

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВАСИЛЯ СТЕФАНІКА»

ФАКУЛЬТЕТ МАТЕМАТИКИ ТА ІНФОРМАТИКИ

# **European Credit Transfer System ECTS – Інформаційний пакет**

**Галузь знань:**

**12 «Інформаційні технології»**

**Спеціальність:**

**122 «Комп'ютерні науки»**

**Івано-Франківськ  
2017**

## **1. ОРГАНІЗАЦІЙНА СТРУКТУРА ФАКУЛЬТЕТУ**

### **1) Адреса:**

Головний корпус ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»,

76018, Україна, м. Івано-Франківськ, вул. Шевченка, 57, ауд. 317.

### **2) Контактні телефони:**

тел.: (0342) 59-60-04, (0342) 59-61-08

### **3) Інформація про склад деканату:**

Декан факультету:

д.ф.-м.н., проф. Пилипів Володимир Михайлович

тел.: (0342) 59-60-04

e-mail: [dekanat\\_mif@pu.if.ua](mailto:dekanat_mif@pu.if.ua)

Заступник декана:

к.ф.-м.н., доц. Соломко Андрій Васильович

тел.: (0342) 59-61-08

e-mail: [andrii.solomko@pu.if.ua](mailto:andrii.solomko@pu.if.ua)

Диспетчер: Череватий Володимир Миколайович

тел.: (0342) 59-61-08.

## **2. ОРГАНІЗАЦІЙНА СТРУКТУРА КАФЕДР ФАКУЛЬТЕТУ**

### **Кафедра інформатики**

### **1) Адреса:**

Головний корпус ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»,

76018, Україна, м. Івано-Франківськ, вул. Шевченка, 57, ауд. 314.

**2) Контактні телефони:** тел.: (0342) 59-60-86

### **3) Інформація про склад кафедри:**

1. Превисокова Наталія Володимирівна – к.т.н., доцент, завідувач кафедри;

2. Петришин Любомир Богданович – д.т.н., професор;

3. Дудка Ольга Михайлівна – к.пед.н., доцент;

4. Горелов Віталій Олевтинович – к.т.н., доцент;

5. Семаньків Марія Василівна – к.т.н., доцент;

6. Ровінський Віктор Анатолійович – к.т.н., доцент;

7. Власій Олеся Орестівна – к.т.н., доцент;

8. Ляш Юрій Юрійович – к.т.н., доцент;

9. Гейко Орест Ярославович – старший викладач;

10. Максимець Василь Зіновійович – викладач;

11. Василик Ольга Ярославівна – старший лаборант кафедри.

## **Кафедра інформаційних технологій**

### **1) Адреса:**

Головний корпус ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»,

76018, Україна, м. Івано-Франківськ, вул. Шевченка, 57, ауд. 319.

### **2) Контактні телефони:**

тел.: (0342) 59-60-58

### **3) Інформація про склад кафедри:**

1. Філевич Петро Васильович – д.ф.-м.н., професор, завідувач кафедри;
2. Козленко Микола Іванович – к.т.н., доцент;
3. Лазарович Ігор Миколайович – к.т.н., доцент;
4. Дрінь Богдан Михайлович – к.пед.н., доцент;
5. Ткачук Валерій Михайлович – к.ф.-м.н., доцент;
6. Хрущ Олеся Зеновіївна – к.е.н., викладач;
7. Гарпуль Оксана Зеновіївна – к.ф.-м.н., викладач;
8. Дутчак Марія Степанівна – старший лаборант кафедри.

## **Кафедра алгебри і геометрії**

### **1) Адреса:**

Головний корпус ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»,

76018, Україна, м. Івано-Франківськ, вул. Шевченка, 57, ауд. 405.

### **2) Контактні телефони:**

тел.: (0342) 59-60-16

### **3) Інформація про склад кафедри:**

1. Никифорчин Олег Ростиславович – д.ф.-м.н., професор, завідувач кафедри;
2. Артемович Орест Дем'янович – д.ф.-м.н., професор;
3. Пилипів Володимир Михайлович – д.ф.-м.н., професор;
4. Собкович Роман Іванович – к.ф.-м.н., доцент;
5. Мазуренко Наталія Іванівна – к.ф.-м.н., доцент;
6. Гаврилків Володимир Михайлович – к.ф.-м.н., доцент;
7. Копорх Катерина Миколаївна – к.ф.-м.н., викладач;
8. Глушак Інна Дмитрівна – викладач;
9. Ліщинський Іван Іванович – викладач;
10. Семак Галина Адамівна – старший лаборант кафедри.

## **Кафедра диференціальних рівнянь і прикладної математики**

### **1) Адреса:**

Головний корпус ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»,

76018, Україна, м. Івано-Франківськ, вул. Шевченка, 57, ауд. 315.

### **2) Контактні телефони:**

тел.: (0342) 59-60-27

### **3) Інформація про склад кафедри:**

1. Заторський Роман Андрійович – д.ф.-м.н., професор, завідувач кафедри;
2. Василюшин Павло Богданович – к.ф.-м.н., доцент;
3. Гой Тарас Петрович – к.ф.-м.н., доцент;
4. Казмерчук Анатолій Іванович – к.ф.-м.н., доцент;
5. Костишин Любов Павлівна – к.ф.-м.н., викладач;
6. Мазуренко Віктор Володимирович – к.ф.-м.н., доцент;
7. Махней Олександр Володимирович – к.ф.-м.н., доцент;
8. Череватий Володимир Миколайович – старший лаборант кафедри.

## **Кафедра математичного та функціонального аналізу**

### **1) Адреса:**

Головний корпус ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»,

76018, Україна, м. Івано-Франківськ, вул. Шевченка, 57, ауд. 302.

### **2) Контактні телефони:**

тел.: (0342) 59-60-50

### **3) Інформація про склад кафедри:**

1. Копач Михайло Іванович – к.ф.-м.н., доцент, завідувач кафедри;
2. Загороднюк Андрій Васильович – д.ф.-м.н., професор;
3. Шарин Сергій Володимирович – к.ф.-м.н., професор;
4. Малицька Ганна Петрівна – к.ф.-м.н., доцент;
5. Федак Іван Васильович – к.ф.-м.н., доцент;
6. Соломко Андрій Васильович – к.ф.-м.н., доцент;
7. Івасюк Іван Ярославович – к.ф.-м.н., викладач;
8. Василюшин Тарас Васильович – к.ф.-м.н., викладач;
9. Марцінків Марія Василівна – к.ф.-м.н., викладач;
10. Кравців Вікторія Василівна – к.ф.-м.н., викладач;
11. Музика Марія Ігорівна – старший лаборант кафедри.

## **Кафедра статистики і вищої математики**

### **1) Адреса:**

Головний корпус ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»,

76018, Україна, м. Івано-Франківськ, вул. Шевченка, 57, ауд. 406.

### **2) Контактні телефони:**

тел.: (0342) 59-60-47

### **3) Інформація про склад кафедри:**

1. Осипчук Михайло Михайлович – к.ф.-м.н., доцент, завідувач кафедри;

2. Кульчицька Наталія Володимирівна – к.пед.н., доцент;

3. Никифорчин Ірина Володимирівна – к.е.н., доцент;

4. Слободян Світлана Ярославівна – к.ф.-м.н., доцент;

5. Шевчук Роман Володимирович – к.ф.-м.н., доцент;

6. Кашуба Григорій Іванович – викладач;

7. Осипчук Ольга Миколаївна – старший лаборант кафедри.

## **3. ПЕРЕЛІК СПЕЦІАЛЬНОСТЕЙ**

На кафедрах здійснюється підготовка фахівців за спеціальностями (освітній рівень бакалавра):

- 014.04 Середня освіта (математика) з ліцензованим обсягом прийому 35 осіб денної форми навчання та 25 осіб заочної форми навчання;

- 014.09 Середня освіта (інформатика) з ліцензованим обсягом прийому 20 осіб денної форми навчання та 40 осіб заочної форми навчання;

- 111 Математика з ліцензованим обсягом прийому 40 осіб денної форми навчання;

- 112 Статистика з ліцензованим обсягом прийому 30 осіб денної форми навчання;

- 113 Прикладна математика з ліцензованим обсягом прийому 40 осіб денної форми навчання;

- 121 Інженерія програмного забезпечення обсягом прийому 30 осіб денної форми навчання;

- 122 Комп'ютерні науки обсягом прийому 40 осіб денної форми навчання та 20 осіб заочної форми навчання.

На кафедрах здійснюється підготовка фахівців за спеціальностями (освітній рівень магістра):

- 014.04 Середня освіта (математика) з ліцензованим обсягом прийому 15 осіб денної форми навчання та 15 осіб заочної форми навчання;

- 014.09 Середня освіта (інформатика) з ліцензованим обсягом прийому 15 осіб денної форми навчання та 15 осіб заочної форми навчання;
- 111 Математика з ліцензованим обсягом прийому 15 осіб денної форми навчання;
- 112 Статистика з ліцензованим обсягом прийому 15 осіб денної форми навчання;
- 113 Прикладна математика з ліцензованим обсягом прийому 15 осіб денної форми навчання;
- 122 Комп'ютерні науки обсягом прийому 40 осіб денної форми навчання та 20 осіб заочної форми навчання.

#### **4. УМОВИ НАВЧАННЯ**

##### **1) Загальна інформація про практику:**

Практика студентів є невід'ємною складовою частиною освітньо-професійної програми підготовки фахівців, основним завданням якої є якість практичної підготовки випускника за освітньо-кваліфікаційним рівнем бакалавр. За період практики у студентів закладаються основи досвіду професійної діяльності, практичних умінь і навичок.

Наскрізна програма практик студентів спеціальностей 122 «Комп'ютерні науки» передбачає проведення практики в три етапи:

на 1 курсі – навчальна практика протягом двох тижнів 2-го навчального семестру;

на 2 курсі – обчислювальна практика протягом 2 тижнів 4-го навчального семестру;

на 3 курсі – виробнича практика протягом 6 тижнів 8-го навчального семестру;

Зміст практики визначається її робочою програмою.

Метою навчальної (обчислювальної) практики є формування у студентів навичок самостійної розробки алгоритмів та програм на одній із алгоритмічних мов програмування, відладка та оформлення програм.

Виробнича практика служить для формування в студентів професійного вміння, навичок, самостійності в прийнятті рішень на конкретній роботі в реальних виробничих умовах шляхом виконання в умовах виробництва різних обов'язків, властивих майбутній професійній, організаційній та громадській діяльності; опанування методами та засобами роботи з реальними об'єктами інформатизації в організаціях та установах.

База практики. Під базою практики слід розуміти усі підприємства, організації та установи різних галузей, на яких студенти університету можуть проходити практику. Ці підприємства, організації та установи (в багатьох

випадках) закріплені за університетом як бази практик спільними наказами (договорами, розпорядженнями, листами або іншими документами).

Місцем проведення практики є сучасні підприємства, організації; різні установи: освіти, науки, державного управління; навчально-практичні центри. Як бази практики можуть використовуватися навчально-виробничі та наукові підрозділи університету (кафедра інформатики, Центр інформаційних технологій, Центр дистанційної освіти та контролю знань).

З базами практики (підприємствами, організаціями, установами будь-яких форм власності) Прикарпатський національний університет імені Василя Стефаника завчасно укладає договори на її проведення за визначеною формою. Тривалість дії договорів може визначатися на період конкретного виду практики або до п'яти років.

Базами практики студентів можуть бути державні, акціонерні та комерційні установи і організації. Зокрема, базами практики є: головне управління статистики в Івано-Франківській області; ВАТ “Укртелеком”; Національний банк України; відділення Промінвест банку; центр по нарахуванню і виплаті пенсій і допомог; вище професійне училище №21; центр дистанційного навчання та контролю знань, центр інформаційних технологій, лабораторія моделювання інфосистем Прикарпатського національного університету імені Василя Стефаника; технікум електронних приладів; науково-виробниче підприємство “Бескид”; виробничо-комерційна фірма “Інтеп”; товариство з обмеженою відповідальністю “СЛОТ”; облводгосп та ряд інших установ, організацій та навчальних закладів.

Кожна база практики має сучасну обчислювальну техніку, необхідне програмне забезпечення, а також робочі місця для практикантів. На кожній базі практики студент має можливість ознайомитись з обчислювальною технікою, з програмно-математичним забезпеченням, що використовується на підприємстві, а також з прикладними задачами, що розв'язуються на підприємстві. Наприклад: створенням автоматизованих систем управління для різних галузей промисловості; встановлення нового програмного забезпечення, що покращує роботу підприємства; редагування і використання спеціальних баз даних; розробка програмного забезпечення нових операцій, зокрема введення пластикових карток; вміння здійснювати комп'ютерну обробку статистичної інформації; здійснювати порівняння розроблених раніше програм з самостійно розробленою програмою та цілий ряд інших програм.

Всі бази практики відповідають як вимогам програм практики, так і освітньо-кваліфікаційним характеристикам підготовки спеціалістів по спеціальності «Комп'ютерні науки».

## 2) Коротка інформація про матеріально-технічну базу:

Існує потужна та якісна матеріально-технічна база для організації навчального процесу: аудиторний фонд, комп'ютерні класи, спеціалізовані лабораторії, бібліотечно-інформаційний центр тощо.

Аудиторно-лабораторний фонд факультету складає 6800 кв.м. у складі 8 лекційних аудиторій, 4 з них обладнані мультимедійною апаратурою, 18 аудиторій для проведення практичних і лабораторних занять, лабораторії інформаційних мережних технологій, 6 спеціалізованих комп'ютерних лабораторій центру інформаційних технологій, навчального комплексу та лабораторії CISCO, Internet-центр, бібліотека.

До послуг студентів кафедри інформатики – «Центр Інформаційних Технологій» (ЦІТ) загальною площею 600 кв.м. зі 200 сучасними персональними комп'ютерами, приєднаними до мережі Інтернет. Для підготовки студентів напрямку «Комп'ютерні науки» передбачено 90 комп'ютерів.

Обладнання, устаткування та програмне забезпечення спеціалізованих комп'ютерних лабораторій, які забезпечують виконання начального плану зі спеціальності «Комп'ютерні науки»

№ п/п	Найменування комп'ютерної лабораторії, її площа	Найменування дисципліни за навчальним планом	Модель і марка персональних комп'ютерів, їх кількість	Назви пакетів прикладних програм (в тому числі ліцензованих)	Можливість доступу до Інтернет, наявність каналів доступу (так/ні)
1	2	3	4	5	6
1	Лаб.307 Лабораторія ЦІТ, 60 м <sup>2</sup>	Програмування Організація та обробка електронної інформації	Intel I5, 19	Open Office, MikTeX, Visual C++ 2010 Express	так
2.	Лаб.309 Лабораторія ЦІТ, 60 м <sup>2</sup>	Серверне програмування Паралельні та розподілені обчислення Програмування та підтримка веб-застосунків	Intel I5, 19	MySQL, Visual C++ 2010 Express	так
3	Лаб.320 Лабораторія ЦІТ, 60 м <sup>2</sup>	Обробка зображень та мультимедіа Паралельні обчислення та засоби їх	Intel I5, 19	Open MPI Project, SciLab	так



		реалізації Теорія програмування			
--	--	---------------------------------------	--	--	--

## 5. ОСНОВНІ МЕТОДИ НАВЧАННЯ, ЩО ВИКОРИСТОВУЮТЬСЯ В НАВЧАЛЬНОМУ ПРОЦЕСІ, СПОСОБИ ОЦІНЮВАННЯ ЗНАНЬ

– *Пояснювально-ілюстративний метод або інформаційно-рецептивний* (студенти одержують знання на лекції, з навчальної або методичної літератури, через екранний посібник в "готовому" виді)

– *Репродуктивний метод (репродукція - відтворення)* (діяльність студентів носить алгоритмічний характер, тобто виконується за інструкціями, приписаннями, правилами в аналогічних, подібних з показаним зразком ситуаціях. Організовується діяльність студентів алгоритмічного характеру за кількарізним відтворенням засвоєних знань, при цьому використовуються різноманітні вправи, лабораторні, практичні роботи, програмований контроль, різні форми самоконтролю.

– *Метод проблемного викладу* (формулюється пізнавальне завдання на основі різних джерел і засобів, показуються способи рішення поставленого завдання. Діяльність студентів базується на розкритті системи доказів, порівнянні різних точок зору та підходів в процесі наукового пошуку).

– *Частково-пошуковий, або евристичний, метод* (організація активного пошуку рішення висунутих у навчанні (або сформульованих самостійно) пізнавальних завдань, пошук рішення відбувається під керівництвом педагога, або на основі евристичних програм і вказівок).

– *Дослідницький метод* (проводиться аналіз матеріалу, постановки проблем і завдань і короткого усного або письмового інструктажу студентів, студенти самостійно вивчають літературу, джерела, ведуть спостереження й виміри й виконують інші дії пошукового характеру).

### Способи оцінювання знань:

- усний;
- письмовий;
- тестовий
- програмований;
- практична перевірка;
- самоконтроль і самооцінка.

### Форми контролю:

- індивідуальна перевірка;
- фронтальна перевірка;
- консультації (перевірка конспектів першоджерел, самостійної роботи над допоміжною літературою, та відпрацювання пропущених лекцій, семінарських занять тощо);
- контрольні роботи;

– підсумковий контроль (заліки, іспити, курсові роботи, дипломні та магістерські роботи, державні іспити).

Рівень якості підготовки магістра, спеціаліста та бакалавра визначається за комплексною системою оцінювання, що поєднує Європейську кредитно-трансферну систему (ЄКТС) (за шкалою «А», «В», «С», «D», «E», «FX», «F»), національну (за шкалою «відмінно», «добре», «задовільно», «незадовільно») та систему оцінювання ДВНЗ «Прикарпатський національний університет імені Василя Стефаника» (за 100-бальною шкалою).

Оцінювання знань, умінь і навичок студентів з навчальної дисципліни при підсумковому контролі проводиться відповідно до наступних загальних рекомендацій:

Рівень навчальних досягнень	Бали	Критерії навчальних досягнень студентів
1	2	3
Високий	90-100	Студент вільно володіє навчальним матеріалом, самостійно творчо виконує усі поставлені завдання. Самостійно знаходить джерела пізнання та використовує їх для вдосконалення виконаних завдань.
Достатній	80-89	Студент вільно володіє вивченим матеріалом, застосовує його на практиці у стандартних ситуаціях, вміє давати обґрунтовані відповіді на поставлені питання.
	70-79	Студент може застосовувати знання в стандартних ситуаціях (завдання за докладними інструкціями), з допомогою викладача аналізувати отримані результати.
Середній	60-69	Студент може з сторонньою допомогою виправити допущені неточності, повністю виконує за методичними вказівками і додатковими поясненнями практичне завдання, виявляє знання та розуміння основних положень предмету.
	50-59	Студент відтворює значну частину поданого матеріалу, з допомогою викладача може виконати фрагменти поставлених завдань.

Початковий	26-49	Студент з допомогою викладача відтворює основну частину викладеного матеріалу, може виконати без пояснень деякі фрагменти практичних завдань.
	1-25	Студент володіє навчальним матеріалом на початковому рівні, відповідає на поставлені питання тільки “так” або “ні”, з допомогою викладача може виконати базові операції при розв’язанні практичних завдань.



## 7. НАВЧАЛЬНИЙ ПЛАН СПЕЦІАЛЬНОСТІ 122 «КОМП'ЮТЕРНІ НАУКИ»

№	Шифр за ОПН	НАЗВА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ, ПРАКТИКИ	Розподіл за семестрами				Кількість кредитів ЕКТС	Кількість годин										Розподіл годин на тижень за курсами і семестрами										
			Екзамени	Зачітки	Практика	Курсова робота		завдяки обсягу				з наск.			Самостійна робота			I курс	II курс				III курс	IV курс				
								лекції	практики	семінари	лабораторні	індивідуальні	1	2	3	4	5		6	7	8							
			3	4	5	6		7	8	9	10	11						12				13	14	15	16	17	18	19
<b>V. ПЛАН НАВЧАЛЬНОГО ПРОЦЕСУ</b>																												
<b>1. Нормативна частина</b>																												
<b>1.1. Цивільно-гуманітарної та соціально-економічної підготовки</b>																												
	ГС.1	Українська мова (за проф. спрявл.)		3			12	360	120	56	16	48	0	0	0	240	0	2	2	0	0	0	0	0	2	3		
	ГС.2	Історія України		2			3	90	30	14	16	16				60												
	ГС.3	Історія української культури		8			3	90	30	14		16				60										3		
	ГС.4	Філософія		7			3	90	30	14		16				60										2		
	<b>1.2. Цивільно-гуманітарної, природничо-наукової підготовки</b>																											
	ФП.01	Математичний аналіз	1,2,3				39	1170	390	170	220	0	0	0	0	780	10	8	7	2	0	0	0	0	0	0		
	ФП.02	Алгебра і геометрія	1				12	360	120	50	70					240	3	3	3									
	ФП.03	Дискретна математика	3				6	180	60	26	34					120												
	ФП.04	Теорія ймовірностей та математична		2			6	180	60	26	34					120												
	ФП.05	Математична логіка та теорія	2	1			9	270	90	40	50					180	3	3										
	ФП.06	Диференціальні рівняння		4			3	90	30	14	16					60										2		
	<b>1.3. Цивільно-професійної та практичної підготовки</b>																											
	ПП.1	Іноземна мова		1,2			6	180	60		60					120	2	2										
	ПП.2	Алгоритми, структури і аналіз даних		3			3	90	30	14			16			60												
	ПП.3	Архітектура обчислювальних систем	5				3	90	30	14			16			60										2		
	ПП.4	Компютерні мережі	6				3	90	30	14			16			60										2		
	ПП.5	Вступ у спеціальність		1			3	90	30	14			16			60												
	ПП.6	Обробка зображень та музичне дід		3			3	90	30	14			16			60										2		





## 8. АНОТАЦІЇ НАВЧАЛЬНИХ ДИСЦИПЛІН

### 1. Цикл гуманітарної та соціально-економічної підготовки

<i>Предмет:</i>	<b>Ділова українська мова</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	2 рік, 3 семестр
<i>Анотація:</i>	Дисципліна вивчає основні вимоги до мовних засобів ділового стилю в математиці; логічна завершеність формування думки, чіткість висловлювань, послідовність і точність викладу думки; деякі складні випадки усного і писемного мовлення і письмо; чітке дотримання прийнятих в суспільстві норм ділового спілкування.
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Історія України</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	1 рік, 2 семестр
<i>Анотація:</i>	Основним змістом є вивчення становлення і розвитку суспільства в Україні, основних історичних періодів, через які пройшло українство, їх особливостей, закономірностей. Дисципліна вивчає буття українського народу на всій його етнічній території із найдавніших часів до сьогодення. Дисципліна «Історія України» передбачає ґрунтовний аналіз походження українського народу, становлення та еволюцію держави, роль різних верств у житті суспільства. Особлива увага звертається на періоди існування самостійної української державності, аналізуються причини її втрати. У процесі вивчення Історії України характеризуються закономірності та особливості політичних, соціально-економічних і національно-духовних змін в Україні нового часу.
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Філософія</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	4 рік, 8 семестр
<i>Анотація:</i>	Формування системи знань про психіку особистості як



	найвищу цінність суспільства; усвідомлення сутності механізмів психічних процесів, станів, якостей особистості як підвалин її формування в процесі виховання, навчання та освіти.
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Історія української культури</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	4 рік, 7 семестр
<i>Анотація:</i>	Формування системи знань з історії української культури. Програма та тематичний план дисципліни орієнтовані на глибоке та ґрунтовне засвоєння студентами предмету та завдань історії української культури, проблем генезису та динаміки української культури, основних напрямів української художньої культури, феномену української культури в контексті європейської культурної традиції, сучасних проблем розвитку української культури.
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Фізична культура</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	1-2 рік, 1-4 семестр
<i>Анотація:</i>	Фізична культура у загальнокультурній та професійній підготовці студентів; соціально-біологічні основи фізичної культури; основи здорового способу та стилю життя; оздоровчі системи та спорт (теорія, методика, практика); професійно-прикладна фізична підготовка студентів; фізичні вправи залежно від професії.
<i>Форми контролю:</i>	Немає

<i>Предмет:</i>	<b>Політологія</b>
<i>Статус:</i>	Вибіркова
<i>Рік, семестр</i>	3 рік, 5 семестр
<i>Анотація:</i>	Вивчення сутності, історії, теорії та методології політичної діяльності і поведінки. Вміти орієнтуватися в основних світових політичних

	школах, концепціях і напрямках, знати і вміти давати характеристику українським політичним вченням, мати уявлення про сутність політичного життя, політичних відносин та процесів, про об'єкт і суб'єкт політики, знати права людини і громадянина, суть і значення політичних систем і режимів у житті держави і суспільства, мати уявлення про процеси міжнародного політичного життя, геополітичну обстановку, місце, роль і статус України в сучасному політичному світі, володіти навичками політичної культури, вміти застосовувати політичні знання в своїй професійній і громадській діяльності.
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Економіка</b>
<i>Статус:</i>	Вибіркова
<i>Рік, семестр</i>	3 рік, 6 семестр
<i>Анотація:</i>	Предметом вивчення дисципліни є основоположні економічні категорії, економічні закони та принципи функціонування економічних систем; економічні відносини, господарчі механізми та дії людей, спрямовані на ефективне господарювання в умовах обмежених економічних і природних ресурсів; особливості ринкових перетворень економіки України; сучасні процеси глобалізації економічного життя людства.
<i>Форми контролю:</i>	Залік

## **2. Цикл фундаментальної та природничо-наукової підготовки**

<i>Предмет:</i>	<b>Математичний аналіз</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	1-2 рік, 1-3 семестр
<i>Анотація:</i>	Мета курсу полягає у наданні майбутнім спеціалістам знань у галузі сучасного математичного аналізу. Курс передбачає засвоєння таких основних змістовних модулів: теорія дійсного числа, теорія границь послідовностей і функцій, неперервність функції однієї змінної, диференціювання функції однієї змінної, дослідження функції за допомогою похідних,

	первісна, невизначений інтеграл, визначений інтеграл та його застосування, невластні інтеграли, диференціальне числення функцій багатьох змінних, неявні функції, числові ряди, функціональні ряди і послідовності, інтеграли залежні від параметра, подвійні і потрійні інтеграли, криволінійні та поверхневі інтеграли, ряди Фур'є.
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Алгебра і геометрія</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	1 рік, 1 семестр
<i>Анотація:</i>	Метою курсу є формування компетенцій майбутнього фахівця в галузі алгебри що сприятимуть здатності до абстрактного та конкретного аналізу, вмінню оцінювати зв'язки між різними математичними об'єктами. Сприяння розвитку логічного та аналітичного мислення студентів. Курс передбачає засвоєння таких основних змістовних модулів: основні алгебраїчні структури, поле комплексних чисел, кільце поліномів, матриці і визначники, системи лінійних рівнянь, векторні простори, квадратичні форми. Метою курсу є формування компетенцій майбутнього фахівця в галузі геометрії, що сприятимуть здатності до абстрактного та конкретного аналізу, розширенню можливостей геометричних уявлень та вмінню оцінювати зв'язки між різними математичними об'єктами. Сприяння розвитку логічного та аналітичного мислення студентів.
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Дискретна математика</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	2 рік, 3 семестр
<i>Анотація:</i>	Мета курсу: розкрити роль та місце дискретної математики в курсі математики; показати основні взаємозв'язки курсу з іншими розділами математики;

	виявити застосування дискретної математики в науковому і прикладному напрямках; забезпечити умови для підготовки кваліфікованого спеціаліста з питань дискретної математики; сформувати вміння і навички самостійного аналізу і застосування положень даної дисципліни до подальшого вивчення різних областей математики.
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Диференціальні рівняння</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	2 рік, 4 семестр
<i>Анотація:</i>	Курс передбачає засвоєння таких основних тем: предмет та методи теорії звичайних диференціальних рівнянь, звичайні диференціальні рівняння першого порядку, розв'язані відносно похідної, звичайні диференціальні рівняння першого порядку, не розв'язані відносно похідної, звичайні диференціальні рівняння вищих порядків, лінійні диференціальні рівняння вищих порядків, системи звичайних диференціальних рівнянь, елементи теорії стійкості, рівняння з частинними похідними першого порядку.
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Теорія алгоритмів і математична логіка</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	1 рік, 1,2 семестр
<i>Анотація:</i>	У результаті вивчення дисципліни „Теорія алгоритмів і математична логіка” студент повинен знати: основні поняття і теореми з теорії алгоритмів та математичної логіки; вміти: виконувати логічні операції над висловлюваннями; з'ясовувати різними методами чи є формула логіки тавтологією, суперечністю, виконуваною; подавати булеві функції у вигляді ДДНФ, ДКНФ, полінома Жегалкіна, з'ясовувати чи є система булевих функцій СДНФ, ТДНФ, МДНФ булевих функцій різними методами, будувати релейно-контактні схеми, складати схеми НАМ,

	оцінювати складність алгоритмів.
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Теорія ймовірності та математична статистика</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	1 рік, 2 семестр
<i>Анотація:</i>	<p>Мета дисципліни: оволодіння студентами теоретико-методичними та практичними навичками дослідження випадкових явищ і процесів, засвоєння ними методів обробки результатів масових спостережень.</p> <p>У процесі освоєння матеріалу курсу студент повинен оволодіти основними поняттями теорії ймовірностей та математичної статистики, розуміти основні теоретико-ймовірнісні та статистичні закономірності, навчитися застосовувати ці поняття та закономірності при розв'язанні практичних задач. Вивчення курсу дає можливість студентам набути навички розв'язання теоретичних та обчислювальних задач в ситуаціях, де мають місце випадкові впливи та спостереження над достатньо великою кількістю об'єктів досліджуваної сукупності.</p>
<i>Форми контролю:</i>	Залік

### **3. Цикл професійної та практичної підготовки**

<i>Предмет:</i>	<b>Іноземна мова</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	1 рік, 1,2 семестр
<i>Анотація:</i>	<p>Мета дисципліни: розвинути у студентів важливі навички і вміння у розумінні іноземної мови як такої, що має власні фонетичну, граматичну, лексико-семантичну системи; виробити навички у розмовній практиці і на письмі, надати базові знання, пояснити основні граматичні, фонетичні, синтаксичні особливості мови, надати і закріпити на практиці певний лексичний об'єм, а також розвинути вміння висловлювати думку іноземною мовою з використанням базової лексики зі спеціальності.</p>
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Алгоритми і структури даних, аналіз даних</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	2 курс, 1 семестр
<i>Анотація:</i>	Вивчення курсу «Алгоритми і структури даних» має за мету сформувати у студентів знання, вміння та навички, які необхідні для створення ефективних алгоритмів та їх використання при розробці програмних додатків. Програмою дисципліни передбачається вивчення наступних питань: класифікація алгоритмів, оцінки складності алгоритмів, методи розробки алгоритмів, абстрактні типи даних, статичні та динамічні, лінійні та нелінійні структури даних: табличні структури, графи, дерева та алгоритми їх обробки, алгоритми сортування та пошуку.
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Архітектура обчислювальних систем</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	3 рік, 5 семестр
<i>Анотація:</i>	Метою дисципліни є вивчення основних понять архітектури сучасних комп'ютерів та комп'ютерних систем, компонентами апаратних засобів, механізмами передавання і управління інформацією, проектуванням та експлуатацією обчислювальних систем масового обслуговування. Зміст дисципліни. Побудова та архітектура ЕОМ. Історія ЕОМ. Організація та принцип дії комп'ютера. Моделі та методи ОС. Моделі навантаження. Класифікація ЕОМ. Множинність потоків. Класифікація Хендлера та Базу. Методи розпаралелювання. Архітектури обробки ЕОМ. Архітектури ЕОМ послідовної, мультипрограмної, конвеєрної обробки.
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Комп'ютерні мережі</b>
<i>Статус:</i>	Нормативна

<i>Рік, семестр</i>	3 рік, 6 семестр
<i>Анотація:</i>	<p>Мета дисципліни: ознайомлення студентів з теоретичними основами функціонування, проектування, документування та обслуговування комп'ютерних мереж, а також з сучасним станом базового апаратного (маршрутизатори, комутатори тощо) та програмного (операційна система) забезпечення.</p> <p>Завдання дисципліни: ознайомити студентів з основними принципами комп'ютерних мереж, навчити створювати локальні мережі, розвинути навички використання глобальних мереж.</p>
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Вступ в спеціальність</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	1 рік, 1 семестр
<i>Анотація:</i>	В курсі навчальної дисципліни перспективи розвитку комп'ютерних наук та інформаційних технологій. вивчаються постановки задач і простіші методи алгоритмізації; простіші методи збору, аналізу і обробки інформації.
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Обробка зображень та мультимедіа</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	2 курс, 3 семестр
<i>Анотація:</i>	<p>Вивчення курсу «Обробка зображень та мультимедіа» має за мету сформуванню у студентів знання, вміння та навички, які необхідні для ефективного використання графічних зображень та мультимедіа технологій в проектуванні та розробці програмних додатків. Програмою дисципліни передбачається вивчення двох розділів: обробка зображень, засоби мультимедіа. Студенти будуть знайомитися з графічним дизайном, колірними моделями, растровою та векторною графікою, кривими Без'є, тривимірною графікою. При вивченні засобів мультимедіа розглядатимуться питання: керування мультимедійним проектом, комп'ютерна анімація, подання та засоби обробки</p>

	звукової інформації, подання та засоби обробки відеоінформації.
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Організація та обробка електронної інформації</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	1 курс, 1 семестр
<i>Анотація:</i>	Вивчення курсу «Організація та обробка електронної інформації» має за мету сформуванню у студентів компетентності, необхідні для створення і використання моделей подання електронної інформації у різних форматах, використання різноманітних методів та засобів обробки електронної інформації, ефективного використання засобів сучасних інформаційних технологій у своїй майбутній професійній діяльності. Студенти повинні оволодіти системною сукупністю знань і вмінь, яка містить практичні навички роботи з комп'ютером, розуміння і знання загальних принципів його побудови та функціонування, вміння використовувати сучасні програмні засоби загального і спеціального призначення (текстові та графічні редактори, електронні таблиці, математичні пакети) для збереження, обробки, пошуку та передачі різних видів інформації.
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Операційні системи</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	3 рік, 6 семестр
<i>Анотація:</i>	Мета опанування дисципліни в контексті підготовки фахівців певної спеціальності: сформулювати у слухачів необхідний обсяг теоретичних знань про появу, генезис та перспективи розвитку комп'ютерних мереж, рівні мережевої моделі взаємодії відкритих систем та протоколи, які їх об'єднують, моделі теорії масового обслуговування в комп'ютерних мережах, практичних навичок використання локальних, муніципальних, корпоративних, національних та



	глобальних комп'ютерних мереж, їх інтеграцію з засобами телекомунікацій, методів маршрутизації та комутації, технологій та засобів передачі даних, налаштування мережевих операційних систем, мереж з заданою якістю обслуговування, програмно-керованих комп'ютерних мереж та прикладних додатків, які реалізують різні мережеві сервіси.
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Системне програмування</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	2-3 рік, 4-5 семестр
<i>Анотація:</i>	Під час вивчення дисципліни розглядаються наступні питання: поняття програмного забезпечення, вимоги до нього та класифікація, програмна апаратна модель процесора IA-32, синтаксис мови асемблера, системи числення, типи даних асемблера, задання операндів та методи адресації в асемблері, поняття консолі та функції Win32 API по роботі з нею, функції перетворення даних, класифікація команд асемблера, команди пересилання даних, арифметичні команди і команди передачі управління, логічні команди та команди зсуву, команди обробки ланцюгів, логічні команди та команди зсуву, складні типи даних, та їх підтримка засобами асемблера, процедури на мові асемблера, правила передачі аргументів, робота зі стеком, структура математичного співпроцесора, правила роботи з дійсними числами, представлення чисел з плаваючою комою, правила виконання операцій, команди роботи з математичним співпроцесором, структура програми з віконним інтерфейсом, методи повідомлень Win32, поняття ресурсу системи Win32 та методи роботи з ним, структура програми на основі діалогу, види та використання стандартних діалогів, робота з графічними об'єктами в системі Win32 API, функції Win32 API по роботі з файлами, розмежування доступу, методи управління пам'яттю в системі Win32, поняття багатопоточності та методи

	синхронізації потоків в системі Win32, динамічні бібліотеки та методи роботи з ними, програмування співпроцесора, використання розширень команд MMX та SSE, структура програм та методи операційної системи по створенню програм з мультидокументним інтерфейсом.
<i>Форми контролю:</i>	Залік, екзамен

<i>Предмет:</i>	<b>Програмування І</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	1 рік, 1-2 семестр
<i>Анотація:</i>	<p>Мета: навчити студентів основам мови C++: використанню виразів, операторів, масивів та рядків, вказівників, функцій, структур, введенню-виведенню у консоль, файлового введенню-виведенню. Забезпечити вироблення практичних навичок роботи з класами, вказівниками, посиланнями та операторами динамічного розподілу пам'яті. Забезпечити засвоєння операцій перевантаження функцій, конструкторів та конструкторів копіювання. Студенти повинні набути практичних навичок реалізації успадкування, віртуальних функцій та поліморфізму, шаблонів та обробки виняткових ситуацій.</p> <p>Завдання викладання навчальної дисципліни: навчити студентів основам застосування мов C/C++. Забезпечити вироблення практичних навичок в проектуванні та програмуванні комплексних програмних продуктів загального призначення.</p> <p>Зміст дисципліни. Основи C++. Вирази мови C++. Оператори мови C++. Масиви і рядки. Вказівники. Функції. Структури, об'єднання, списки та оператор typedef. Введення-виведення у консоль.</p> <p>Операції з файлами та робота з пам'яттю.</p> <p>Файлове введення-виведення. Масиви, вказівники, посилання і оператори динамічного розподілу пам'яті. Препроцесор. Обробка виняткових ситуацій. Основи системи введення-виведення. Файлова система.</p>
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Програмування II</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	2 рік, 3-4 семестр
<i>Анотація:</i>	<p>Метою курсу є навчити студентів основам мови C#: використанню виразів, операторів, масивів та рядків, вказівників, функцій, структур, введенню-виведенню у консоль, файлового введенню-виведенню. Забезпечити вироблення практичних навичок роботи з класами, вказівниками, посиланнями та операторами динамічного розподілу пам'яті. Забезпечити засвоєння операцій переважання функцій, конструкторів та конструкторів копіювання. Студенти повинні набути практичних навичок реалізації успадкування, віртуальних функцій та поліморфізму, шаблонів та обробки виняткових ситуацій.</p> <p>Зміст дисципліни. Принципи побудови сучасного багатопотокового програмного забезпечення. Концепція MVC. Основи побудови мобільних додатків. Загальна характеристика мови C# (історія, платформи, фундаментальні властивості, віртуальна машина і байт-код, програмні моделі, безпека). Написання найпростіших консольних прикладних програм. Типи даних та основні операції з ними. Основні управляючі конструкції. Масиви. Рядки. Основні сервісні функції. Основні ООП характеристики C#. Наслідування і поліморфізм. Абстрактні класи й інтерфейси. Внутрішні класи. Виключення. Багатопотоковість.</p>
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Паралельні та розподілені обчислення</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	4 курс, 7 семестр
<i>Анотація:</i>	Мета викладання навчальної дисципліни полягає в: розкритті актуальних аспектів проведення

	<p>високоєфективних обчислень шляхом застосування алгоритмів багатопоточного, паралельного та розподіленого програмування; формуванні у студентів цілісної системи знань та вмінь з метою використання ними сучасних інформаційних технологій при розробці систем реального часу та систем математичної обробки інформаційних масивів.</p> <p>Завдання викладання навчальної дисципліни: навчити студентів здійснювати вибір типу обчислювального процесу (розподіленого, багатопоточного чи розпаралеленого), а також методології його реалізації.</p> <p>Зміст дисципліни. Програмування із розподіленими змінними. Задача паралельного програмування. Інтерактивний та рекурсивний паралелізм. Процеси і синхронізація. Неподільні дії та оператори очікування. Блокування і бар'єри. Семафори. Монітори. Реалізація семафорів та моніторів.</p> <p>Розподілене та синхронне паралельне програмування. Передача повідомлень. Віддалений виклик процедур і рандеву. Синхронні сокети. Асинхронні сокети. Моделі взаємодії процесів. Наукові обчислення. Мови, компілятори та інструментальні засоби.</p>
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Платформи корпоративних інформаційних систем</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	4 рік, 7 семестр
<i>Анотація:</i>	<p>Мета: набуття студентами необхідних теоретичних знань та практичних навичок для розробки програмного забезпечення на платформі JavaEE. Завдання викладання навчальної дисципліни: навчити студентів основам застосування мов JAVA EE. У результаті вивчення навчальної дисципліни студент повинен знати: принципи побудови сучасного багатошарового програмного забезпечення. Концепцію MVC. Основи побудови серверних додатків.</p> <p>Java Platform, Enterprise Edition — обчислювальна корпоративна платформа Java.</p>

	<p>Платформа надає API та середовище для розробки і виконання корпоративного програмного забезпечення, включаючи мережеві та веб сервіси, інші масштабовані, розподілені додатки. Java EE розширює стандартну платформу Java (Java SE - Java Standart Edition).</p> <p>Зміст дисципліни. Основи Java EE, огляд концепцій і можливостей; масштабовуваність; багаторівнева архітектура; архітектура Java EE; інструменти розробки (на прикладі середовища розробки IntelliJ IDEA); основи Java Server Faces; HTTP Servlets; Java Server Pages (JSP); Faces Servlet; сервлетні фільтри; HTTP сесії та JSF; настроювання JSF; обробка виключень; компонентна модель і MVC у застосуванні до Java Server Faces; message Driven EJB – призначення і застосування. Java Message Service API; основи Object Relational Mapping; Веб-сервіси та Java EE, Java Web Services (JWS).</p>
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Програмування та підтримка веб-застосунків</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	4 рік, 7 семестр
<i>Анотація:</i>	<p>Мета викладання навчальної дисципліни полягає в: розкритті актуальних аспектів побудови динамічних web-додатків; формуванні цілісної системи знань та вмінь з метою використання сучасних інформаційних технологій при розробці систем з віддаленим доступом; формуванні практичних навичок використання систем управління базами даних для web-розробок; формуванні практичних навичок застосування web-програмування для розв’язування прикладних задач; встановленні предметних зв’язків даної навчальної дисципліни з іншими нормативними та варіативними дисциплінами.</p> <p>Завдання викладання навчальної дисципліни: навчити студентів використовувати основні оператори мови PHP, маніпулювати рядками за допомогою регулярних виразів, використовувати системи управління базами</p>

	<p>даних, зокрема MySQL, у повсякденній діяльності. Студенти повинні оволодіти ООП на PHP, вміти керувати сеансами, вирішувати задачі аутентифікації тощо.</p> <p>Зміст дисципліни. Основи PHP. Основи серверної мови створення сценаріїв – PHP. Масиви у PHP. Багатократне використання коду. Створення функцій. Збереження і відновлення даних. Робота з файлами. Використання бази даних MySQL за допомогою PHP. Об'єктно-орієнтоване програмування у PHP.</p> <p>Методи обробки даних . Задача аутентифікації. Керування сеансами. Використання функцій для роботи з мережею та протоколами. Маніпулювання рядками та регулярні вирази. Використання функцій PHP для роботи з XML.</p>
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Теорія програмування</b>
<i>Статус:</i>	Нормативна
<i>Рік, семестр</i>	4 курс, 8 семестр
<i>Анотація:</i>	<p>Мета курсу Теорія програмування – сформулювати у студентів компетенції, необхідні для ефективного використання теоретичних положень програмування при розробці прикладного і системного програмного забезпечення.</p> <p>Програмою дисципліни передбачається вивчення основних поняття теорії програмування, природних та формальних мов, підходів до формалізації мов специфікацій та програмування, синтактики: формальні мови та граматики, автоматних формалізмів сприйняття мов, розв'язні та нерозв'язні проблеми теорії формальних мов, семантики програм та методи їх подання, рекурсії в мовах програмування, теорії найменшої нерухомої точки та її застосування.</p>
<i>Форми контролю:</i>	Екзамен

#### **4. Цикл дисциплін самостійного вибору навчального закладу**

<i>Предмет:</i>	<b>Цифрова техніка</b>
-----------------	------------------------

<i>Статус:</i>	Вибіркова
<i>Рік, семестр</i>	1 рік, 2 семестр
<i>Анотація:</i>	У курсі подаються інформаційні, математичні, та логічні основи цифрової техніки; схемотехніка цифрових елементів комбінаційних та послідовнісних пристроїв; принципи побудови і функціонування елементів, типових вузлів і пристроїв цифрової і мікропроцесорної техніки; програмні засоби проектування цифрових та мікропроцесорних систем.
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Мікропрограмування</b>
<i>Статус:</i>	Вибіркова
<i>Рік, семестр</i>	2 рік, 3 семестр
<i>Анотація:</i>	Метою курсу є отримання студентами необхідного рівня знань, необхідного для побудови спеціалізованих обчислювальних процесів на базі програмованих логічних інтегральних схем різних типів, з використанням мови програмування Verilog
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Мікропроцесорні системи</b>
<i>Статус:</i>	Вибіркова
<i>Рік, семестр</i>	2 курс, 4 семестр
<i>Анотація:</i>	Метою є отримання студентами необхідного рівня знань для самостійної творчої праці в галузі цифрової електроніки та мікропроцесорної техніки, та отримання необхідного рівня знань які дозволяють проектувати системи автоматичного керування на базі мікроконтролерів.
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Теорія інформації та кодування</b>
<i>Статус:</i>	Вибіркова
<i>Рік, семестр</i>	2 курс, 4 семестр
<i>Анотація:</i>	Теорія інформації та кодування – це наукова дисципліна, яка вивчає способи передачі і зберігання інформації найбільш надійним та економічним

	методом. Метою викладання дисципліни є оволодіння знаннями по теорії передачі інформації по каналах зв'язку для контролю і управління технологічними процесами. Положення цієї теорії можуть також використовуватися при розробці будь-яких цифрових пристроїв у різних галузях науки і техніки.
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Перетворення форми інформації</b>
<i>Статус:</i>	Вибіркова
<i>Рік, семестр</i>	3 курс, 5 семестр
<i>Анотація:</i>	У курсі подаються основи перетворення форми інформації, методи та засоби аналого-цифрового та цифро-аналогового перетворення, теорія та принципи побудови аналого-цифрових і цифро-аналогових перетворювачів. Зміст дисципліни. Сигнали та їх види. Квантування. АЦП. Класифікація АЦП. Сфера застосування АЦП. Види АЦП. Цифро-аналогові перетворювачі та їх види.
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Цифрова обробка інформації</b>
<i>Статус:</i>	Вибіркова
<i>Рік, семестр</i>	3 курс, 6 семестр
<i>Анотація:</i>	Метою дисципліни є вивчення основ цифрової обробки інформації, сучасних методів та засобів цифрової обробки інформації. Методи обробки цифрових сигналів. Перетворення Фур'є. Спектри, їх класифікація. Дискретне перетворення Фур'є. Дискретні ортогональні перетворення. Кореляційний аналіз. Засоби цифрової обробки інформації.
<i>Форми контролю:</i>	Екзамен

<i>Предмет:</i>	<b>Безпека інфотехнологій</b>
<i>Статус:</i>	Вибіркова
<i>Рік, семестр</i>	2 курс, 4 семестр



<i>Анотація:</i>	Метою дати студентам знання в галузі криптографії та криптоаналізу. Дисципліна знайомить з основними принципами роботи криптографічних систем, математичними моделями джерел інформації, поняттями теоретичної та практичної секретності. Конкретні типи алгоритмів шифрування та криптографічних перетворень розглядаються у відповідності з їх класифікацією на класичні схеми, системи потокового шифрування, системи блочного шифрування та системи захисту інформації з відкритим ключем.
<i>Форми контролю:</i>	Екзамен

### **5. Цикл дисциплін вільного вибору студента**

<i>Предмет:</i>	Спеціалізація Комп'ютеризовані системи та мережі
<i>Статус:</i>	Вибіркова
<i>Рік, семестр</i>	3-4 рік, 5-8 семестр
<i>Анотація:</i>	Студент в результаті вивчення дисциплін даної спеціалізації володітиме знаннями, вміннями і навиками, які необхідні для роботи систем комп'ютерної математики та математичних розрахунків в цілому; оволодіє навичками використання систем комп'ютерної математики для автоматизації розрахунків і моделювання; Студент отримає базові теоретичні знання і практичні навички з ефективного кодування повідомлень і завадозахищеного обміну даними, необхідних для застосування при розробці та експлуатації розосереджених інформаційних систем. Студент вмітиме формувати вимоги до програмного забезпечення, працювати індивідуально і в команді, проявляти вміння та навички, необхідні для професійного, особистісного розвитку; вивчить психологічні аспекти ефективної комунікації та сформує навички ефективного управління командою та роботі в команді.
<i>Форми контролю:</i>	Залік

<i>Предмет:</i>	<b>Спеціалізація Системний інжиніринг</b>
-----------------	---

<i>Статус:</i>	Вибіркова
<i>Рік, семестр</i>	3-4 рік, 5-8 семестр
<i>Анотація:</i>	<p>Студент в результаті вивчення дисциплін даної спеціалізації розроблятиме індивідуальне програмне забезпечення (на замовлення) та адаптуватиме пакети програм до специфічних потреб користувачів; використовуватиме спеціалізовані інтегральні мікросхеми при побудові пристроїв і засобів автоматизації технологічних об'єктів; проектуватиме системи управління на основі мікропроцесорних систем. володітиме знаннями, вміннями і навиками, які необхідні для розроблення, впровадження і використання систем оброблення інформації алгоритмічними методами з використанням комп'ютерної техніки, математичних методів і алгоритмів у різних галузях науки і народного господарства. Студент розроблятиме індивідуальне програмне забезпечення (на замовлення) та адаптуватиме пакети програм до специфічних потреб користувачів; використовуватиме спеціалізовані інтегральні мікросхеми при побудові пристроїв і засобів автоматизації технологічних об'єктів; проектуватиме системи управління на основі мікропроцесорних систем. Студент вмітиме формувати вимоги до програмного забезпечення, працювати індивідуально і в команді, проявляти вміння та навички, необхідні для професійного, особистісного розвитку; вивчить психологічних аспектів ефективної комунікації та сформує навички ефективного управління.</p>
<i>Форми контролю:</i>	Залік

## **9. ФАХОВЕ СПРЯМУВАННЯ ТА КВАЛІФІКАЦІЙНІ ВИМОГИ ДО ФАХІВЦІВ СПЕЦІАЛЬНОСТІ 122 «КОМП'ЮТЕРНІ НАУКИ»:**

За період навчання студенти отримують неперервну наскрізну підготовку з сучасних технологій програмування, методів проектування програмного забезпечення, розробки Інтернет-додатків та Web-сайтів, ефективних структур баз даних, мережевих технологій, адміністрування розподілених комп'ютерних систем. Крім того, студенти набувають знання з аналізу і синтезу інтелектуальних систем різного призначення, у тому числі здатних самонавчатися, набувають навички розробки систем для захисту інформації, розпізнавання образів, штучних нейронних мереж, Web-дизайну та інше. Випускники працюють спеціалістами з розробки програмного забезпечення, інформаційної безпеки, керівниками проектів, системними адміністраторами мереж, розробниками систем Internet-комерції, Web-дизайнерами, викладачами навчальних закладів.

Фахівець з спеціальності 122 «Комп'ютерні науки» повинен:

1. Вміти аналізувати предметну область і давати опис предмету дослідження – природних, техногенних, економічних та соціальних об'єктів та процесів інформатизації, – використовуючи методи збору, аналізу та обробки інформації.
2. Вміти з'ясувати особливості предмету дослідження на базі методів системного аналізу та кібернетики.
3. Вміти будувати інформаційні моделі предмету дослідження: описувати його суттєві параметри та змінні величини, виокремлювати його вхідні та вихідні параметри та встановлювати причинно-наслідкові зв'язки між ними.
4. Вміти розробляти математичні моделі об'єктів і процесів інформатизації, використовуючи методи формального опису систем, математичної логіки, моделювання та системного аналізу на основі результатів проведе-них досліджень.
5. Вміти розробляти детерміновані та стохастичні моделі об'єктів та процесів інформатизації, використовуючи методи математичного моделювання, вміти ідентифікувати їх параметри.
6. Вміти аналітично досліджувати властивості математичних моделей (коректність, повнота, складність, точність моделей; існування, єдиність і стійкість розв'язків, тощо).
7. Вміти розробляти та досліджувати математичні моделі оптимізації, прогнозування, оптимального керування та прийняття рішень для об'єктів та процесів інформатизації.

8. Вміти аналізувати адекватність моделі предмету дослідження та вносити необхідні корективи до розробленої моделі.
9. Вміти розробляти концепції комп'ютерної реалізації моделі предмету дослідження на основі алгоритмічного, структурного, об'єкто-зорієнтованого, компонентного, аспектно-орієнтованого, сервіс-орієнтованого, мультиагентного та інших сучасних підходів, використовувати концепції паралельної обробки інформації.
10. Вміти визначати оптимальний варіант концепції комп'ютерної реалізації моделі в процесі аналізу вимог на різних етапах її життєвого циклу та розробляти концепцію відповідної комп'ютеризованої системи.
11. Вміти використовувати, розробляти та досліджувати математичні методи та алгоритми обробки даних (статистичні, алгебраїчні, комбінаторні, теоретико-інформаційні та інші).
12. Вміти використовувати, розробляти та досліджувати алгоритми розв'язування задач моделювання об'єктів і процесів інформатизації, задач оптимізації, прогнозування, оптимального керування та прийняття рішень, тощо.
13. Вміти використовувати, розробляти та досліджувати алгоритми функціонування комп'ютеризованих систем методами неперервної, дискретної математики, математичної логіки тощо.
14. Вміти розробляти та використовувати математичні методи та алгоритми обчислювальної геометрії.
15. Вміти оцінювати складові ефективності алгоритмів функціонування комп'ютеризованих систем.
16. Вміти узагальнювати досвід побудови адекватних математичних моделей природних, техногенних та соціальних процесів.
17. Вміти оформлювати отримані результати у вигляді науково-технічної документації, звітів та статей.
18. Вміти збирати та систематизувати інформацію про предмет проектування за допомогою методів добування даних та знань.
19. Вміти формулювати бізнес-вимоги, вимоги користувача, системні вимоги, функціональні, нефункціональні, експлуатаційні вимоги, антивимоги, тощо до комп'ютеризованих систем.
20. Вміти враховувати та застосовувати вимоги чинних державних та міжнародних стандартів щодо виконання робіт з проектування комп'ютеризованих систем.

21. Вміти специфікувати вимоги (давати формалізований опис) з використанням мов специфікацій та формулюванням критеріїв перевірки.
22. Вміти аналізувати специфікації на узгодженість, повноту та несуперечливість, реалізованість, пріоритетність, необхідність та однозначність використання, можливість перевірки тощо.
23. Вміти забезпечувати трасування вимог, будувати прототип системи, розробляти тести.
24. Вміти використовувати програмні методи та засоби підтримки побудови та аналізу специфікацій.
25. Вміти будувати абстрактну архітектуру (логічну модель) комп'ютеризованої системи.
26. Вміти відокремлювати основні архітектурні компоненти, описувати їх функції, зв'язки (інтерфейси) між ними та правила, що регламентують ці зв'язки в централізованій та розподіленій архітектурі.
27. Вміти обирати адекватний архітектурний стиль та необхідні архітектурні шаблони.
28. Вміти проводити верифікацію архітектурних рішень та оцінювати їх ефективність за допомогою прототипів, імітаційних моделей, логіко-математичних доведень тощо.
29. Вміти визначати апаратну платформу та програмне середовище, що відповідають обраній архітектурі.
30. Вміти використовувати програмні методи та засоби підтримки архітектурного проектування.
31. Вміти документувати прийняті архітектурні рішення.
32. Вміти використовувати основні парадигми проектування програмного забезпечення: структурну, об'єкто-орієнтовану, компонентну, аспектно-орієнтовану, сервіс-орієнтовану, мультиагентну, розподілену тощо для розробки проекту комп'ютеризованої системи.
33. Володіти методами опису основних понять програмування, вміти задавати семантику та синтаксис конструкцій мов програмування.
34. Володіти мовами моделювання програмного забезпечення комп'ютеризованих систем (UML, GPSS тощо).
35. Вміти планувати життєвий цикл програмного забезпечення та розробляти модель керування ресурсами.

36. Вміти проводити аналіз дефектів, помилок та ризиків у життєвому циклі програмного забезпечення, обирати та формувати вимоги до характеристик якості.
37. Вміти розробляти фізичну модель (компонентну модель, модель розгортання тощо) програмного забезпечення.
38. Вміти проектувати тести для перевірки окремих компонентів програмного забезпечення.
39. Вміти застосовувати алгоритми та методи захисту інформації у проектах комп'ютеризованих систем.
40. Вміти використовувати програмні (автоматизовані) методи та засоби підтримки проектування програмного забезпечення.
41. Вміти розробляти концептуальну модель СКБД на основі збору, аналізу і формулювання вимог до даних.
42. Вміти розробляти логічну модель СКБД на основі порівняльного аналізу моделей подання даних: реляційних, ієрархічних, об'єкто-орієнтованих, мережних, розподілених, багатовимірних та інших.
43. Вміти розробляти фізичну модель СКБД (компонентну модель, модель розгортання тощо) на основі визначення особливостей зберігання даних, методів доступу і т.п.
44. Для реляційних БД вміти розробляти таблиці, використовувати методи редагування даних, використовувати методи реляційної алгебри, вміти проводити нормалізацію відношень тощо.
45. Вміти класифікувати інтелектуальні системи та розробляти їх концептуальні моделі на основі аналізу предметної області, використовуючи методи добування та структурування знань.
46. Володіти моделями подання знань (формально-логічні, фреймові, продукційні, семантичні тощо).
47. Володіти методами логічного виведення (дедуктивні, індуктивні, семантичні тощо).
48. Вміти розробляти фізичну модель інтелектуальної системи (компонентна модель, модель розгортання тощо) на основі визначення особливостей зберігання даних, методів доступу і т.п.
49. Володіти теорією нечітких множин, лінгвістичних змінних, моделями та методами виведення для систем з нечіткою логікою.
50. Вміти використовувати поширені протоколи Інтернет при проектуванні комп'ютеризованих систем.
51. Вміти розробляти модель та структуру Інтернет-серверу, використовуючи технології розподілених застосувань.

52. Вміти проектувати інформаційні веб-ресурси з інтеграцією зовнішніх даних і програмних продуктів за допомогою технологій Java, Perl, PHP тощо.
53. Володіти основами веб-дизайну.
54. Вміти розробляти концепцію побудови локальних комп'ютерних мереж на основі стандартних протоколів і інтерфейсів (HTTP, FTP, TCP/IP, WAP, Wi-Fi тощо). Вміти вибирати топологію комп'ютерної мережі.
55. Вміти планувати мережну інфраструктуру, програмне і апаратне забезпечення, фізичне розміщення користувачів, ділення мережі на сегменти, мережні протоколи тощо. Вміти розробляти логічну і фізичну структуру локальної комп'ютерної мережі, топологію структурованих кабельних систем.
56. Вміти вибирати програмне забезпечення комп'ютерних мереж за допомогою нормативно-довідкової інформації, використовуючи процедури аналізу типових проектних рішень.
57. Володіти методами захисту інформації в локальних мережах.
58. Володіти методами цифрового подання та обробки графічної, звукової та відео інформації; знати та вміти обирати формати графічних, звукових та відео файлів; володіти засобами їх перетворення, методами підготовки мультимедійних презентацій тощо.
59. Володіти основами комп'ютерної графіки, вміти використовувати моделі передачі кольору, моделі візуалізації інформації (растрові, векторні, фронтальні, та інші), формати графічних файлів.
60. Володіти методами проектування динамічних графічних об'єктів (двовимірних та тривимірних) для програмних систем комп'ютерної графіки (ігри, відео кліпи тощо).
61. Вміти використовувати у своєму житті норми етики та правила коректної поведінки по відношенню до людей, тварин та довкілля, методики та засоби організації здорового способу життя.
62. Вміти дотримуватись законів екології довкілля та безпеки життєдіяльності, норм діючого чинного законодавства.
63. Бути творчою та креативною особистістю, використовувати системний підхід для розв'язування професійних завдань, наполегливо досягати мету та якісно виконувати роботу у професійній сфері.
64. Вміти працювати в колективі розробників та організувати його роботу з проектування та розроблення комп'ютеризованої системи та після проектної її супроводу з врахуванням етичних, філософських

- та релігійних позицій, історії та культури суспільства, особливостей психології поведінки членів колективу.
65. Володіти методами міжособистісних комунікацій, дотримуючись норм толерантності, та вміти адаптувати-ся до різних практичних умов для ефективної праці в колективі
  66. Вміти планувати роботи з проектування та розроблення комп'ютеризованої системи, оцінювати потрібний час та витрати праці, оформлювати технічне завдання та бізнес-план програмістського проекту, з урахуванням бюджету проекту, штатного розпису, кваліфікації розробників тощо.
  67. Вміти організувати роботу колективу з дотриманням техніки безпеки та гігієни праці, попередження або зменшення рівня аварійності, рівня ймовірного пошкодження обладнання.
  68. Вміти планувати та керувати роботами з проектування та розроблення комп'ютеризованої системи та післяпроектного її супроводу.
  69. Вміти приймати управлінські рішення на основі чинного законодавства, з урахуванням їх впливу на права членів колективу розробників та розумінням особистої відповідальності щодо їх наслідків.
  70. При прийнятті управлінських рішень вміти враховувати особливості культури, етики, віросповідання, психології особистості членів колективу тощо
  71. Володіти засобами менеджменту проектів
  72. Вміти забезпечувати захист персоналу та об'єкту господарювання згідно з діючими нормативними документами в умовах надзвичайних ситуацій.
  73. Вміти дотримуватись законів екології довкілля та безпеки життєдіяльності.
  74. Володіти основами програмування та мовами різних рівнів (машинними, асемблерними, високого рівня, проблемно та предметно орієнтованими).
  75. Володіти загальними принципами функціонування та архітектури комп'ютерних систем та основами операційних систем.
  76. Володіти системним програмним забезпеченням, знати принципи роботи компіляторів, інтерпретаторів, компонувальників, налагоджувачів, утиліт, систем управління файлами, драйверів тощо.
  77. Вміти використовувати прикладне програмне забезпечення, зокрема пакети прикладних програм, офісні, навчальні, мультимедійні,



- графічні, типографські системи, системи керування вмістом (contentmanagement), порталом, підприємством тощо.
78. Володіти методами збереження, обробки та редагування інформації в системах керування базами даних.
  79. Вміти використовувати інтелектуальні системи, бази знань.
  80. Володіти основами Інтернет-технологій і методами адміністрування Інтернет-серверів, розробки та підтримки інформаційного порталу Інтернет, веб-інтерфейсів.
  81. Знати основи комп'ютерних мереж, володіти технологіями побудови та адміністрування мереж.
  82. Володіти технологіями та методами захисту інформації в комп'ютеризованих системах та мережах.
  83. Вміти розробляти програмне забезпечення комп'ютеризованої системи з використанням технологій програмування, заснованими на структурній, об'єктно-орієнтованій, компонентній, аспектно-орієнтованій, сервіс-орієнтованій, мультиагентній, розподіленій, логічній та інших парадигмах.
  84. Вміти розробляти програмне забезпечення для локальних комп'ютерних мереж, Інтернет-серверів, інформаційних порталів Інтернет, веб-інтерфейсів.
  85. Володіти основами технологій розробки баз даних та систем керування базами даних, інтелектуальних систем, баз знань тощо.
  86. Вміти використовувати мультимедійні, графічні системи, пакети графічних бібліотек для розробки програмних систем комп'ютерної графіки (ігри, відео кліпи тощо).
  87. Володіти технологіями та методами розроблення програмного забезпечення для захисту інформації в комп'ютеризованих системах та мережах.
  88. Вміти контролювати якість виконання проектних робіт та розроблення комп'ютеризованої системи колективом розробників із врахуванням діючих державних, міжнародних, професійних та корпоративних стандартів.
  89. Вміти проводити контроль відповідності розробленої комп'ютеризованої системи встановленим замовником вимогам.
  90. Вміти контролювати правильність роботи програмно-го забезпечення розробленої комп'ютеризованої системи за допомогою тестування на різних рівнях (модульному, інтеграційному, системному, тощо).

91. Вміти розробляти тести (тестові набори, сценарії та коди) для контролю комп'ютеризованих систем.
92. Вміти контролювати та перевіряти правильність експлуатації встановленого програмного забезпечення комп'ютеризованої системи згідно чинних норм та стандартів.
93. Вміти контролювати та здійснювати моніторинг працездатності системного та прикладного програмного забезпечення в умовах експлуатації комп'ютеризованих систем.
94. Вміти контролювати дотримання вимог безпеки праці, санітарно-гігієнічних вимог на робочому місці.
95. Вміти аналізувати повідомлення спеціалізованих інформаційних видань та фірм – виробників програмного забезпечення про тенденції у створенні нових інформаційних технологій, вміти робити відповідний прогноз щодо їх розвитку та можливих застосувань.
96. Знати основи апаратної частини комп'ютерів (персональних, спеціалізованих комп'ютерів, кластерів тощо) та володіти навичками з обслуговування комп'ютерної техніки на рівні користувача.
97. Володіти навичками обслуговування інфраструктури комп'ютерних мереж, відповідного програмного і апаратного забезпечення.
98. Володіти методами та сучасними програмними засобами для налагодження програм та програмних комплексів.
99. Вміти налагоджувати та обслуговувати системне програмне забезпечення та операційні системи, встановлені у сучасних установах, підприємствах та фірмах.
100. Вміти налагоджувати та обслуговувати прикладне програмне забезпечення, зокрема пакети прикладних програм, офісні, мультимедійні, графічні, навчальні системи, системи керування вмістом (contentmanagement), порталом, підприємством тощо.
101. Володіти технологіями налагодження та обслуговування та експлуатації програмного забезпечення комп'ютерних мереж.
102. Вміти налагоджувати та обслуговувати програмне забезпечення Інтернет-серверів, інформаційних порталів Інтернет, веб-інтерфейсів.
103. Вміти налагоджувати та обслуговувати комерційні системи керування базами даних, інтелектуальні системи, бази знань, супроводжувати експлуатацію розроблених програмних продуктів в організаціях та на підприємствах.

## **10. ОПИС РЕЗУЛЬТАТІВ НАВЧАННЯ (КОМПЕТЕНЦІЇ)**

### **Загальні компетентності**

1. Здатність до абстрактного мислення, аналізу та синтезу.
2. Здатність застосовувати знання у практичних ситуаціях.
3. Знання та розуміння предметної області та розуміння професійної діяльності.
4. Здатність спілкуватися іноземною мовою.
5. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
6. Здатність генерувати нові ідеї (креативність).
7. Здатність розробляти та управляти проектами.
8. Здатність приймати обґрунтовані рішення.
9. Здатність оцінювати та забезпечувати якість виконуваних робіт.

### **Професійні компетентності**

- 1) Здатність презентувати результати науково-дослідницької діяльності, готувати наукові публікації, доповідати на наукових конференціях, симпозіумах.
- 2) Здатність до математичного та логічного мислення, формулювання та досліджування математичних моделей, зокрема дискретних математичних моделей, обґрунтування вибору методів і підходів для розв'язування теоретичних і прикладних задач в галузі комп'ютерних наук, інтерпретування отриманих результатів.
- 3) Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: структурного, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління.
- 4) Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.
- 5) Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.
- 6) Здатність розробляти засоби реалізації інформаційних технологій (методичні, інформаційні, алгоритмічні, технічні й програмні).
- 7) Здатність використовувати сучасні технології проектування в розробці алгоритмічного та програмного забезпечення ІСТ.

8) Здатність виконувати розрахунки основних параметрів системи та її елементів.

9) Здатність тестувати та налагоджувати об'єкти що розроблюються.

10) Здатність виконувати розрахунки основних параметрів системи та її елементів.

## **Результати навчання**

Здобувати систематичні знання в галузі комп'ютерних наук, аналізувати проблеми з точки зору сучасних наукових парадигм, осмислювати і робити обґрунтовані висновки з наукової і навчальної літератури та результатів експериментів.

Професійно спілкуватись державною та іноземними мовами, розробляти державною та іноземними мовами документацію на системи, продукти і сервіси інформаційних технологій, читати, розуміти та застосовувати технічну документацію українською та іноземними мовами в професійній діяльності.

Використовувати технології та інструментарії пошукових систем, методи інтелектуального аналізу даних і текстів, здійснювати опрацювання, інтерпретацію та узагальнення даних.

Застосовувати у роботі міжнародні стандарти з оцінки якості програмного забезпечення, управління та обслуговування ІТ сервісів, моделі оцінки зрілості процесів розробки ПЗ.

Аналізувати проблемні ситуації, ставити собі певні цілі щодо розв'язання професійних задач і свідомо добиватися їх реалізації, вибирати шлях для майбутніх дій, визначати засоби, потрібні для досягнення мети, приймати рішення.

Ефективно використовувати сучасний математичний апарат в професійній діяльності для розв'язання задач теоретичного та прикладного характеру в процесі аналізу, синтезу та проектування інформаційних систем за галузями.

Формулювати мету управління організаційно-технічною та економічною системами, формувати систему критеріїв якості управління, будувати математичну модель задачі, вибирати та застосовувати відповідний метод розв'язування задачі оптимізації, знаходити її оптимальний розв'язок, коригувати модель й розв'язок на основі отриманих нових знань про задачу й операцію.

Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для

реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук, створювати надійне та ефективне програмне забезпечення.

## 11. ГЛОСАРІЙ

### Програмування

Алгоритм – набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв'язання задачі за скінченну кількість дій; система правил виконання дискретного процесу, яка досягає поставленої мети за скінченний час.

Змінна – в імперативному програмуванні – поійменована, або така, яку адресують іншим способом, область пам'яті, адресу якої можна використовувати для здійснення доступу до даних, що знаходяться у змінній (тобто за даною адресою пам'яті).

Константа в програмуванні – спосіб адресації даних, зміна яких даною програмою не передбачається або забороняється.

Блок-схема – графічне представлення алгоритму, зображають у вигляді послідовності пов'язаних між собою функціональних блоків, кожен з яких відповідає виконанню однієї або кількох дій.

Адреса змінної – адреса першого байта ділянки пам'яті, зайнятого змінною.

Вказівник (англ. pointer або англ. reference) — тип даних в комп'ютерних мовах програмування. Значення вказівника посилається на інше значення, що записане будь-де у пам'яті комп'ютера (фактично містить його адресу).

Ідентифікатор – лексична одиниця, яка використовується як ім'я елемента мови програмування; ім'я, що присвоюється даному елементові у вигляді послідовності латинських букв, цифр, спеціальних знаків. Ідентифікатор зазвичай починається з букви.

Цикл – різновид керуючої конструкції у високорівневих мовах програмування, призначеної для організації багаторазового виконання набору інструкцій (команд). Також циклом може називатися будь-яка багатократно виконувана послідовність команд, організована будь-яким чином (наприклад, із допомогою умовного переходу).

Команди переходу (анг. branch instructions) – різновид команд передачі управління в архітектурах систем команд ЕОМ, які змінюють послідовний порядок виконання програми (тобто перезавантажують лічильник команд адресою, записаною в самій команді переходу, а не збільшуючи його значення на довжину поточної команди). Команди переходу можуть бути умовними (conditional branch) та безумовними (unconditional branch).

Об'єктно-орієнтоване програмування (ООП) – одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Основу ООП складають три основні концепції: інкапсуляція, успадкування та поліморфізм.

Інкапсуляція – один з трьох основних механізмів об'єктно-орієнтованого програмування, суть якого полягає у тому, що об'єкт вміщує не тільки дані, але і правила їх обробки, оформлені в вигляді виконуваних фрагментів (методів). При цьому зазвичай доступ до стану об'єкта напряду заборонено, і ззовні з ним можна взаємодіяти через заданий інтерфейс (відкриті поля та методи), що дозволяє знизити зв'язність.

Успадкування – в об'єктно-орієнтованому програмуванні — механізм утворення нових класів на основі використання вже існуючих. При цьому властивості і функціональність батьківського класу переходять до класу нащадка (дочірнього).

Поліморфізм – концепція ООП, відповідно до якої використовують спільний інтерфейс для обробки даних різних спеціалізованих типів. На противагу поліморфізму, концепція мономорфізму вимагає однозначного зіставлення. У контексті об'єктно-орієнтованого програмування найпоширенішим різновидом поліморфізму є здатність екземплярів підкласу грати роль об'єктів батьківського класу, завдяки чому екземпляри підкласу можна використовувати там, де використовують екземпляри батьківського класу.

Клас – це спеціальна конструкція, яку використовують для групування пов'язаних змінних та функцій. При цьому згідно із термінологією ООП глобальні змінні класу (члени-змінні) називають полями (також властивостями або атрибутами), а члени-функції називають методами класу. Створений та ініціалізований екземпляр класу називають об'єктом класу. На основі одного класу можна створити безліч об'єктів, що відрізнятимуться один від одного своїм станом (значеннями полів).

Конструктор класу (від англ. constructor) – спеціальний метод класу, який автоматично викликається при створенні об'єкта. Призначення конструктора – встановити початковий стан об'єкта шляхом ініціалізації атрибутів об'єкта.

Конструктор копіювання – особливий конструктор, котрий використовують для створення нових об'єктів як копій існуючого об'єкта. Першим аргументом такого конструктора є посилання (константне або ні) на об'єкт того ж типу, що й тип об'єкта який конструюють.

Деструктор – в об'єктно-орієнтованому програмуванні – спеціальний метод класу, який викликається автоматично при знищенні об'єкта і призначений для його деініціалізації (наприклад звільнення пам'яті).

Ітератор – об'єкт, що абстрагується за єдиним інтерфейсом доступ до елементів колекції. Ітератор іноді також називають курсором, особливо якщо мова йде про базу даних.

Шаблон проектування або патерн (англ. Design pattern) у розробці програмного забезпечення – повторювана архітектурна конструкція, що представляє собою рішення проблеми проектування в рамках деякого часто повторюваного контексту.

Функція у програмуванні – поіменованний фрагмент програмного коду (підпрограма), до якого можна звернутися з іншого місця програми. З ім'ям функції нерозривно пов'язано адресу першої інструкції (оператора), що входить у функцію, якій передається управління при зверненні до функції. Після виконання функції, управління повертається назад у точку повернення – точку програми, звідки дана функція була викликана.

### **Аналіз даних**

Аналіз даних – розділ математики, що займається розробкою методів обробки даних незалежно від їх природи. Можна виділити такі етапи аналізу даних: отримання даних, обробка, аналіз та інтерпретація результатів обробки.

Регресійний аналіз – розділ математичної статистики, присвячений методам аналізу залежності однієї величини від іншої. На відміну від кореляційного аналізу не з'ясовує чи істотний зв'язок, а займається пошуком моделі цього зв'язку, вираженої у функції регресії.

Кореляційний аналіз – це статистичне дослідження (стохастичної) залежності між випадковими величинами (англ. correlation – взаємозв'язок). У найпростішому випадку досліджують дві вибірки (набори даних), у загальному – їх багатовимірні комплекси (групи). Мета кореляційного аналізу – виявити чи існує істотна залежність однієї змінної від інших.

Дисперсійний аналіз – статистичний метод аналізу результатів, які залежать від якісних ознак. Кожен фактор може бути дискретною чи неперервною випадковою змінною, яку розділяють на декілька сталих рівнів (градацій, інтервалів). Якщо кількість вимірювань (проб, даних) на всіх рівнях кожного з факторів однакова, то дисперсійний аналіз називають рівномірним, інакше – нерівномірним.

Дискримінантний аналіз – різновид багатовимірного аналізу, призначеного для вирішення задач розпізнавання образів. Використовується

для прийняття рішення про те, які змінні розділюють (тобто «дискримінують») певні масиви даних (так звані «групи»).

Кластерний аналіз – задача розбиття заданої вибірки об'єктів (ситуацій) на підмножини, що називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися. Завдання кластеризації відноситься до статистичної обробки, а також до широкого класу завдань навчання без вчителя.

Розвідувальний аналіз – займається попереднім експрес-аналізом даних шляхом їх перетворення та/або представлення у зручному вигляді: графічному, табличному, схем, діаграм і т.д.

Повний факторний експеримент (ПФЕ) – експеримент, що реалізує всі можливі неповторювані комбінації незалежних змінних, кожна з яких примусово (активно) варіює на двох рівнях.

Аналіз часових рядів – сукупність математико-статистичних методів аналізу, призначених для виявлення структури часових рядів і для їх прогнозування. Сюди належать, зокрема, методи регресійного аналізу. Виявлення структури часового ряду необхідно для того, щоб побудувати математичну модель того явища, яке є джерелом аналізованого часового ряду. Прогноз майбутніх значень часового ряду використовується для ефективного прийняття рішень.

Коваріаційний аналіз – розділ аналізу даних, що намагається визначити модель зв'язку між залежною величиною, та набором кількісних та якісних величин.

Коефіцієнт кореляції Пірсона (позначають « $r$ ») – в статистиці, показник кореляції (лінійної залежності) між двома змінними  $X$  та  $Y$ , який набуває значень від  $-1$  до  $+1$  включно. Він широко використовується в науці для вимірювання ступеня лінійної залежності між двома змінними.

Коваріація (англ. Covariance) – в теорії ймовірностей та математичній статистиці, числова характеристика залежності випадкових величин. Сутність коваріації полягає в тому, що вона виникає внаслідок невизначеності результату перемножування двох сукупностей чисел.

Випадкова величина – це величина, яка приймає в результаті дослідження одне значення з безлічі можливих, причому появу того чи іншого значення цієї величини до її вимірювання не можна точно передбачити.

Матриця відстаней – це квадратна матриця типу «об'єкт-об'єкт» (порядку  $n$ ), що містить в якості елементів відстані між об'єктами в метричному просторі.



Вибірка – це множина об'єктів, подій, зразків або сукупність вимірів, за допомогою визначеної процедури вибраних зі статистичної популяції або генеральної сукупності для участі в дослідженні.

Дисперсія – міра відхилення значень випадкової величини від центру розподілу. Більші значення дисперсії свідчать про більші відхилення значень випадкової величини від центру розподілу.

Навчання без вчителя – один зі способів машинного навчання, за якого випробовувана система спонтанно навчається виконувати поставлене завдання без втручання з боку експериментатора.

Конкуруюча гіпотеза – альтернативне пояснення результатів дослідження, яке з логічної точки зору не може бути правильним, якщо правильна початкова гіпотеза.

Медіана – 50-й перцентиль або квантиль 0,5 - статистика, яка ділить ранжовану сукупність (варіаційний ряд вибірки) на дві рівні частини: 50% «нижніх» членів ряду даних будуть мати значення ознаки не більше, ніж медіана, а «верхні» 50% - значення ознаки не менше, ніж медіана.

Квантиль – значення, яке задана випадкова величина не перевищує з фіксованою ймовірністю.

Розподіл ймовірностей – це закон, що описує область значень випадкової величини та ймовірності їх результату (появи).

Імовірність – ступінь (відносна міра, кількісна оцінка) можливості настання деякої події.

Рівень значущості критерію перевірки статистичної гіпотези — ймовірність того, що критерій не приведе до відхилення гіпотези у випадкові її істинності.

Нормальний розподіл – розподіл ймовірностей, який в одномірному випадку задається функцією щільності ймовірності, що збігається із функцією Гаусса.

### **Теорія алгоритмів та математична логіка**

Алгоритм – це скінченна множина чітко визначених правил для механічного розв'язання задач певного класу.

Булева функція від  $n$  аргументів – це функція, яка задана на множині  $\{0,1\}^n$  і приймає значення на множині  $\{0,1\}$ .

Диз'юнкція двох висловлень  $P$  і  $Q$  – нове висловлення, яке позначається  $P \vee Q$  (читається « $P$  або  $Q$ »), що істинно в тих випадках, коли хоча б одне з висловлень  $P$  або  $Q$  істинно, і хибне в єдиному випадку, коли обоє висловлення  $P$  і  $Q$  помилкові.

Диз'юнктивна нормальна форма це нормальна форма, в якій булева формула має вид диз'юнкції декількох кон'юнктив (кон'юнкції декількох пропозиційних символів або їх заперечень).

Імплікація двох висловлень  $P$  і  $Q$  – нове висловлення, що позначається  $P \Rightarrow Q$  (читається: «якщо  $P$ , то  $Q$ », або «з  $P$  треба  $Q$ », або « $P$  тягне  $Q$ », або « $P$  достатньо для  $Q$ », або « $Q$  необхідно для  $P$ »), яке хибне в єдиному випадку, коли висловлення  $P$  істинне, а  $Q$  – хибне, а у всіх інших випадках – істинне.

Еквівалентність двох висловлень  $P$  та  $Q$  – нове висловлення, що позначається  $P \Leftrightarrow Q$  (читається: « $P$  еквівалентно  $Q$ », або « $P$  необхідно й достатньо для  $Q$ », або « $P$  тоді й тільки тоді, коли  $Q$ », або « $P$ , якщо й тільки якщо  $Q$ »), що істинне тоді і тільки в тоді, коли одночасно обоє висловлювань  $P$  та  $Q$  або істинні, або хибні, а у всіх інших випадках – хибне.

Загально рекурсивна функція – це всюди визначена частково рекурсивна функція називається.

Заперечення висловлення  $P$  – це нове висловлення, що позначається  $\neg P$ , або  $\bar{P}$  (читається: «не  $P$ » або «не вірно, що  $P$ »), що істинне, якщо висловлення  $P$  хибне, і хибне, якщо висловлення  $P$  істинне.

Кон'юнкція двох висловлень  $P$  і  $Q$  – нове висловлення, яке позначається  $P \wedge Q$  або  $P \& Q$  (читається: « $P$  і  $Q$ »), що істинне лише в єдиному випадку, коли істинні вихідних висловлення  $P$  і  $Q$ , та хибне у всіх інших випадках.

Кон'юнктивна нормальна форма - це нормальна форма, в якій булева формула має вид кон'юнкції декількох диз'юнктив (диз'юнкції декількох пропозиційних символів або їх заперечень).

Логічна операція – це такий спосіб побудови складного висловлення з даних висловлень, при якому істинне значення складного висловлення повністю визначається істинними значеннями вихідних висловлень.

Логічне слідування або наслідок формул  $F_1, F_2, \dots, F_n$  - це формула, яка приймає значення істинність при будь-яких наборах  $\alpha_1, \alpha_2, \dots, \alpha_n$ , які перетворюють формули  $F_1, F_2, \dots, F_n$  в істинність.

Машина з натурально значними регістрами – це абстрактна обчислювальна машина, яка складається з нескінченної кількості регістрів, у кожному з яких може бути записане натуральне число з множини  $N_0$ .

Машина Тюрінга — це абстрактна машина (автомат), що працює зі стрічкою, що складається із окремих комірок, в яких записано символи.

Мінімізація мулевої функції – це перетворення функції, яке веде до зменшення числа символів, а отже, числа змінних.

Нормальна форма – це така форма чого-небудь, що не допускає подальших спрощень.

Нормальний алгоритм Маркова – система послідовних застосувань, підстановок, які реалізують певні процедури отримання нових слів із базових, які побудовані на певному алфавіті.

Нумерація – це бієкція між певною множиною об'єктів, та множиною натуральних чисел.

Предикат – це булевозначна функція  $P: X \rightarrow \{0, 1\}$ .

Примітивно рекурсивна функція – це функція, яка може бути отримана з базисних функцій слідування, додавання одиниці і функції-проектора застосуванням скінченої кількості раз операцій суперпозиції та примітивної рекурсії.

Пропозиційне числення – це числення в алфавіті логіки висловлень, де аксіомами є пропозиційні формули, а правила виведення дозволяють з формул отримувати тільки формули.

Рівносильність двох формул – це співпадання значень даних формул при будь-яких наборах, які підставляються замість змінних.

Складність обчислювальних процесів – це оцінка ресурсів (зазвичай часу) необхідних для виконання алгоритму.

Тавтологія – це формула, яка при будь-яких наборах значень змінних перетворюється у істинне висловлення.

Формальна теорія  $S$  – це множина  $A$  символів, які утворюють алфавіт; множина  $F$  слів алфавіту  $A$ , які називаються формулами; підмножина  $U$  формул,  $U \subset F$ , які називаються аксіомами; множина  $P$  відношень  $R$  на множині формул,  $R \subset P$ ,  $R \subset F^{n+1}$ , які називаються правилами виведення.

Функціональна (логічна) схема – це схема, що складається з логічних елементів, яка виконує певну функцію.

Частково рекурсивна функція – це функція, яка може бути отримана з базисних функцій слідування, додавання одиниці і функції-проектора застосуванням скінченої кількості раз операцій суперпозиції, примітивної рекурсії й мінімізації.

Числення висловлень (логіка висловлень) – це формальна система  $(L, U, R)$ , де  $L$ - мова логіки висловлень,  $U$  – множина аксіом ЧВ,  $R$ - множина правил виведення ЧВ.

## **Цифрова техніка**

Цифрова техніка – це техніка зі створення, передавання, перетворення і обробки сигналів.

Логічна функція  $Y = f(X_1, X_2, \dots, X_{12})$  – це складне висловлення з кількох простих, які пов'язані між собою логічними операціями. Логічна функція  $Y$

набуває значення 0 або 1 ( $Y \in \{0,1\}$ ) при наборі логічних двійкових змінних (аргументів)  $X_i \in \{0,1\}$ .

Інтегральні мікросхеми – це мікроелектронні вироби з високою щільністю упакування електрорадіоелементів (резисторів, діодів, транзисторів) і з'єднань між ними.

Елементами в цифровій схемотехніці називаються найменші неподільні мікроелектронні схеми (вироби), призначені для виконання логічних операцій або зберігання біта інформації.

Типовими функціональними вузлами мікропроцесорної техніки називаються мікроелектронні схеми, призначені для виконання однієї або декількох мікрооперацій.

Цифровий пристрій – це пристрій, призначений для приймання, обробки й передавання цифрової інформації. Наприклад, арифметико-логічний пристрій.

Шифратор – це комбінаційний пристрій, призначений для перетворення цифрової інформації, що подана  $n$ -розрядним розрядно-позиційним кодом, у еквівалентний двійковий  $m$ -розрядний код.

Дешифратор – це комбінаційний пристрій, який перетворює комбінацію вхідних змінних в активний сигнал “лог. 1” або “лог. 0” тільки на одному із виходів.

Мультиплексор – це комбінаційний пристрій, призначений для передачі (комутації) сигналів від одного з кількох інформаційних входів  $X_i$  (шини даних) на один вихід.

Перетворювачем кодів називається функціональний вузол, призначений для перетворення двійкового коду із одної форми в іншу.

Суматором називається функціональний вузол комп'ютера, призначений для додавання двох  $n$ -розрядних слів (чисел).

Цифрові компаратори – це комбінаційні схеми, призначені для порівняння величин двох чисел, поданих у двійковому або двійково-десятковому коді, і формування сигналів результату порівняння.

Тригер – це цифровий автомат, який має два стійких стани 0 або 1 і призначений для зберігання одного біту даних.

Регістром називається послідовнісний пристрій, що призначений для приймання, запам'ятовування, перетворення і передачі двійкової інформації.

Лічильник – це функціональний пристрій, що призначений для підраховування числа імпульсів, які надійшли на його вхід, і фіксації цього числа на його регістрі у вигляді коду.

Мікропроцесор (МП) – програмований логічний пристрій обробки даних, виконаний на основі однієї або декількох ВІС. Стандартний логічний

блок, конкретне призначення (функцію) якого можна визначити (запрограмувати) після його виготовлення.

Однокристальний мікропроцесор є конструктивно завершеним виробом у вигляді однієї ВІС або НВІС.

CISC (Complete Instruction Set Computing)-процесори – процесори з повним набором команд.

RISC (Reduced Instruction Set Computing)-процесори – процесори зі скороченим набором команд.

Системою команд мікропроцесора називається сукупність команд, які може виконувати мікропроцесор.

Мікропроцесорна система (МПС) – сукупність інформаційно-обчислювальних засобів для забезпечення роботи декількох мікропроцесорних пристроїв з одним або декількома мікропроцесорами.

Мультимікропроцесорна (або мультипроцесорна) система – система, яка утворюється об'єднанням деякої кількості універсальних або спеціалізованих МП, завдяки чому забезпечується паралельна обробка інформації і розподілене керування.

Мікропроцесорний комплект (МПК) – сукупність інтегральних схем, сумісних за електричними, інформаційними та конструктивними параметрами і призначених для побудови електронно-обчислювальної апаратури та мікропроцесорних систем керування.

Мікроконтроллер – пристрій, який виконаний в одному корпусі інтегральної схеми і містить основні складові МПК: процесор, пам'ять даних, пам'ять програм, програмовані інтерфейси.

### **Алгоритми та структури даних**

Алгоритм – скінченна послідовність чітких інструкцій, спрямованих на розв'язання певного класу задач після скінченної кількості операцій.

Абстрактний тип даних – математична модель із сукупністю операторів, визначених в рамках цієї моделі.

Алгоритм сортування – алгоритм упорядкування даних за певним правилом.

Бінарне дерево – дерево, кожен з вузлів якого має не більше двох нащадків.

Граф – сукупність елементів та зв'язків між ними.

Дерево – сукупність елементів (які називають вузлами) і відношень, які утворюють ієрархічну структуру вузлів.

Дек – список, в якому вставка і видалення елементів відбувається як на початку, так і на кінці.

Запис – упорядкований набір даних, що називаються полями, де кожне поле має власне ім'я і тип.

Коректний алгоритм – алгоритм, для кожного входу якого є скінченний вихід (результат).

Масив – сукупність однотипних (логічно однорідних) елементів, упорядкованих за індексами, що визначають положення елемента в масиві.

Неорієнтований (простий) граф – сукупність скінченної множини  $V$  елементів та множини  $E$  невпорядкованих пар різних елементів з  $V$ .

Орієнтований граф – сукупність скінченної множини  $V$  елементів та множини  $E$  впорядкованих пар різних елементів з  $V$ .

Складність алгоритму – оцінка ресурсів (часу, пам'яті тощо), необхідних для виконання алгоритму, та певних характеристик (стійкість, природність і т.п.) алгоритму.

Список – послідовність однотипних записів, у якій на місце розташування наступного елемента списку вказує поточний запис.

Стек – список, в якому вставка і видалення елементів відбувається тільки на одному кінці, який називають вершиною стеку.

Структура даних – спосіб організації абстрактних типів даних у комп'ютері.

Черга – список, в якому вставка елементів відбувається в кінці, а видалення – на початку.

Черга з пріоритетом – черга, в якій видалення елемента відбувається на початку, а вставка здійснюється на основі вибраного пріоритету.

### **Організація та обробка електронної інформації**

Дані – це відомості, отримані шляхом вимірювання, спостереження, логічних або арифметичних операцій і представлені у формі, придатній для постійного зберігання, передачі та автоматизованої обробки.

Електронна форма представлення даних – це спосіб фіксації даних, який дозволяє їх збереження, обробку, розповсюдження та представлення користувачеві за допомогою засобів обчислювальної техніки.

Електронна пошта – мережева служба, яка дозволяє користувачам обмінюватися повідомленнями або документами без застосування паперових носіїв.

Електронна таблиця – комп'ютерна програма, яка дозволяє проводити обчислення з даними, представленими у вигляді двовимірних масивів, тобто у табличній формі.

Електронний документ – документ, інформація в якому зафіксована у вигляді електронних даних, включаючи обов'язкові реквізити документа.

Зберігання даних – це процес підтримки вихідної інформації у вигляді, що забезпечує видачу даних за запитами кінцевих користувачів у встановлені терміни.

Збирання електронних даних – це процес отримання відомостей із зовнішнього світу і приведення їх до вигляду, стандартного для прикладної інформаційної системи.

Інтерактивні інформаційні технології – інформаційні технології, які дають можливість ефективно та безпосередньо впливати на інформаційний об'єкт, що створюється або демонструється.

Інформаційна система – сукупність технічного, програмного й організаційного забезпечення, а також персоналу, призначена для того, щоб вчасно забезпечувати користувачів належними даними.

Інформаційна технологія – процес, що використовує сукупність засобів і методів збирання, обробки й передачі даних (первинної інформації) для одержання даних нової якості про стан об'єкта, процесу або явища (інформаційного продукту).

Інформаційний ресурс – це сукупність документів у інформаційних системах (бібліотеках, архівах, банках даних тощо) або сукупність інформаційних продуктів певного призначення, які необхідні для забезпечення інформаційних потреб споживачів у визначеній сфері діяльності.

Мережні інформаційні технології – інформаційні технології, які є об'єднанням технологій збору, зберігання, передачі й обробки інформації на комп'ютері з технікою зв'язку й телекомунікацій.

Мультимедіа – сукупність комп'ютерних технологій, які одночасно використовують кілька інформаційних середовищ: графіку, текст, відео, фотографію, анімацію, звукові ефекти, високоякісний звуковий супровід та ін.

Накопичення даних – процес формування вихідного, несистематизованого масиву даних.

Обмін даними – процес, в ході якого джерело даних їх передає, а одержувач – приймає.

Обробка інформації – сукупність операцій (збирання, введення, записування, перетворення, зчитування, зберігання, знищення, реєстрація), що здійснюються за допомогою технічних і програмних засобів, включаючи обмін по каналах передачі даних.

Презентація – інформаційний чи рекламний інструмент, що дозволяє повідомити потрібну інформацію про певний об'єкт у зручній для одержувача формі.

Мультимедійна презентація – набір слайдів і спецефектів (слайд-шоу), текстових нотаток, що зберігаються в одному файлі.

Операційна система – сукупність програмних засобів, що забезпечують керування апаратними ресурсами обчислювальної системи, процесами, апаратурою і дію користувачів в інформаційній системі.

Прикладні системи(пакети прикладних програм) – комп'ютерні програми, призначені для розв'язання конкретної задачі чи класу задач або для надання користувачеві певних послуг.

Текстовий редактор – комп'ютерна програма, що створює текстові файли без елементів форматування.

Текстовий процесор – комп'ютерна програма, яка дозволяє формувати текст, вставляти в документ графіку та інші об'єкти, що не відносяться до класичного поняття текст.

Формат файлів – усталений стандарт запису інформації у файлі конкретного типу.

Формат електронного документа – формат файлу, в якому міститься текстова і/або аудіовізуальна інформація.

### **Системне програмування**

Бібліотека – збірка підпрограм або об'єктів, що використовуються для розробки програмного забезпечення.

Вихідний модуль – програмний модуль вихідною мовою, оброблюваний транслятором, що представляється як єдине ціле, достатнє для проведення трансляції.

Дані – частина програми, сукупність значень певних елементів пам'яті, перетворення яких здійснює код.

Драйвер – програмне забезпечення, за допомогою якого інше програмне забезпечення (операційна система) дістає доступ до апаратного забезпечення деякого пристрою.

Інструкція (оператор) – найменша автономна частина мови програмування; команда.

Інтегроване середовище програмування – система програмних засобів, використовувана програмістами для розробки програмного забезпечення.

Інтерпретатор – програма (різновид транслятора), що виконує пооператорний (покомандний, порядковий) аналіз, обробку і виконання вихідної програми або запиту.

Керуюча програма – системна програма, що реалізує набір функцій керування, що містить у собі керування ресурсами й взаємодія із зовнішнім



середовищем системи обробки інформації, відновлення роботи системи після прояву несправностей у технічних засобах.

Команда процесора – інструкція, що визначає операцію яка повинна бути виконана процесором.

Компілятор – програма, що виконує перетворення вихідного модуля на мові високого рівня в семантично еквівалентний модуль на мові низького рівня.

Машинний код – система команд конкретної обчислювальної машини, яка інтерпретується безпосередньо мікропроцесором або мікропрограмами даної обчислювальної машини.

Мова асемблера – мова програмування низького рівня, мнемонічні команди якого відповідають інструкціям процесора системи обробки інформації.

Мова високого рівня – мова програмування, поняття й структура якого зручні для сприйняття людиною.

Налагоджувальник – модуль середовища розробки або окремий додаток, призначене для пошуку помилок у програмі.

Мова програмування низького рівня – мова програмування, близька до програмування безпосередньо в машинних кодах використовуваного реального або віртуального процесора.

Об'єктний модуль – програмний модуль, одержуваний у результаті трансляції вихідного модуля, або це модуль машинною мовою, з невизначеними зовнішніми посиланнями.

Операційна система – це сукупність програм для керування ресурсами комп'ютера і забезпечення взаємодії з користувачем.

Прикладна програма – програма, призначена для рішення завдання або класу завдань у певній області застосування системи обробки інформації.

Програма – це дані, призначені для керування конкретними компонентами системи обробки інформації з метою реалізації певного алгоритму.

Програмне забезпечення – сукупність програм системи обробки інформації й програмних документів, необхідних для їхньої експлуатації.

Системна програма – програма, призначена для підтримки працездатності системи обробки інформації або підвищення ефективності її використання.

Системне програмне забезпечення – комплекс програм, які забезпечують управління компонентами системи обробки інформації, та забезпечує інтерфейс обладнання з прикладними програмами.

Системне програмування – це процес розробки системних програм.

Транслятор – програма або технічний засіб, що виконує перетворення програми, представленої на одній з мов програмування, в програму іншою мовою і, в певному сенсі, рівносильну першій.

### **Теорія програмування**

Алгоритм – чітке визначення послідовності скінченного числа дій, що обов'язково приводить до розв'язання певного класу задач.

Алфавіт мови програмування – це список допустимих символів, які можуть використовуватися при створенні програм.

Блок-схема – представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають основні елементи алгоритму.

Відношення – це особливий тип логічних відносин між сутностями, показаних на діаграмах класів і об'єктів.

Декларативне програмування – методологія програмування, відповідно до якої, програма описує, який результат необхідно отримати, замість описання послідовності отримання цього результату.

Діаграма діяльності – UML-діаграма, на якій показано розкладання деякої діяльності на її складові частини.

Діаграма класів – статичне представлення UML структури моделі в термінології класів об'єктно-орієнтованого програмування.

Контекстно-вільна граматики – формальна граматики в усіх правилах цього виду зліва стоїть тільки один нетермінал.

Контекстно-залежна граматики – формальна граматики, у якій вибір означення не впливає на множину мов, породжуваних граматики цього класу.

Лексема – допустима послідовність символів алфавіту мови, яка однозначно розпізнається компілятором.

Логічне програмування – парадигма програмування, заснована на автоматичному доказі теорем.

Мова програмування – формальна знакова система, призначена для запису комп'ютерних програм.

Мови програмування високого рівня – мови в яких особливості конкретних комп'ютерних архітектур не враховуються.

Об'єктно-орієнтоване програмування (ООП) – одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою.

Парадигма програмування – це система ідей і понять, які визначають стиль написання комп'ютерних програм.

Підпрограма – пойменована чи іншим чином ідентифікована частина комп'ютерної програми, яка містить опис певного набору дій.

Програма – це дані, призначені для керування конкретними компонентами системи обробки інформації з метою реалізації певного алгоритму.

Програмна документація – сукупність документів, що містять відомості, необхідні для розробки, виготовлення, супроводу та експлуатації програм.

Програмування – процес розробки комп'ютерних програм на основі створення і використання алгоритмів і структур даних.

Процедура – будь-яка підпрограма, яка не є функцією.

Процедурне програмування — парадигма програмування, заснована на концепції виклику процедури.

Рекурсія – виклик функції чи процедури з неї самої, як правило з іншими значеннями вхідних параметрів.

Семантика мови програмування – початкове смислове значення операторів, основних конструкцій мови.

Сигнатура функції – унікальний ідентифікатор, що включає ім'я функції, а також типи її параметрів і результату.

Синтаксис мови програмування – набір правил, що описує комбінації символів алфавіту, що вважаються правильно структурованою програмою.

Синтаксична діаграма – це спрямований граф з одним вхідним ребром і одним вихідним ребром і позначеними вершинами.

Структура мови програмування – докладно визначає припустимі способи групування лексем у вирази, оператори та інші одиниці мови.

Структурне програмування – методологія програмування полягає в тому, що будь-яка програма може бути створена використовуючи наступні конструкції: послідовне виконання, розгалуження, цикл.

Уніфікована мова моделювання (UML) – мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення.

Форма Бекуса-Наура – це спосіб запису правил контекстно-вільної граматики (формою опису формальної мови).

Функціональне програмування – парадигма програмування, яка розглядає програму як обчислення математичних функцій та уникає станів та змінних даних.

Функція – пойменована частина програми, яка може викликатися з інших частин програми стільки разів, скільки необхідно. Функція обов'язково повертає значення.

Java – об'єктно-орієнтована мова програмування, розроблена компанією Sun Microsystems (в подальшому придбаній компанією Oracle). Програми Java зазвичай транслюються в спеціальний байт-код, тому вони можуть працювати на будь-якій віртуальній Java-машині незалежно від комп'ютерної архітектури. Дата офіційного випуску - 23 травня 1995 року.

Об'єктно-орієнтоване програмування (ООП) – одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Основу ООП складають три основні концепції: інкапсуляція, успадкування та поліморфізм.

Інкапсуляція – один з трьох основних механізмів об'єктно-орієнтованого програмування, суть якого полягає у тому, що об'єкт вміщує не тільки дані, але і правила їх обробки, оформлені в вигляді виконуваних фрагментів (методів). При цьому зазвичай доступ до стану об'єкта напряду заборонено, і ззовні з ним можна взаємодіяти через заданий інтерфейс (відкриті поля та методи), що дозволяє знизити зв'язність.

Успадкування – в об'єктно-орієнтованому програмуванні – механізм утворення нових класів на основі використання вже існуючих. При цьому властивості і функціональність батьківського класу переходять до класу нащадка (дочірнього).

Поліморфізм – концепція ООП, відповідно до якої використовують спільний інтерфейс для обробки даних різних спеціалізованих типів. На противагу поліморфізму, концепція мономорфізму вимагає однозначного зіставлення. У контексті об'єктно-орієнтованого програмування найпоширенішим різновидом поліморфізму є здатність екземплярів підкласу грати роль об'єктів батьківського класу, завдяки чому екземпляри підкласу можна використовувати там, де використовують екземпляри батьківського класу.

Клас – це спеціальна конструкція, яку використовують для групування пов'язаних змінних та функцій. При цьому згідно із термінологією ООП глобальні змінні класу (члени-змінні) називають полями (також властивостями або атрибутами), а члени-функції називають методами класу. Створений та ініціалізований екземпляр класу називають об'єктом класу. На основі одного класу можна створити безліч об'єктів, що відрізнятимуться один від одного своїм станом (значеннями полів).

Конструктор класу (від англ. constructor, деколи скорочують ctor) – спеціальний метод класу, який автоматично викликається при створенні об'єкта. Призначення конструктора – встановити початковий стан об'єкта шляхом ініціалізації атрибутів об'єкта.

Пакет – Java package (пакет Java) – механізм, що дозволяє організувати Java класи у просторі імен аналогічно до модулів мови програмування Модуля. Java пакети можуть міститися і у стислому вигляді в JAR файлах. Зазвичай в пакети об'єднують класи однієї і тієї ж категорії, або подібної функціональності. Кожен пакет надає унікальний простір імен для свого вмісту. Допустимі вкладені пакети. Класи, визначені без явно заданих модифікаторів доступу, (public, protected, private), видимі тільки всередині пакета.

Java Platform, Standard Edition, скорочено Java SE (раніше J2SE або Java 2 Standard Edition) – стандартна версія платформи Java 2, призначена для створення і виконання аплетів і додатків, розрахованих на індивідуальне користування або на використання в масштабах малого підприємства. Не включає в себе багато можливості, надані більш потужною і розширеною платформою Java 2 Enterprise Edition (J2EE) розрахованої на створення комерційних додатків масштабу великих і середніх підприємств.

Java Platform, Enterprise Edition, скорочено Java EE (до версії 5.0 – Java 2 Enterprise Edition або J2EE) – обчислювальна корпоративна платформа Java. Платформа надає API та середовище для розробки і виконання корпоративного програмного забезпечення, включаючи мережеві та веб сервіси, та інші масштабовані, розподілені додатки. Java EE розширює стандартну платформу Java (Java SE - Java Standart Edition).

Аплет – коротка комп'ютерна програма, що функціонально розширює можливості основної програми або інтернет-додатку. Наприклад, додає до веб-сторінки функцію анімації. Для створення аплетів в мережі Інтернет використовується мова програмування Java. Також забезпечує можливість використання у World Wide Web «виконуваного вмісту» шляхом виконання в гіпертекстовому документі міні-програм, відомих як аплети. Незвичайним є формат готового до виконання коду, у який транслюють аплети Java, – це апаратно-незалежний побайтовий формат. Завдяки використанню такого нейтрального формату відкомпільовані Java-програми не залежать від операційної системи, у якій виконуються.

Сервлет – клас Java, реалізація якого розширює функціональні можливості сервера. Сервлет взаємодіє з клієнтами за допомогою принципу запит-відповідь. Хоча сервлети можуть обслуговувати будь-які запити, вони зазвичай використовуються для розширення веб-серверів. Для таких додатків технологія Java Servlet визначає HTTP-специфічні сервлет класи.

Абстрактний клас – це базовий клас, від якого не можна створити екземпляр. На практиці абстрактні класи реалізують один з принципів ООП – поліморфізм. В абстрактному класі можна описати (або не визначити)

абстрактні методи та властивості. Абстрактний метод не реалізують в класі в якому описують, реалізують його у неабстрактному нащадкові. Абстрактні класи вважаються найбільшими узагальненими абстракціями, тобто відношення об'єму описів до об'єму реалізації у них найбільше.

Внутрішній клас (англ. Inner class) – в об'єктно-орієнтованому програмуванні клас, який оголошено усередині іншого класу.

Анонімний внутрішній клас – це клас без імені. Анонімний внутрішній клас можна розглядати як важливий різновид локального внутрішнього класу. При створенні екземпляру анонічного внутрішнього класу використовують назву суперкласу або інтерфейсу.