**

|  |  |
| --- | --- |
| Поточне оцінювання | Сума |
| Модуль № 1 | Модуль № 2 | Модуль № 3 | Індивідуальне завдання |
| 20 | 30 | 20 | 30 | 100 |

|  |  |  |
| --- | --- | --- |
| Поточне оцінювання | Підсумковий тест (екзамен) | Сума |
| Змістовий модуль № 1 | Змістовий модуль № 2 | Змістовий модуль №3 |
| ~10 | ~30 | ~30 | ~30 | 100 |

**Курсова робота**

|  |  |  |  |
| --- | --- | --- | --- |
| ЗМ1.ООАналіз та ООПроектування | ЗМ2. Розробка ООПрограми | ЗМ3. Оформлення та захист роботи | Сума |
| ~30 | ~30 | ~40 | 100 |

**Шкала оцінювання: національна та ECTS**

| Сума балів за всі види навчальної діяльності | ОцінкаECTS | Оцінка за національною шкалою |
| --- | --- | --- |
| для екзамену, курсового проекту (роботи), практики | для заліку |
| 90 – 100 | **А** | відмінно  | зараховано |
| 82-89 | **В** | добре  |
| 74-81 | **С** |
| 64-73 | **D** | задовільно  |
| 60-63 | **Е**  |
| 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 |

**11. Методичне забезпечення**

1. Електронний навчальний курс Гончаренко Т.А. Методичні вказівки до виконання лабораторних робіт з дисципліни «Об’єктно-орієнтоване програмування» К.:КНУБА , 2018 [Електронний документ]. Режим доступу [http://do2.knuba.edu.ua/course/view.php?id=55](http://do2.knuba.edu.ua/course/view.php?id=30)

**13. Рекомендована література**

**Базова**

1. Глинський Я.М., Анохін В.Є., Ряжська В.А. C# і C# Builder: Навч.посібник. - 3-е видання. Львів: Деол, СПД Глинський, 2006. - 198 с.
2. Ларман К. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку: Учебное пособие. - 3-е издание М.:Вильямс, 2004. - 736 с.
3. Страуструп Б. Язык программирования С#. – М.:Бином-Пресс, 2008. - 1098

**Допоміжна**

1. Буч Г., Объектно-ориентированный анализ и проектирование с примерами приложений на С#, 3-е изд. \ Пер. с англ.- М.:ООО «И.Д. Вильямс», 2008 г. – 721 с. ил.
2. Шилдт Г. С#: базовый курс .-М.:Вильямс, 2008 .-620с.
3. Шилдт Г. Самоучитель С#. - СПб.: БХВ-Петербург, 2003. – 687 с.

**14. Інформаційні ресурси**

1. <http://library.knuba.edu.ua/> <http://org2.knuba.edu.ua/course/view.php?id=55>