

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

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

Заступник декана:

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

тел.: (0342) 59-61-08

e-mail: 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. Іляш 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 20 осіб денної 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, спеціаліст, магістр. За період практики у студентів закладаються основи досвіду професійної діяльності, практичних умінь і навичок.

Наскрізна програма практик студентів спеціальності «Комп'ютерні науки» передбачає проведення практики в два етапи:

- 1) на 1 курсі – виробнича практика протягом 6 тижнів 2-го навчального семестру;
- 2) на 2 курсі – підготовка магістерської роботи протягом 12 тижнів 3-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.

Заключною ланкою практичної підготовки є переддипломна практика (підготовка магістерської роботи) студентів, яка проводиться перед

виконанням дипломної роботи. Під час цієї практики поглиблюються та закріплюються теоретичні знання з усіх дисциплін навчального плану, формується фактичний матеріал для виконання дипломної роботи або складання державних екзаменів.

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 Промінвестбанку; 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 ²	1. Методологія промислової розробки програмних продуктів 2. Інформаційні наукові комунікації	Intel I5, 19	Open Office, MikTeX, Visual C++ 2010 Express	так

2.	Лаб.309 Лабораторія ЦІТ, 60 м ²	1. Серверне WEB- програмування	Intel I5, 19	MySQL, Visual C++ 2010 Express	так
3	Лаб.320 Лабораторія ЦІТ, 60 м ²	1. Теорія масового обслуговування 2. Паралельні обчислення та засоби їх реалізації 3. Аналіз і проектування телекомунікаційн их систем	Intel I5, 19	Open MPI Project, SciLab	так

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 засвоєваних 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 відпрацьовування 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», «Е», «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.

Міністерство освіти і науки України
Державний вищий навчальний заклад
"Прикарпатський національний університет імені Василя Стефаника"

НАВЧАЛЬНИЙ ПЛАН

І. ГРАФІК НАВЧАЛЬНОГО ПРОЦЕСУ

Курс	Теоретичне навчання	Екзаменаційна сесія	Матеріалькова робота	Виробнича практика	Контроль за самостійною роботою	Державна атестація	Кількість	Усього
I	27	5		6	2		12	52
II	5	2	12		1	1	2	23
Разом	32	7	12	6	3	1	4	75

III. ПРАКТИКА

Назва практики	Семестр	Тижні
Виробнича практика	2	6
Підготовка магістерської роботи	3	12

IV. АТЕСТАЦІЯ

Назва навчальної дисципліни	Форма атестації (екзамен, дипломний проект (робота))	Семестр
Комп'ютерні науки та інформаційні технології	Дипломна робота	3

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

V. ПЛАН НАВЧАЛЬНОГО ПРОЦЕСУ																				
Шифр за ОПШ	НАЗВА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ, ПРАКТИКИ	Розподіл за семестрами				Кількість кредитів ЕКТС		Кількість годин								Розподіл годин на тиждні за курсами і семестрами				
		Екзамени	Заліки	Практика	Курсова робота	7	8	загальний обсяг				аудиторних				Самостійна робота	І курс		II курс	
								всього	лекції	практичні	семінарські	лабораторні	індивідуальні	1	2		3			
																		з них:		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18			
1. Нормативна частина																				
	1.1. Цикл професійно-орієнтованої гуманітарної та соціально-економічної підготовки (загально-професійної підготовки)						12	360	120	12	90	18			240	4	2	2		
ГС.1	Методологія та організація наукових досліджень		1				3	90	30	12		18			60	2,0				
ГС.2	Англійська мова наукового спілкування		1,2,3				9	270	90		90				180	2,0	2,0	2,0		
	1.2. Цикл професійної та практичної підготовки (спеціальної підготовки)						18	540	180	72			108		360	4	2	9		
ПП.1	Методологія промислової розробки	3					3	90	30	12			18		60			3,0		
ПП.2	Серверне WEB-програмування	1					3	90	30	12			18		60	2,0				
ПП.3	Завадозахищені методи інфообміну	1					3	90	30	12			18		60	2,0				
ПП.4	Об'єктно-орієнтований аналіз і проектування	2					3	90	30	12			18		60		2,0			
ПП.5	Компоненти, засоби та системи ТЗ ІТ	3					3	90	30	12			18		60			3,0		
ПП.6	Розосереджені системи та кластерні обчислення		3				3	90	30	12			18		60			3,0		
Всього за нормативною частиною циклу		5	5				30	900	300	84	90	18	108		600	8	4	11		
2. Варіативна частина																				
	2.1. Дисципліни за вибором ВНЗ (поглиблена підготовка за спеціальністю)						6	180	60	24	36				120		3,0	3,0		
СВЗ.1	Наукові семінари		2,3				6	180	60	24	36				120		3	3		
	2.2. Дисципліни вільного вибору студента						24	720	240	120	120				480	6	7			
	2.2.1. Дисципліни спеціалізації		1,2		2		24	720	240	120	120				480	6	7			
Всього за варіативною частиною частини циклу			4		1	30	900	300	144	156					600	6	10	3		
3. Науково-дослідницька робота і практика																				

8. АНОТАЦІЇ НАВЧАЛЬНИХ ДИСЦИПЛІН

1. Цикл професійно-орієнтованої гуманітарної та соціально-економічної підготовки (загально-професійної підготовки)

<i>Предмет:</i>	Методологія та організація наукових досліджень
<i>Статус:</i>	нормативна
<i>Рік, семестр</i>	1 курс, 1 семестр
<i>Анотація:</i>	Мета курсу: ознайомлення з методологічними засадами наукових досліджень в галузі інформаційних та телекомунікаційних технологій, вибором та формулюванням проблеми дослідження, організацією пошуку наукової інформації та її представленням в рефераті, визначенням об'єкту і предмету наукового дослідження, формулюванням та перевіркою робочих гіпотез дослідження, обґрунтуванням комплексу методик для дослідження та збору емпіричних даних, їхньою обробкою та інтерпретацією, проведенням наукової дискусії, оформленням звіту про наукове дослідження в формі тез, статті, монографії, реферату, дисертації, представлення результатів дослідження на наукових симпозиумах, конференціях і семінарах.
<i>Форми контролю:</i>	залік

<i>Предмет:</i>	Англійська мова наукового спілкування
<i>Статус:</i>	нормативна
<i>Рік, семестр</i>	1 курс, 1- 2 семестр
<i>Анотація:</i>	Мета курсу – сформулювати у студентів загальні та професійно-орієнтовані комунікативні мовленнєві компетенції для забезпечення їхнього ефективного спілкування в академічному та професійному середовищі. Оволодіння навиками оглядового, інформативного та поглибленого читання оригінальної науково-технічної літератури за фахом; використання термінологічних словників і стандартів. Формування умінь читати та перекладати іноземний текст з професійної тематики; складати словник і робити анотацію іноземних наукових праць; читати без словника література за фахом з метою пошуку необхідної інформації; виконувати аналіз та

	презентацію результатів досліджень з використанням професійної іноземної мови.
<i>Форми контролю:</i>	залік

2. Цикл професійної та практичної підготовки (спеціальної підготовки)

<i>Предмет:</i>	Методологія промислової розробки програмних продуктів
<i>Статус:</i>	нормативна
<i>Рік, семестр</i>	2 курс, 3 семестр
<i>Анотація:</i>	Мета курсу – теоретична підготовка з питань архітектури програмного забезпечення, методологія колективної розробки програмного забезпечення; використання типових структур (шаблонів) фрагментації програмного забезпечення; методи організації виробничого процесу; методи контролю якості розроблюваного програмного забезпечення; основні навички і особливості використання екстремального програмування.
<i>Форми контролю:</i>	екзамен

<i>Предмет:</i>	Серверне WEB програмування
<i>Статус:</i>	нормативна
<i>Рік, семестр</i>	1 курс, 1 семестр
<i>Анотація:</i>	Мета курсу – теоретична і практична підготовка з розробки серверного програмного забезпечення з використанням мови Java; уміння використовувати фреймворки JSF та Spring; використовувати системи контролю версій; оволодіння практичними навичками тестування програмного забезпечення.
<i>Форми контролю:</i>	екзамен

<i>Предмет:</i>	Завадозахищені методи інфообміну
<i>Статус:</i>	нормативна
<i>Рік, семестр</i>	1 курс, 1 семестр
<i>Анотація:</i>	Курс присвячений питанням аналізу і способам захисту інформації від ненавмисних перешкод у каналах, комп'ютерних системах та мережах. Метою дисципліни є вивчення теоретичних основ теорії

	інформації та кодування, одержання навичок у постановці та реалізації інженерно-технічних задач захисту і надійного передавання даних.
<i>Форми контролю:</i>	екзамен

<i>Предмет:</i>	Об'єктно-орієнтований аналіз і проектування систем
<i>Статус:</i>	вибіркова
<i>Рік, семестр</i>	1 курс, 1 семестр
<i>Анотація:</i>	Метою курсу є отримання студентами необхідного рівня знань, які необхідні для технологічного проектування програмних інформаційних комплексів, на основі об'єктно-орієнтованого аналізу природних, технічних та соціальних явищ, предметів, об'єктів та систем
<i>Форми контролю:</i>	екзамен

<i>Предмет:</i>	Компоненти, засоби та системи ТЗ ІТ
<i>Статус:</i>	нормативна
<i>Рік, семестр</i>	2 курс, 3 семестр
<i>Анотація:</i>	Предмет охоплює проблеми створення ефективного інструментального забезпечення інформаційних технологій: організацію нових і вдосконалення існуючих комп'ютеризованих та комп'ютерних систем і мереж, їх апаратних і алгоритмічно-програмних засобів, розподілених комп'ютерних систем, методів та засобів реалізації комунікацій в них; розробку наукових фізико-технологічних, логічних, алгоритмічних, мовно-програмних основ побудови та автоматизації проектування апаратних та програмних засобів; науково-технічні дослідження та розробки в галузі первинних і вторинних системних перетворень інформації, аналогових, гібридних та цифрових компонентів комп'ютерних систем, методів та засобів опрацювання знань та природомовних інформаційних об'єктів; створення алгоритмічного, апаратно-програмного, контрольно-діагностичного та інформаційно-вимірювального забезпечення процесів

	формування, збору, зберігання, захисту, обробки, передачі, вводу, виводу та перетворення інформації у комп'ютерних та інформаційно-вимірювальних системах і мережах; визначення, вимірювання та оцінку параметрів комп'ютерних систем, мереж та їх компонент.
<i>Форми контролю:</i>	екзамен

<i>Предмет:</i>	Розосереджені системи і кластерні обчислення
<i>Статус:</i>	нормативна
<i>Рік, семестр</i>	2 курс, 1 семестр
<i>Анотація:</i>	Освоєння базових знань з питань архітектури та організації паралельних обчислювальних систем, а також основних технологій організації паралельних обчислень на багатопроцесорних обчислювальних комплексах з розподіленою або загальною оперативною пам'яттю. Основи паралельного програмування з використанням програмного інтерфейсу. Архітектура високопродуктивних процесорів і кластерних систем, побудова швидких мереж передачі даних і операційних систем для кластерних систем.
<i>Форми контролю:</i>	залік

3. Цикл дисциплін за вибором навчального закладу

<i>Предмет:</i>	Наукові семінари
<i>Статус:</i>	вибіркова
<i>Рік, семестр</i>	1-2 курс, 2-3 семестр
<i>Анотація:</i>	Оволодіння теоретичними знаннями з питань методології наукових досліджень, формування у студентів навичок постановки наукових задач і їх вирішення на теоретичному і емпіричному рівнях. Формування вміння застосовувати її у практичній діяльності, організовувати дослідницьку діяльність. Система та процес наукового дослідження.
<i>Форми контролю:</i>	залік

4. Цикл дисциплін вільного вибору студента

<i>Предмет:</i>	Мультиплатформне програмування
<i>Статус:</i>	вибіркова
<i>Рік, семестр</i>	1 курс, 1 семестр
<i>Анотація:</i>	Метою дисципліни є теоретичне вивчення та отримання практичних навичок у питаннях принципів технології розробки мультиплатформних програмних систем і принципів використання засобів мультиплатформного програмування. Забезпечує спеціальну підготовку студентів в області вивчення засобів мультиплатформного програмування та створення відповідних програмних додатків з їх використанням.
<i>Форми контролю:</i>	залік

<i>Предмет:</i>	Сучасні технології створення програмного забезпечення
<i>Статус:</i>	вибіркова
<i>Рік, семестр</i>	1 курс, 1 семестр
<i>Анотація:</i>	Поняття програмного забезпечення (ПЗ) та проблеми розробки складного програмного забезпечення. Життєвий цикл і процеси розробки програмного забезпечення. Міжнародні та національні стандарти розробки складних програмних продуктів. Методи та засоби розробки програмного забезпечення. Методології розробки ПЗ. Архітектура ПЗ. Засоби автоматизації розробки програмних продуктів. Якість ПЗ, метрики якості, стандарти якості ПЗ. Верифікація, валідація та тестування. Випробування і супровід програмних продуктів. Документування та маркетинг програмного забезпечення.
<i>Форми контролю:</i>	залік

<i>Предмет:</i>	Мережеві технології
<i>Статус:</i>	вибіркова
<i>Рік, семестр</i>	1 курс, 1 семестр
<i>Анотація:</i>	Теоретична підготовка з питань функціонування стандартних протоколів передавання даних та

	програмно-апаратних засобів, які забезпечують їх реалізацію; застосування теоретичних та емпіричних методів вибору топології мереж; методик і принципів розробки; розробки етапів вибору методів доступу до середовища передачі даних; специфіки вибору кадрів даних, типу кодування сигналів, швидкості передачі даних в локальній мережі.
<i>Форми контролю:</i>	залік

<i>Предмет:</i>	Проектування телекомунікаційних мереж
<i>Статус:</i>	вибіркова
<i>Рік, семестр</i>	1 курс, 1 семестр
<i>Анотація:</i>	Загальні принципи проектування систем передавання інформації. Перетворення сигналів у системах передавання інформації. Задачі про потоки в мережах. Математичні методи розв'язання задач оптимізації проектування систем і мереж зв'язку. Побудова транспортної мережі з використанням синхронної цифрової ієрархії. Синхронізація в цифрових системах передавання. Проектування цифрових систем передавання. Проектування багатоканальних модемів. Принципи побудови систем керування сучасних телекомунікаційних мереж.
<i>Форми контролю:</i>	залік

<i>Предмет:</i>	Інтерпретовані мови програмування
<i>Статус:</i>	вибіркова
<i>Рік, семестр</i>	1 курс, 2 семестр
<i>Анотація:</i>	Мета курсу – вивчити можливості мови програмування Python для реалізації процедурного та об'єктно-орієнтованого програмування. Компільовані та інтерпретовані мови програмування. Елементарні конструкції мови Python. Функції в Python. Керуючі конструкції в Python. Структури даних в Python. Об'єктно-орієнтоване програмування в Python.
<i>Форми контролю:</i>	залік

<i>Предмет:</i>	Програмування мобільних аплікацій
<i>Статус:</i>	вибіркова
<i>Рік, семестр</i>	1 курс, 2 семестр
<i>Анотація:</i>	Застосування сучасних досягнень веб-технологій для створення програмного забезпечення: розробка програмного забезпечення для платформи Android. Використання системи контролю версій. Оволодіння практичними навичками тестування програмного забезпечення.
<i>Форми контролю:</i>	залік

<i>Предмет:</i>	Програмування портативних обчислювальних систем
<i>Статус:</i>	вибіркова
<i>Рік, семестр</i>	1 курс, 2 семестр
<i>Анотація:</i>	Мета – засвоїти необхідні знання щодо технологій створення додатків, які базуються на сучасних мобільних платформах і техніки їх застосування у реалізації бізнес-комунікацій. Зміст: основні відомості про мобільні пристрої та застосування. Сучасні мобільні операційні системи (платформи); основи розробки і побудови мобільних додатків.
<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, мережових 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, 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 «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'ясовувати 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 проведених 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'єктно-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 формувати 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'єктно-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 організовувати його роботу з проектування та розроблення комп'ютеризованої системи та після проектного 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 адаптуватися 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 організовувати 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 програмного 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 елементу 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 напрямую 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.

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 – поименований 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 «г») – 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 одновимірному випадку задається функцією щільності ймовірності, що збігається із функцією Гаусса.

Основи цифрової та мікропроцесорної техніки

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 підрахування 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 BIC або HBIC.

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.

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

Інтегровані системи управління

Система – це сукупність взаємопов'язаних елементів, що становить певну цілісність, єдність.

Управління – це сукупність цілеспрямованих дій, що включає оцінку ситуації та стану об'єкта управління, вибір керівних дій і їх реалізацію.

Система управління (СУ) – сукупність ланок, які здійснюють управління, і зв'язків між ними.

Об'єкт управління (ОУ) – умовно виокремлена частина системи, на яку впливає система управління для досягнення необхідного результату.

Автоматизована система управління – автоматизована система (АС), призначена для автоматизації процесів збирання та пересилання інформації про об'єкт управління, її перероблення та видачі керівних дій на об'єкт управління.

Автоматизована система управління – сукупність математичних методів, технічних засобів і організаційних комплексів, що забезпечують раціональне управління складним об'єктом або процесом відповідно до заданої мети.

Інтегровані системи управління – це багаторівневі ієрархічні автоматизовані системи управління, які забезпечують комплексну автоматизацію останнього на усіх рівнях.

Інтегрована автоматизована система управління є ієрархічно організованим комплексом організаційних методів, технічних, програмних, алгоритмічних і інформаційних засобів, які мають модульну структуру і забезпечують наскрізне узгоджене управління матеріальними та інформаційними потоками об'єкта управління.

Інтегрована АСУ підприємством (об'єднанням) – це багаторівнева автоматизована система управління, яка призначена для комплексної автоматизації функцій управління інженерно-технічною, адміністративно-господарською, виробничо-технологічною і соціальною діяльністю промислових підприємств і забезпечує ефективніше розв'язання задач з планування, випуску, розробки, освоєння, виробництва і реалізації продукції.

Інформаційне забезпечення АСУ – сукупність єдиної системи класифікації та кодування техніко-економічної інформації, уніфікованих систем документації і масивів інформації, які використовуються в автоматизованих системах управління, в тому числі форми документів, відеограм, масивів і інтерфейси або протоколи обміну даними.

Лінгвістичне забезпечення – сукупність мовних засобів для формалізації природної мови, побудови і поєднання інформаційних одиниць у процесі спілкування персоналу АСУ із засобами обчислювальної техніки.

Технічне забезпечення – це комплекс технічних засобів (КТЗ), який включає пристрої отримання інформації про технологічні параметри і стани технологічного обладнання; формування і передавання інформації в системі; локального регулювання і управління; обчислювальної техніки; подання інформації обслуговуючому персоналу; передавання інформації в суміжні і несуміжні АСУ, а також виконавчі механізми.

Програмне забезпечення автоматизованої системи – це сукупність програм, процедур, правил та документації, що стосуються функціонування АС.

Математичне забезпечення АСУ – це сукупність математичних методів, моделей і алгоритмів обробки інформації, які використовуються при розв'язуванні функціональних задач і в процесі автоматизації проектних робіт АСУ.

Організаційне забезпечення АСУ – це комплекс документів, які регламентують діяльність персоналу АСУ в умовах функціонування.

Правове забезпечення АСУ є сукупністю правових норм, які регламентують правовідносини при створенні й упровадженні АСУ.

Ергономічне забезпечення АСУ (ЕЗ) – це сукупність методів і засобів, які використовуються на різних етапах розробки та функціонування АСУ, призначене для створення оптимальних умов високоефективної і безпомилкової діяльності людини, спрямованої на швидше освоєння цієї системи.

Алгоритм управління – сукупність вказівок, яка визначає характер дій ззовні на об'єкт управління, що забезпечує алгоритм його функціонування.

Давач – вимірювальний пристрій у вигляді конструктивної сукупності одного або декількох вимірювальних перетворювачів величини, що вимірюється і контролюється, у вихідний сигнал для дистанційного передавання та використання в системах управління.

SCADA – Supervisory Control And Data Acquisition – (Дистанційне управління та збір даних) – системи дистанційного моніторингу та управління.

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

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

Промисловий комп'ютер – це Windows-сумісний комп'ютер в промисловому виконанні.

Універсальний програмований контролер – це мікропроцесорна система, потужність якої практично не відрізняється від потужності промислового комп'ютера.

Програмований логічний контролер (ПЛК) – це мікропроцесорна система спеціального призначення з проблемно-орієнтованим програмним забезпеченням для реалізації алгоритмів логічного управління і/або замкнутих систем автоматичного управління у сфері промислової автоматики.

Контролер збору даних є мікропроцесорною системою, призначеною тільки для збору інформації.

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.

Системний аналіз і теорія прийняття рішень

Системи і методи прийняття рішень

Система – це об'єкт будь-якої природи чи сукупність об'єктів довільної, в тому числі і різної природи, який володіє вираженою системною властивістю, тобто властивістю, якою не володіє ні одна з частин системи при будь-якому способі поділу, і яка не виводиться з властивостей частин.

Основні характеристики системи – вхідний вплив, стан вихід.

Динамічні (інерційні) системи – системи де зміна стану (виходу) не може відбуватися миттєво, описуються за допомогою інтегро-диференціальних рівнянь.

Статичні (без інерційні) – системи де припускається що зміна стану (виходу) може відбуватися миттєво, процеси в них описуються за допомогою функціональних залежностей між входом станом і виходом.

Детерміновані (визначені) системи – системи де вважається, що невизначеність повністю відсутня і при наявності повної інформації про функції входу і стану вихід системи визначається однозначно.

Стохастичні (ймовірнісні) системи – системи де вважається, що процеси в системі мають статистичний характер, описуються за допомогою апаратів теорії ймовірності та математичної статистики.

Складні системи – системи що володіють особливими властивостями (унікальності, слабо передбачуваності, негентропійності), прикладом є складні технічні системи, біологічні, організаційні системи, системи де складовою підсистемою є людина.

Автоматизовані системи – складні система з визначеною роллю елементів двох типів: технічних засобів і дій людини.

Адитивність – системна властивість коли результат дорівнює сумі складових.

Субадитивність – системна властивість коли результат менше суми складових.

Суперадитивність – системна властивість коли результат більше суми складових.

Однорідність – системна властивість коли результат змінюється у стільки разів у скільки і складова.

Система лінійна – якщо система лінійна відносно нульового початкового стану, лінійна відносно нульового вхідного впливу і задовольняє властивості декомпозиції.

Структура системи – є її розбиття на групи елементів із вказуванням зв'язків між ними, яка не змінюється за весь час розгляду і дає уявлення про систему в цілому.

Декомпозиція системи – є поділ системи на частини, зручний для яких-небудь операцій з цією системою.

Декомпозиція системи за принципом слабкості зв'язків – поділ на підсистеми коли зв'язки між елементами підсистеми набагато сильніші ніж між елементами різних підсистем.

Ієрархія – структура з наявністю підлеглості, тобто нерівноправних зв'язків між елементами, коли вплив в одному з напрямків здійснює набагато більшу дію на елемент, ніж іншому.

Модуль – група елементів системи, яка описується тільки своїми входами і виходами і яка володіє певною цілісністю.

Процесом у системі – називається набір станів системи, що відповідає впорядкованій неперервній чи дискретній зміні деякого параметру, що визначає характеристики (властивості) системи.

Унікальність – довільна складна система не має аналогів поведінки або вони настільки рідко зустрічаються, що їх наявністю можна знехтувати.

Слабопередбачуваність – ніякі як завгодно детальні знання морфології і функцій елементів (підсистем) не дозволяють однозначно визначити функції системи, ніякі як завгодно детальні і точні знання про поведінку системи в минулому не дозволяють точно передбачити її поведінку в майбутньому.

Негентропійність (цілеспрямованість) – система в визначених межах здатна керувати своєю ентропією (зменшувати її, зберігати сталою, гальмувати збільшення) при випадкових несприятливих діях середовища; або (і) здатна здійснювати поведінку, яка переслідує досягнення визначеної мети.

Емпірико-інтуїтивний – ймовірно, найбільш давній, оснований на спостереженнях (експериментах) та "вгадуванні" взаємозв'язків між ними

Дедуктивно-аксіоматичний – процес пізнання прийнятий Евклідом в його "Началах"

Конструктивний – процес пізнання узагальнений Сократом, що йде від часткового до загального та уникаючи догматизму

Асоціативний – процес пізнання більш всього характерний для Максвелла та Ейнштейна, заснований на вловлюванні подібності між дуже віддаленими фактами та з'єднанні розрізнених елементів за допомогою нової більш глибокої точки зору.

Функція – правило отримання результатів, предписаних призначенням системи. Функція системи описується за допомогою таких понять, як відношення між змінними, векторами, множинами. Функція встановлює, що робить система для досягнення поставленої мети безвідносно до фізичних засобів (елементів, зв'язків), які складають систему, і не визначає, як побудована система.

Функціонування – реалізація функції системи.

Синтез – це процес породження функцій і структур, необхідних і достатніх для отримання визначених результатів.

Абстрактний синтез – це синтез функції, оскільки виявляючи функції, які реалізуються системою, визначають деяку абстрактну систему, про яку відомо тільки те, що вона буде робити.

Структурний синтез – породження структури, яка реалізує задані функції.

Аналіз – по відомій структурі і відомим параметрам вивчається поведінка системи, тобто досліджуються її властивості і характеристики.

Подібність – це взаємно однозначна відповідність (ізоморфізм, ін'єкція) між двома об'єктами, при якій відомі методи переходу від параметрів одного об'єкта до параметрів іншого, а математичний опис цих об'єктів може бути перетворений до тотожного.

Моделювання – процес представлення об'єкта дослідження адекватною йому моделлю і проведення досліджень моделі для отримання інформації про об'єкт досліджень.

Модель – фізична або абстрактна система, яка адекватно представляє об'єкт дослідження.

Фізична модель – утворюється з сукупності матеріальних об'єктів; при її побудові використовуються фізичні властивості об'єктів, при чому природа цих об'єктів не обов'язково та ж сама, що і в вихідному об'єкті.

Абстрактна модель – модель, в якій компонентами є поняття: словесний опис, схеми, алгоритми, програми, математичний опис.

Дескриптивна модель – словесний опис закономірностей, які характеризують процес, який досліджується, а також постановка прикладної задачі або чітке формулювання мети досліджень (як правило складається спеціалістом в конкретній області без активної участі системотехніка). Обов'язково повинна містити перелік залежностей, які підлягають оцінці.

Формалізована схема – здійснюється, якщо неможливий безпосередній перехід від дескриптивної моделі до математичної. Може бути словесною, але містити строгий формальний опис об'єкта. Тут вибирається сукупність характеристик станів і параметрів об'єкта, формується математична мета досліджень. Вихідна інформація представляється у вигляді графіків, таблиць.

Ідентифікація – побудова оптимальної у деякому змісті моделі реальної системи по результатах спостережень за вхідними і вихідними її змінними.

Покращення систем – процес, який забезпечує роботу системи згідно очікуваному. При цьому передбачається, що система вже створена, однак зміни, які пропонуються позитивні. Методологія, яка лежить в основі покращення систем, називається науковою парадигмою (система понять і методів, яка застосовуються в природничих науках).

Проектування систем (системний підхід) – напрям методології досліджень, який полягає в дослідженні об'єкта як цілісної множини елементів в сукупності відношень і зв'язків між ними, тобто розгляд об'єкта як системи. Методологія, яка лежить в основі проектування систем, називається системною парадигмою (системним підходом).

Поняття – це представлення думки у вигляді впорядкованих знань з можливістю їх передачі.

Шкала найменувань – проста класифікація, позначення класів за допомогою імен або номерів для їх упізнання. Це є першим кроком у ієрархії процедур вимірів. В загальному вона призначена для встановлення еквівалентностей між елементами перетворення, завдяки якому можлива модифікація процесу вимірів без зміни результатів.

Порядкова шкала – служить для побудови впорядкованого класу або встановлення співвідношень нерівності між числами або елементами, яким відповідають ці числа. Допускаються перетворення у вигляді монотонних функцій.

Інтервальна шкала – призначається для встановлення рівності (нерівності) числових інтервалів. Допускаються перетворення у вигляді лінійних функцій

Шкала відношень – призначена для відновлення рівності (нерівності) числових відношень. Допускаються перетворення у вигляді лінійних функцій, що проходять через початок координат.

ОПР – особа що приймає рішення.

Альтернатива – це напрямки можливої діяльності в момент прийняття рішення.

Множина альтернатив – сукупність усіх можливих напрямків діяльності.

Підмножина вибраних альтернатив – це ті альтернативи з множини альтернатив, які задовольняють критерію вибору.

Критерій вибору – умова, що дозволяє з множини альтернатив відібрати альтернативи, які в найкращій мірі відповідають заданій цілі.

Критерій переваг – правило, згідно з яким альтернативі ставлять у відповідність певну величину переваги над іншими альтернативами, виражену в ранговій або більш сильній шкалі.

Оцінка альтернативи – це присвоєння альтернативі певного значення, бальної оцінки, тощо. Оцінка альтернативи може проводитись за одним або декількома критеріями і мати кількісний або якісний характер.

Прийняття рішення – дія над множиною альтернатив, у результаті якої одержуємо підмножину вибраних альтернатив. Прийняття рішення здійснюється на основі оцінки альтернатив.

Функція корисності u – числовим представленням відношення переваги.

Ефективна точка – точка, у якій виконується принцип домінування відносно будь якої точки.

Множина Парето – множина ефективних точок.

Метод рівномірної оптимізації – функція корисності і тим самим глобальна властивість альтернатив визначається як сума критеріїв, які відповідають локальним властивостям.

Метод лінійної згортки (метод зважених сум) – коли ОПП приписує різну значимість критеріям і вони є однакові протягом всього діапазону їх змін.

Система підтримки прийняття рішень – є інтерактивною комп'ютерною системою для підтримки різних видів діяльності при прийнятті рішень із слабо структурованих і неструктурованих проблем.

Структуровані проблеми або кількісно сформульовані проблеми – ті, в яких істотні залежності цілком з'ясовані і можуть бути подані за допомогою числових оцінок.

Слабкоструктуровані, або змішані проблеми – ті, які містять як якісні так і кількісні елементи, причому якісні, маловідомі та невизначені сторони проблеми мають тенденцію домінувати.

Неструктуровані проблеми – ті, в яких відомий тільки перелік основних параметрів, але кількісні зв'язки між ними встановити неможливо.

Інтегрована інтелектуальна СППР є потужною, гнучкою, адаптивною системою, яка охоплює широке коло галузей застосування, методів оптимізації ресурсів, методів прийняття рішень і здатна надати ефективну допомогу ОПП в умовах визначеності та невизначеності.

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 граматика – формальна граматика в усіх правилах цього виду зліва стоїть тільки один нетермінал.

Контекстно-залежна граматика – формальна граматика, у якій вибір означення не впливає на множину мов, породжуваних граматиками 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 – поійменована 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 – поійменована 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 Standard 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 анонімного внутрішнього класу використовують назву суперкласу або інтерфейсу.