



ПРОГРАМУВАННЯ АЛГОРИТМІЧНИХ СТРУКТУР

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітня програма	Комп'ютерний моніторинг та геометричне моделювання процесів та систем
Статус дисципліни	Нормативна
Форма навчання	Заочна
Рік підготовки, семестр	1 курс, осінній семестр
Обсяг дисципліни	4,5 кредити, 135 годин, з яких 10 години аудиторних (4 год лекції, 2 год практичні, 4 год лабораторні), 125 годин становить самостійна робота
Семестровий контроль/ контрольні заходи	Залік/модульна контрольна робота, розрахункова графічна робота
Розклад занять	http://rozklad.kpi.ua/
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: д.т.н., проф. Барабаш Олег Володимирович, bar64@ukr.net (у робочий час) Практичні та лабораторні заняття: д.т.н., проф. Барабаш Олег Володимирович, bar64@ukr.net (у робочий час)
Розміщення курсу	Кампус

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Навчальна дисципліна «Програмування алгоритмічних структур» є тією частиною знань з програмування, яка пов'язана з умінням вирішувати завдання шляхом розробки алгоритмів та створення програм, які реалізують ці алгоритми. За останні десятиліття значно зросла роль створення програмних систем у сучасному світі. Це зумовлено потребою розв'язання важливих практичних задач в галузі інформаційних технологій.

Метою навчальної дисципліни є формування у студентів алгоритмічного мислення та набуття навиків розробки програм на мові програмування Java для розв'язування теоретичних і практичних задач у професійній діяльності, а також розвитку логічного та алгоритмічного мислення при виявленні та дослідженні закономірностей, яким підпорядковуються реальні процеси в сфері програмування.

Предметом вивчення є основи алгоритмізації та програмування на алгоритмічній мові, роботи в інтегрованих середовищах програмування, методики розробки програм необхідних для розв'язання професійних задач.

Програмні результати

Результатом вивчення навчальної дисципліни є формування у студентів **компетентностей**:

— здатність застосовувати професійні знання й уміння на практиці (СК-2) (Освітньо-професійна програма першого (бакалаврського) рівня вищої освіти ступеня “бакалавр”. Галузь

знань 12 Інформаційні технології. Спеціальність 122 Комп'ютерні науки та інформаційні технології. — Київ: НТУУ “КПІ”, 2016);

— здатність використовувати математичні методи для прийняття ефективних рішень під час розв'язання професійних задач в процесі розробки ІС та ІТ (ПК-1);

— до проектування архітектури комп'ютерної системи, вибору і інтегруванню компонентів технічного і стандартного програмного забезпечення при реалізації ІС та ІТ (ПК-3);

— здатність аналізувати проблеми, ставити постановку цілей і завдань, виконувати вибір способу й методів дослідження, а також оцінку його якості СК-5;

— здатність застосовувати принципи структурного програмування, сучасних процедурно-орієнтованих мов, основних структур даних під час програмної реалізації алгоритмів для систем комп'ютерної графіки ПК-1с;

— володіння алгоритмічним мисленням, методами програмної інженерії для реалізації програмного забезпечення з урахуванням вимог до його якості, надійності, виробничих характеристик.

Згідно з вимогами програми навчальної дисципліни студенти після вивчення дисципліни мають продемонструвати такі результати навчання:

знання:

— основ організації обчислювального процесу на ЕОМ;

— теоретичних основ алгоритмізації, проектування та тестування програм;

— методів структурного програмування;

— алгоритмів розв'язування типових задач;

— принципів структурного програмування;

— сучасних об'єктно-орієнтованих мов;

— основних структур даних, способів та методів навчання;

— методів самоосвіти;

вміння:

— застосовувати мови програмування (зокрема мову програмування Java), мови опису інформаційних ресурсів, мови специфікацій, інструментальні засоби під час проектування та створення інформаційних систем, продуктів і сервісів інформаційних;

— програмно реалізувати алгоритми розв'язання задач, розроблення системного і прикладного програмного забезпечення інформаційних систем і технологій;

— проектувати компоненти програмного забезпечення;

— проектувати людино-машинний інтерфейс інформаційних систем;

— розробляти оптимальні алгоритми для широкого кола задач;

— реалізувати алгоритми мовою програмування Java як закінченого програмного продукту;

— виконувати аналіз коректності програм, їхнє налагодження та тестування з використанням сучасних технологій програмування;

— застосовувати набуті базові знання з дисципліни в професійній діяльності під час розробки, налагодження та експлуатації ІС та технологій;

— застосовувати принципи структурного програмування при проектуванні і розробці програм;

— застосовувати основні структури даних під час програмної реалізації алгоритмів;

— застосовувати типові алгоритми обробки даних.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Згідно з робочим навчальним планом навчальна дисципліна “Програмування алгоритмічних структур” викладається студентам першого року підготовки першого (бакалаврського) рівня вищої освіти ступеня “бакалавр” спеціальності “Комп'ютерні науки” освітньої програми “Комп'ютерний моніторинг та геометричне моделювання процесів та систем” у першому навчальному семестрі. Дисципліна “Програмування алгоритмічних структур” вивчається на основі

знань, отриманих з дисциплін: “Дискретна математика”, “Алгоритмізація та програмування”. Вона забезпечує вивчення таких навчальних дисциплін як: “Програмування алгоритмічних структур-2.” “Операційні системи”, “Об’єктно-орієнтоване програмування”.

Зміст навчальної дисципліни

Розділ 1. Основи програмування на мові Java.

Тема 1.1. Вступ до програмування на Java.

Тема 1.2. Прості структури даних. Операції в Java

Тема 1.3. Програмування розгалужених алгоритмів.

Тема 1.4. Масиви та їх обробка із застосуванням циклічних алгоритмів.

Розділ 2. Програмування алгоритмів обробки інформації.

Тема 2.1. Алгоритми матричних перетворень.

Тема 2.2. Алгоритми пошуку елементів масивів. Рекурсія.

Тема 2.3. Алгоритми сортування масивів.

Тема 2.4. Класи, об’єкти, методи в мові Java.

Тема 2.5. Конструктори класів в мові Java.

Тема 2.6. Алгоритми обробки текстової інформації.

Тема 2.7. Відлагодження та тестування програм.

Розділ 3. Програмування алгоритмів на Java із застосуванням складних структур даних.

Тема 3.1. Колекції в Java.

Тема 3.2. Алгоритми рішення комбінаторних задач.

Тема 3.3. Структуризація алгоритмів в класах та методах.

Тема 3.4. Обробка виключень в мові Java.

Тема 3.5. Потoki вводу-виводу. Робота із файлами.

Тема 3.6. Аналіз алгоритмів. Обчислювальна складність алгоритмів.

Тема 3.7. Якість та ефективності програм.

3. Навчальні матеріали та ресурси

Основна література:

1. Програмування алгоритмічних структур [Електронний ресурс]: навчальний посібник для студентів спеціальності 122 – «Комп’ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Крячок, М. П. Воронько. – Електронні текстові дані (1 файл: 456,78 Кбайт). Київ : КПІ ім. Ігоря Сікорського, 2020. 62 с.

2. Крячок О.С., Методичні вказівки до виконання комп’ютерних практикумів “Основи програмування та алгоритмічні мови – 2. Спеціальні засоби мови програмування” ч. 2, Київ, НТУУ “КПІ” ВПІ ВПК “Політехніка” 2012. 41 с.

3. Аккуратов Е. Е. Знакомьтесь Java. Самоучитель. 244 с.

4. Герберт Шилдт. Java-8. Руководство для начинающих. 7-е изд. –М.: ООО «Диалектика», 2019. 810 с.

Додаткова література:

1. Брюс Экель Философия Java. 4-е изд. СПб.: Питер, 2009. 640 с.

2. Герберт Шилдт. Java-8. Полное руководство. 9-е изд. М.: ООО «И.Д.Вильямс», 2015. 1377с.

4. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальна дисципліна охоплює 4 години лекцій, 2 години практичних занять і 4 години лабораторних занять, передбачає виконання розрахунково-графічної роботи, а також модульної контрольної роботи, яка проводиться на останньому занятті і триває 1 академічну годину.

На лекціях студенти одержують основний теоретичний матеріал. Основні завдання циклу лабораторні заняття полягають в тому, щоб студенти закріпили і поглибили засвоєння матеріалу лекцій, отримали практичні навички. На лабораторних заняттях розглядаються завдання та особливості виконання лабораторних робіт та відбувається захист лабораторних робіт, здійснюється поточний контроль.

Розрахунково-графічна робота виконується студентами самостійно на основі знань, отриманих під час лекцій, практичних занять, лабораторних занять і самостійної роботи. Вона сприяє поглибленому засвоєнню та узагальненню теоретичного матеріалу і закріпленню практичних навичок.

Передбачається одна модульна контрольна робота, яка проводиться у кінці семестру. Мета контрольної роботи полягає у перевірці засвоєння матеріалу дисципліни. Для проведення контрольної роботи виділяється 1 година за рахунок лабораторних занять. На контрольну роботу виносяться практичні завдання розробки програм на мові Java.

Лекційні заняття

Розділ 1. Основи мови програмування на мові Java	
Тема 1.1. Вступ до програмування на Java.	
1	Історія створення мови Java. Особливості мови Java. Відмінності Java від C++. Найпростіша програма на Java. Hello World!.
Тема 1.2. Прості структури даних. Операції в Java.	
2	Типи даних та змінні в Java. Примітивні типи. Особливості застосування літералів в Java. Оголошення та ініціалізація змінних. Перетворення і приведення типів даних. Операції в Java (арифметичні, побітові, операції відношення, логічні, присвоєння, умовна операція, пріоритети операцій). Особливості вводу-виводу.

Практичні заняття

1	Складання простих програм на Java, використання різних типів даних та операцій
---	--

Лабораторні заняття

1	Програмування лінійних алгоритмів
2	Програмування розгалужених алгоритмів
3	Програмування циклічних алгоритмів
4	Програмування задач пошуку та сортування масивів даних

5. Самостійна робота студента/аспіранта

Оскільки для студентів заочної форми навчання на аудиторні заняття виділяється незначна кількість годин, то основна частка матеріалу виноситься на самостійне опрацювання. Самостійна робота студентів заочної форми навчання є невід'ємною складовою процесу

навчання і основним способом оволодіння навчальним матеріалом. Самостійна робота виконується студентами за такими темами.

1	Розділ 1. Основи програмування на мові Java
	Тема 1.3. Програмування розгалужених алгоритмів. Особливості застосування оператора break [4, стор. 84]. Тема 1.4. Масиви та їх обробка із застосуванням циклічних алгоритмів. Перетворення та приведення типів даних [4, стор. 88].
2	Розділ 2. Програмування алгоритмів обробки інформації
	Тема 2.1. Алгоритми матричних перетворень. Методи обробки n-вимірних масивів [4, стор. 93]. Тема 2.2. Алгоритми пошуку елементів масивів. Рекурсія. Алгоритм бінарного пошуку [4, стор. 98]. Тема 2.3. Алгоритми сортування масивів. Порівняльний аналіз методів сортування [4, стор. 106]. Тема 2.4. Класи, об'єкти, методи в мові Java. Збирання сміття та методи завершення [4, стор. 167]. Тема 2.5. Конструктори класів в мові Java. Ключове слово this [4, стор. 171]. Тема 2.6. Алгоритми обробки текстової інформації. Особливості стандартних потоків вводу та виводу [4, стор. 121]. Тема 2.7. Відлагодження та тестування програм. Модульне тестування за допомоги JUnit [4, стор. 132].
3	Розділ 3. Програмування алгоритмів на Java із застосуванням складних структур даних
	Тема 3.1. Колекції в Java. Клас TreeMap. Ітератори [4, стор. 204]. Тема 3.2. Алгоритми рішення комбінаторних задач. Алгоритми визначення всіх комбінацій [4, стор. 328]. Тема 3.3. Структуризація алгоритмів в класах та методах. Повернення об'єктів із методів [4, стор. 223]. Тема 3.4. Обробка виключень в мові Java. Створення своїх класів виключень [4, стор. 265]. Тема 3.5. Потoki вводу-виводу. Робота із файлами. Клас Console [4, стор. 286]. Тема 3.6. Аналіз алгоритмів. Обчислювальна складність алгоритмів. Переборні алгоритми та їх реалізація з точки зору обчислювальної складності [4, стор. 326]. Тема 3.7. Якість та ефективність програм. Поняття про Java Code Conventions [4, стор. 332].
4	Виконання розрахунково-графічної роботи

Практичні заняття, що виносяться на самостійну роботу

2	Програмування розгалужених алгоритмів.
3	Програмування циклічних алгоритмів. Алгоритми пошуку елементів масивів.
4	Алгоритми обробки матриць, сортування даних.
5	Програмування із застосуванням класів, об'єктів та методів.
6	Алгоритми обробки текстової інформації.
7	Методи відлагодження та тестування алгоритмів.

8	Програмування об'єктів, їх властивостей та методів.
9	Модульна контрольна робота

Лабораторні роботи, що виконуються під час самостійної роботи

1	Програмування лінійних алгоритмів
2	Програмування розгалужених алгоритмів
3	Програмування циклічних алгоритмів
4	Програмування задач пошуку та сортування масивів даних

Політика та контроль

6. Політика навчальної дисципліни (освітнього компонента)

Для успішного проходження курсу та складання контрольних заходів необхідним є вивчення навчального матеріалу за кожною темою. Специфіка курсу передбачає акцент на розумінні підходів і принципів, отримання практичних навичок, а не просто запам'ятовування визначень. Кожен студент повинен ознайомитися і слідувати Положенню про академічну доброчесність, Статуту і розпорядку дня університету. Для успішного засвоєння програмного матеріалу студент зобов'язаний:

- не запізнюватися на заняття;
- не пропускати заняття, а в разі пропуску відновити за допомогою консультування з викладачем та з використанням Кампус конспект, самостійно вивчити матеріал пропущеного заняття та скласти відповідні контрольні заходи в індивідуальному порядку;
- конструктивно підтримувати зворотній зв'язок на всіх заняттях;
- брати активну участь у освітньому процесі;
- своєчасно і старанно виконувати завдання для самостійної роботи;
- бути доброзичливим до однокурсників та викладачів;
- брати участь у контрольних заходах;
- за об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись індивідуально (в дистанційній on-line формі за погодженням із деканом факультету);
- будь-яке копіювання або відтворення результатів чужої праці (у тому числі списування), якщо тільки робота не має груповий формат, використання чужих завантажених з Інтернету матеріалів кваліфікується як порушення норм і правил академічної доброчесності та передбачає притягнення винного до відповідальності, у порядку, визначеному чинним законодавством та Положенням про академічну доброчесність університету. Результатом невиконання та/або недотримання правил може бути оцінка «не зараховано» за курс.

7. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Максимальна кількість балів з кредитного модуля дорівнює 100.

Рейтинг студента з кредитного модуля складається з балів, отриманих за:

- 1) роботу на практичному занятті;
- 2) виконання та захист чотирьох лабораторних робіт;
- 3) виконання розрахунково-графічної роботи;
- 4) написання модульної контрольної роботи.

1.Робота на практичних заняттях

На практичному занятті може бути проведено бліцопитування студентів щодо виконання домашнього завдання. Такі опитування проводяться на практичному занятті в обсязі декількох питань. Ваговий бал за вірну відповідь – 1. Максимальна кількість балів, що може отримати кожен студент за практичне заняття – 5 балів.

2.Виконання лабораторних робіт

Оцінюються 4 роботи, передбачених робочою програмою. Максимальний ваговий бал – $15 \cdot 4 = 60$ балів

Рейтингові бали кожної роботи складаються з балів за правильність виконання (від 0 до 7) та захист роботи (від 0 до 6), балів за оформлення протоколу роботи (від 0 до 2). За несвоєчасну здачу звіту з лабораторної роботи – штраф 5 балів.

3. Виконання розрахунково-графічної роботи

Оцінюються 1 робота, що передбачена робочою програмою. Максимальний ваговий бал – 15 балів.

Рейтингові бали кожної роботи складаються з балів за виконання (5 балів) - за правильність виконання (від 0 до 7) та балів за оформлення роботи (від 0 до 3). За несвоєчасну здачу роботи – штраф 5 балів.

4. Модульний контроль

Максимальний ваговий бал – 20.

Сума вагових балів контрольних заходів протягом семестру складає:

$$R = 5 + 60 + 15 + 20 = 100 \text{ балів}$$

Необхідною умовою допуску до заліку є зарахування модульної контрольної роботи та розрахунково-графічної роботи, а також стартовий рейтинг (r_c) не менше 40% від R , тобто 40 балів.

Студенти, які набрали протягом семестру рейтинг з кредитного модуля менше $0,6R$, зобов'язані виконувати залікову контрольну роботу.

Студенти, які набрали протягом семестру необхідну кількість балів ($RD \geq 0,6R$), мають можливості:

- отримати залікову оцінку (залік) так званим «автоматом» відповідно до набраного рейтингу (таблиця);
- виконувати залікову контрольну роботу з метою підвищення оцінки (у разі отримання оцінки, більшої ніж «автомат» з рейтингу, студент отримує оцінку за результатами залікової роботи).

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

8. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль:

1. Історія створення мови Java. Особливості мови Java.
2. Відмінності Java від C++
3. Примітивні типи даних в Java

4. Особливості застосування літералів в Java
5. Оголошення та ініціалізація змінних
6. Звичайна та динамічна ініціалізація змінних.
7. Область видимості та час існування змінних.
8. Перетворення і приведення типів даних
9. Явне та автоматичне перетворення типів даних
10. Програмування розгалужених алгоритмів із застосуванням оператора if-else. Вкладені конструкції if-else.
11. Програмування розгалужених алгоритмів із застосуванням оператора switch
12. Особливості застосування операторів break, default при застосуванні оператора switch.
13. Особливості використання операторів вибору для розгалуження алгоритмів
14. Види операцій в Java. Умовна операція в Java.
15. Арифметичні та побітові операції в Java.
16. Операції відношення, логічні операції в Java
17. Операції присвоєння, умовна операція
18. Пріоритети операцій в Java
19. Масиви. Методи оголошення та ініціалізації одновимірних масивів.
20. Масиви. Методи оголошення та ініціалізації багатовимірних масивів.
21. Доступ до елементів масиву. Методи перебору елементів масивів
22. Ітераційні оператори (цикли) в Java.
23. Особливості застосування циклів із передумовою та післяумовою.
24. Особливості застосування операторів переходу break, continue, return.
25. Приклади використання циклів.
26. Основні відомості про класи та об'єкти. Визначення класу.
27. Порядок створення об'єктів. Доступ до полів об'єкту.
28. Особливості присвоєння та порівняння змінних посиляльного (об'єктного) типу.
29. Правила створення методів в класі.
30. Модифікатори доступу.
31. Використання параметрів, що передаються в метод.
32. Особливості повернення результатів роботи методів.
33. Особливості використання параметризованих та непараметризованих методів.
34. Застосування конструкторів в класах.
35. Особливості використання параметризованих та непараметризованих конструкторів.
36. Особливості оператора new.
37. Особливості застосування ключового слова this.
38. Особливості типу String.
39. Способи створення текстових рядків.
40. Методи роботи з рядками
41. Особливості створення масивів рядків.
42. Особливості присвоєння та порівняння рядків.
43. Основні методи класу String.

44. Особливості застосування масиву символів та текстових рядків.
45. Методи перебору символів текстового рядку.
46. Властивість незмінності рядків
47. Використання рядків для управління оператором switch
48. Особливості стандартних потоків вводу та виводу.
49. Особливості рекурсивного виклику методів
50. Найпростіші прийоми відлагодження.
51. Використання протоколювання для відлагодження.
52. Інструментальні засоби відлагодження.
53. Тестування програм та його роль в програмному проекті.
54. Модульне тестування за допомоги JUnit.
55. Особливості стандартних потоків вводу та виводу.
56. Особливості рекурсивного виклику методів

Робочу програму навчальної дисципліни (силабус) «Програмування алгоритмічних структур»:

Складено професором кафедри АПЕПС, д.т.н., проф. Барабашом Олегом Володимировичем

Ухвалено кафедрою АПЕПС (протокол № 16 від 18.06.2021 р.)

Погоджено Методичною комісією ТЕФ КПІ ім. Ігоря Сікорського ¹ (протокол № 11 від 24.06.2021 р.)