

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Одеський державний екологічний університет

ЗАТВЕРДЖЕНО

на засіданні групи забезпечення спеціальності
від « 22 » 09 2020 року
протокол № 5
Голова групи д.т.н., проф. Мещеряков В.І.

УЗГОДЖЕНО

Декан факультету комп'ютерних наук,
управління та адміністрування
к.геогр.н., доцент Коваленко Л.Б.

СИЛЛАБУС

навчальної дисципліни

Алгоритмізація та програмування+Теорія алгоритмів

(назва навчальної дисципліни)

122 – «Комп'ютерні науки»

(шифр та назва спеціальності)

«Комп'ютерні науки»

(назва освітньої програми)

РВО «Бакалавр»

(рівень вищої освіти)

денна

(форма навчання)

1 рік н.

(рік навчання)

2 семестр

(семестр навчання)

8кр./240 год.

(кількість кредитів ЄКТС/годин)

іспит

(форма контролю)

Інформаційних технологій

(кафедра)

Одеса, 2020 р.

Автор: Кузніченко С.Д., завідувач кафедри ІТ, к.геогр.н., доцент

Поточна редакція розглянута на засіданні кафедри інформаційних технологій від 31 серпня 2020 року, протокол № 1.

Викладачі: лекційні та лабораторні заняття – Кузніченко Світлана Дмитрівна, завідувач кафедри ІТ, к.геогр.н., доцент

Перелік попередніх редакцій

Прізвища та ініціали авторів	Дата, № протоколу	Дата набуття чинності

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

<p>Мета</p>	<p>Формування у студентів теоретичних знань та практичних навичок, необхідних для роботи у наукових та виробничих підрозділах, що здійснюють обслуговування та розробку комп'ютеризованих систем, а також програмних та програмно-технічних систем, виготовлення, тестування, відлагодження та запровадження програмних продуктів</p>
<p>Компетентність</p>	<p>ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності. ЗК6. Здатність вчитися й оволодівати сучасними знаннями. ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт. СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.</p>
<p>Результат навчання</p>	<p>ПР2. Використовувати сучасний математичний апарат неперервного та дискретного аналізу, лінійної алгебри, аналітичної геометрії, в професійній діяльності для розв'язання задач теоретичного та прикладного характеру в процесі проектування та реалізації об'єктів інформатизації. ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій</p>
<p>Базові знання</p>	<p>Основні типові алгоритмічні конструкції, етапи обробки програм на ПЕОМ (редагування, трансляція, компонування), основні оператори мови Java, типи даних, рекурсивні алгоритми та алгоритми сортування масивів. Основні поняття ООП: класи, об'єкти, методи; динамічні типи даних: стеки, черги, списки, бінарні дерева та алгоритми їх створення.</p>
<p>Базові вміння</p>	<p>Використовувати основні поняття алгоритмічних</p>

	структур для побудови алгоритмів рішення задач за допомогою ЕОМ, розробляти алгоритми методом покрокового уточнення, складати лінійні та розгалужені програми мовою Java, працювати з різними типами даних, у тому числі масивами, рядками, списками, чергами, стеками, бінарними деревами, файлами і класами, відлагоджувати та виконувати Java-програму.
Базові навички	Знати основні поняття, необхідні для конструювання алгоритмів та програм; Знати основні принципи організації програм мовою програмування java
Пов'язані ссиллабуси	Алгоритмізація та програмування, частина 1
Попередня дисципліна	-
Наступна дисципліна	Об'єктно-орієнтоване моделювання
Кількість годин	лекції: 30 год.; лабораторні заняття: 30 год; самостійна робота студентів: 180 год.

2. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1 Лекційні модулі

Код	Назва модуля та тем	Кількість годин	
		аудиторні	СРС
ЗМ-Л1	<i>Процедурно-орієнтоване програмування.</i> <i>Основні поняття ООП</i>		
	• Процедурно-орієнтоване програмування. Методології розробки програм: низхідне та висхідне проектування, модульне програмування	1	3
	• Основні поняття ОПП. Класи. Об'єкти. Методи. Видимість і ініціалізація змінних	2	4
	• Методи. Оголошення і виклик методів. Сигнатура методу. Передача аргументів по посиланню та за значенням	2	4
	• Поняття конструктора. Операція new. Ключове слово this	1	4
	<i>Аналіз алгоритмів та алгоритмічні стратегії</i>		
	• Алгоритмічні стратегії. Основи теорії обчислюваності. Класи складності P і NP	2	3+5
	• Рекурсія. Особливості рекурсивних програм	2	4
• Алгоритми сортування, злиття та пошуку.	2	4	

	Використання методів класу Arrays. Генератор випадкових чисел. Комбінаторні алгоритми		
ЗМ-Л2	<i>Файлові структури даних</i> <ul style="list-style-type: none"> • Робота з рядками. Методи класу String і StringBuilder. Клас StringTokenizer. Приклади рішення завдань із використанням алгоритмів обробки рядків • Файлове введення- виведення • Робота з файлами в Java • Файлові структури даних <i>Динамічні структури даних Фундаментальні алгоритми та їх побудова</i> <ul style="list-style-type: none"> • Стеки, черги, списки та алгоритми їх оброблення • Бінарні дерева. Фундаментальні алгоритми на графах і деревах • Геометричні алгоритми • Криптографічні алгоритми 	2 1 1 2 2 2 1 1	4 2 3+5 4 4 4 3 2
ЗМ-Л3	<ul style="list-style-type: none"> • <i>Основи роботи з вікнами та графічними об'єктами</i> • Створення фреймових вікон в Java • Малювання графічних примітивів. Установка кольору і режиму малювання • Робота з текстом та з шрифтом. Позиціонування тексту 	2 2 2	8 8+5 9
Іспит			20
Разом:		30	112

Консультації:

Кузніченко Світлана Дмитрівна, **понеділок 12⁴⁵ – 14²⁰ год.** 320 ауд. НЛК№1

2.2 Практичні модулі

Код	Назва модуля та тем	Кількість годин	
		аудиторні	СРС
ЗМ-П1	1) Посильні типи даних. Робота з двомірними масивами.	5	8
	2) Рекурсія. Використання рекурсивних методів.	5	8
ЗМ-П2	3) Структури та алгоритми їх оброблення. Використання методів класу Arrays для обробки	5	8

	масивів. 4) Рядки. Використання методів класів String і StringBuilder. Організація файлового введення та виведення	5	8
ЗМ-ПЗ	5) Динамічні структури даних (стеки, черги, списки)	5	8
	6) Реалізація та алгоритми оброблення бінарних дерев.	5	8
ЗМ-ІЗ	Виконання курсової роботи		20
Разом:		30	68

Перелік лабораторій та лабораторного обладнання:

Заняття проводяться в комп'ютерних класах лабораторії кафедри інформаційних технологій з використанням програмного забезпечення Microsoft Word, Microsoft Excel, Java Developers Kit v1.5.0., BlueJ 3.1.1, IDE Eclipse

Консультації:

Кузніченко Світлана Дмитрівна, **понеділок 12⁴⁵ – 14²⁰ год.** 320 ауд. НЛК№1

2.3 Самостійна робота студента та контрольні заходи

Код	Завдання на СРС та контрольні заходи	Кількість годин	Строк проведення
ЗМ-Л1	• Підготовка до лекційних занять	26	1-6 тижні
	• Підготовка до модульної контрольної роботи № 1	5	1-6 тижні
	• Модульна контрольна робота № 1 (обов'язкова)		6 тиждень
ЗМ-Л2	• Підготовка до лекційних занять	26	7-12 тижні
	• Підготовка до модульної контрольної роботи № 2	5	7-12 тижні
	• Модульна контрольна робота № 2 (обов'язкова)		12 тиждень
ЗМ-Л3	• Підготовка до лекційних занять	25	13-15 тижні
	• Підготовка до модульної контрольної роботи № 3	5	13-15 тижні
	• Модульна контрольна робота № 3 (обов'язкова)		15 тиждень
ЗМ-П1	• підготовка до усного опитування напередодні відповідної лабораторної роботи (обов'язкове)	6	1-5 тижні
	• підготовка до захисту звіту з лабораторних робіт (обов'язковий)	5x2=10	1-5 тижні
ЗМ-П2	• підготовка до усного опитування напередодні відповідної лабораторної роботи (обов'язкове)	6	6-10 тижні
	• підготовка до захисту звіту з лабораторних робіт (обов'язковий)	5x2=10	6-10 тижні

ЗМ-ПЗ	• підготовка до усного опитування напередодні відповідної лабораторної роботи (обов'язкове)	6	11-15 тижні
	• підготовка до захисту звіту з лабораторних робіт (обов'язковий)	5x2=10	11-15 тижні
ЗМ-ІЗ	Підготовка курсової роботи	20	15 тиждень
	Підготовка до іспиту	20	
Разом:		120	

1. Методика проведення та оцінювання контрольного заходу для ЗМ-Л1

Контроль проводиться після вивчення лекційного матеріалу модуля ЗМ-Л1 в формі письмової модульної контрольної роботи МКР-1 в якій студенти відповідають на 5 запитань. Результати роботи оформлюються на окремому аркуші. Час, що виділяється на виконання МКР-1 визначається при видачі завдання і не перевищує 1 академічної години.

Критерії оцінювання результатів контрольного заходу: максимальна оцінка за контрольну роботу складає 10 балів; розподіл балів за правильні відповіді має наступний вигляд:

Номер запитання	Кількість балів за правильну відповідь
1	1
2	1
3	3
4	2
5	3
Разом:	10

Критерії оцінювання результатів контрольного заходу: відмінно (9-10 балів); добре (7-8 балів); задовільно (5-6); незадовільно (менше 5 балів).

2. Методика проведення та оцінювання контрольного заходу для ЗМ-Л2

Контроль проводиться після вивчення лекційного матеріалу модуля ЗМ-Л2 в формі письмової модульної контрольної роботи МКР-2 в якій студенти відповідають на 5 запитань. Результати роботи оформлюються на окремому аркуші. Час, що виділяється на виконання МКР-2 визначається при видачі завдання і не перевищує 1 академічної години.

Максимальна оцінка за контрольну роботу складає 10 балів. Розподіл балів за правильні відповіді має наступний вигляд:

Номер запитання	Кількість балів за правильну відповідь
1	1
2	1
3	2
4	3
5	3
Разом:	10

Критерії оцінювання результатів контрольного заходу: відмінно (9-10 балів); добре (7-8 балів); задовільно (5-6); незадовільно (менше 5 балів).

3. Методика проведення та оцінювання контрольного заходу для ЗМ-Л3

Контроль проводиться після вивчення лекційного матеріалу модуля ЗМ-Л1 в формі письмової модульної контрольної роботи МКР-1 в якій студенти відповідають на 4 запитання. Результати роботи оформлюються на окремому аркуші. Час, що виділяється на виконання МКР-1 визначається при видачі завдання і не перевищує 1 академічної години.

Максимальна оцінка за контрольну роботу складає 10 балів; розподіл балів за правильні відповіді має наступний вигляд:

Номер запитання	Кількість балів за правильну відповідь
1	1
2	4
3	1
4	4
Разом:	10

Критерії оцінювання результатів контрольного заходу: відмінно (9-10 балів); добре (7-8 балів); задовільно (5-6); незадовільно (менше 5 балів).

4. Методика підсумкового оцінювання контрольних заходів для всіх лекційних модулів

Підсумкова оцінка за всі лекційні модулі дорівнює сумі набраних балів за лекційні модулі ЗМ-Л1, ЗМ-Л2, ЗМ-Л3 яка не може перевищувати 30 балів. Якщо обов'язкові заходи контролю виконуються студентом після строків, визначених у програмі навчальної дисципліни, кількість балів, що може отримати студент, не може перевищувати 60% від максимально можливої для цієї форми контролю.

5. Методика проведення та оцінювання контрольного заходу для ЗМ-П1, ЗМ-П2 та ЗМ-П3

Розподіл балів, які може отримати студент за практичні модулі наведено в таблиці.

ЗМ-П1	1) Посильні типи даних. Робота з двомірними масивами.	6
	2) Рекурсія. Використання рекурсивних методів.	6
ЗМ-П2	3) Структури та алгоритми їх оброблення. Використання методів класу Arrays для обробки масивів.	6
	4) Рядки. Використання методів класів String і StringBuilder. Організація файлового введення та виведення	6

ЗМ-ПЗ	5) Динамічні структури даних (стеки, черги, списки)	8
	6) Реалізація та алгоритми оброблення бінарних дерев.	8

Контроль по кожній лабораторній роботі проводиться в формі:

- *усного опитування* при підготовці до кожної лабораторної роботи з метою допуску до її виконання (максимальна кількість балів – 3),
- *захисту результатів* лабораторної роботи наведених у звіті до лабораторної роботи (кількість запитань залежить від ходу виконання студентом роботи і якості звіту, максимальна кількість балів – 3 за лабораторні роботи ЗМ-П1 і ЗМ-П2 та 5 – за лабораторні роботи ЗМ-П3).

Для кожної лабораторної роботи, якщо студент за *усне опитування* одержав менше 2 балів він не допускається до виконання роботи.

Підсумковою оцінкою за кожну лабораторну роботу буде сума балів за *усне опитування* і *захист результатів*. Якщо обов'язкові заходи контролю виконуються студентом після строків, визначених у програмі навчальної дисципліни, кількість балів, що може отримати студент, не може перевищувати 60% від максимально можливої для цієї форми контролю.

Підсумковою оцінкою за кожний практичний модуль буде сума балів за всі лабораторні роботи.

б. Методика оцінювання індивідуального завдання для ЗМ-ІЗ

Індивідуальним завданням ЗМ-ІЗ є виконання курсової роботи. Мета курсової роботи – виробити у студентів практичні навички реалізації прикладних задач на ЕОМ, у т. ч. підготовка та розробка вхідної інформації, розробка алгоритму та програми на мові високого рівня Java, відлагодження програми, виконання розрахунків. Завдання курсової роботи подані за варіантами і надаються студентам у вигляді елементів електронної схеми.

Виконана курсова робота надається викладачу у паперовому та електронному вигляді. Паперовий варіант курсової роботи повинен мати титульний аркуш, зміст і виконані завдання курсової роботи відповідно до вимог, встановлених в методичних вказівках по виконанню курсової роботи. Курсова робота повинна бути захищена перед початком заліково-екзаменаційної сесії. Перед допуском до захисту студентів електронні версії наданих текстових документів обов'язково перевіряються на оригінальність із встановленням частки оригінального тексту згідно з п. 2.3 Тимчасового положення про заходи щодо недопущення академічного плагіату в ОДЕКУ.

Максимальна сума балів за виконання курсової роботи – 30.

Терміни виконання курсової роботи

№ з/п	Назва етапів курсової роботи	Строк виконання етапів роботи Тиждень
-------	------------------------------	--

1.	Отримання варіанту курсової роботи.	2
2.	Створення частини кода програми: оголошення класу, полів класу та методу main(). Виділення пам'яті для масивів розрахункових величин.	3,4
3.	Створення методу розрахунку параметрів електричної схеми.	5,6,7
4.	Створення методу виводу розрахункових даних в вихідний файл. Уведення вхідних даних із клавіатури за допомогою класа InputStreamReader.	8,9,10
5.	Розробка методу побудови графіків: побудова осей координат, написання методу розрахунку ціни поділки, виведення написів під осями і насічками.	11,12
6.	Виведення графіків у фреймове вікно. Побудова легенди. Відладка програми.	13
7.	Оформлення курсової роботи.	14
8.	Захист курсової роботи.	15

При умові, що студент своєчасно виконував усі етапи курсової роботи та оформив її згідно вимогам, він має можливість отримати 60% за її виконання без усного захисту, тобто 18 балів.

7. Методика оцінювання за всіма змістовними модулями

Підсумковою оцінкою за всіма змістовними модулями (ОЗ) буде сума балів за лекційні модулі, практичні модулі та за виконання курсової роботи.

8. Методика проведення та оцінювання підсумкового контрольного заходу

Підсумковий контрольний захід проводиться у формі іспиту. Студент вважається допущеним до підсумкового контролю з навчальної дисципліни, якщо він виконав всі види робіт поточного контролю, що передбачені програмою дисципліни і набрав за накопичувальною системою суму балів не менше 50% (тобто більше 35 балів) від максимально можливої за практичну частину дисципліну, у т.ч. захистив курсову роботу.

Час, що виділяється на виконання іспиту не перевищує 2 академічні години.

Оцінювання письмових відповідей студентів проводиться у відповідності з Положенням «Про критерії оцінки знань студентів в ОДЕКУ». Загальна екзаменаційна оцінка (бал успішності) еквівалента відсотку правильних відповідей із загального обсягу питань екзаменаційного білету, який складає 20 питань. Тобто кожне питання оцінюється в 5 балів. Максимальна сума балів становить 100.

Таблиця для визначення кількісних показників успішності за підсумками іспиту

За системою університету (у відсотках)	Визначення
90 – 100	відмінне виконання лише з незначною кількістю помилок
82 – 89	вище середнього рівня з кількома помилками
74 – 81	В загальному правильна робота з певною кількістю грубих помилок
64 – 73	непогано, але зі значною кількістю помилок
60 – 63	виконання задовольняє мінімальні критерії
35 – 59	з можливістю перескласти
1 – 34	з обов'язковим повторним курсом навчання

9. Методика підсумкового оцінювання за дисципліну

Накопичена підсумкова оцінка засвоєння студентом навчальної дисципліни, що закінчується іспитом, розраховується як:

$$ПО = 0,5ОПК + 0,5ОЗ. \quad (1)$$

де: ОПК – кількісна оцінка (у відсотках від максимально можливої) заходу підсумкового контролю;

ОЗ – кількісна оцінка (у відсотках від максимально можливої) заходів за всіма змістовними модулями.

Шкала відповідності оцінок в системах університету, національній системі та системі ЄКТАС наведена в таблиці.

За шкалою ECTS	За національною системою	Бал успішності
	для іспиту	
A	5 (відмінно)	90–100
B	4 (добре)	82–89,9
C	4 (добре)	74–81,9
D	3 (задовільно)	64–73,9
E	3 (задовільно)	60–63,9
FX	2 (незадовільно)	35–59,9
F	2 (незадовільно)	1–34,9

3. РЕКОМЕНДАЦІЇ ДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

Рекомендується наступний порядок вивчення дисципліни «Алгоритмізація та програмування+Теорія алгоритмів»:

–зміст кожної теми курсу вивчається за допомогою навчальної та методичної літератури, що наведена в списку;

–після засвоєння змісту кожної теми курсу потрібно відповісти на «запитання самоперевірки», що наведені у силлабусу і відповідній літературі;

–якщо виникли питання при вивченні теоретичного матеріалу або при виконанні контрольних робіт, то потрібно звернутись до викладача, який проводив лекції.

Модуль ЗМ-Л1 «Процедурно-орієнтоване програмування. Основні поняття ООП. Аналіз алгоритмів та алгоритмічні стратегії»

3.1.1. Повчання

Розділи модуля ЗМ-Л1 формують у студентів уявлення про загальні принципи процедурно-орієнтованого програмування та основні поняття об'єктно-орієнтованого програмування, а саме: методології розробки програм, особливості створення рекурсивних програм, алгоритми сортування, злиття та пошуку, комбінаторні алгоритми.

При вивченні цих розділів необхідно звернути увагу на особливості оголошення класів, об'єктів та методів в Java.

Програма ЗМ-Л1. Процедурно-орієнтоване програмування. Основні поняття ООП. Аналіз алгоритмів та алгоритмічні стратегії

- Процедурно-орієнтоване програмування. Методології розробки програм: низхідне та висхідне проектування, модульне програмування [1, с.200-202]
- Основні поняття ОПП. Класи. Об'єкти. Методи. Видимість і ініціалізація змінних [1, с. 202-226]
- Методи. Оголошення і виклик методів. Сигнатура методу. Передача аргументів по посиланню та за значенням [1, с.208-212]
- Поняття конструктора. Операція new. Ключове слово this [1, с.220-224]
- Алгоритмічні стратегії. Основи теорії обчислюваності. Класи складності P і NP [3, с.51-57]
- Рекурсія. Особливості рекурсивних програм [1, с.229-218; 3, с.64-84]
- Алгоритми сортування, злиття та пошуку. Комбінаторні алгоритми [1, с.236-245]
- Використання методів класу Arrays. Генератор випадкових чисел [1, с.245-248]

3.1.2. Питання для самоперевірки

Запитання, що входять до тестів до модуля ЗМ-Л1 і являють собою необхідний мінімум знань, який потрібний для засвоєння дисципліни «Алгоритмізація та програмування+Теорія алгоритмів», наведені нижче:

1. Приведіть формат оголошення методу.
2. Що таке сигнатура методу? Які з наведених нижче методів несумісні в одному класі?

```
void get() {}
int get() {}
void get(int x) {}
void get(int y) {}
public int get() {}
private int get() {}
```
3. У яких випадках у тілі методу використовується return-вираження?
4. Чим відрізняються формальні і фактичні параметри?
5. Дайте визначення глобальним і локальним змінним. Приведіть приклади.
6. У яких випадках при оголошенні полів і методів класу використовуються наступні модифікатори доступу: private, public, static, final?
7. Дайте визначення передачі параметрів в метод за значенням і за посиланням. Як відбувається передача параметрів в Java?
8. Як організувати генератор випадкових чисел за допомогою методів класу Random? Приведіть приклади.
9. Для чого призначений метод System.arraycopy()? Приведіть приклади його використання.
10. Опишіть призначення і формати запису методів класу Arrays, призначених для порівняння, сортування і заповнення масивів.
11. Як організувати бінарний пошук у відсортованому масиві за допомогою методів класу Arrays? Приведіть приклади.
12. Що неправильно в наступній програмі? Виправте її.

```
public class SomethingIsWrong {
public static void main(String[] args) {
Rectangle myRect;
myRect.width = 40;
myRect.height = 50;
System.out.println("myRect's area is " + myRect.area()); } }
```
13. Фрагмент коду створює один об'єкт Point і один об'єкт Rectangle. Скільки посилань до цих об'єктів існує після того, як код буде виконаний? Чи буде який-небудь з об'єктів кандидатом для «збирання сміття»?

```
...
Point point = new Point(2,4);
Rectangle rectangle = new Rectangle(point, 20, 20);
point = null;
...
```

14. Використовуючи поняття сигнатури методу, визначте, які методи несумісні один з одним в одному класі

```
class Complex {  
    void set(double re) {}  
    void set(double re, double im) {}  
    void set(Complex re) {}  
    void set(double x, double y) {}  
    public void set(Complex re) {} }  
}
```

15. Визначте, що буде виведено на екран при компіляції і запуску наступного коду:

```
public class Example{  
    static int i;  
    public static void main(String[] args){  
        System.out.print(i);  
    }  
}
```

16. Визначте, що буде виведено на екран при компіляції і запуску наступного коду:

```
public class MyClass {  
    {System.out.print("1");}  
    static{System.out.print("2");} 227  
    MyClass(){System.out.print("3");}  
    public static void main(String[] args) {  
        System.out.print("4");  
    }  
}
```

17. Визначте, що буде виведено на екран при компіляції і запуску наступного коду:

```
class My{  
    int x,y;  
    My(int x, int y){this.x=x; this.y=y;}  
    My(int x){this(x,-2);}  
    My(){this(1,2);}  
    public static void main (String[] args){  
        My a=new My(5);  
        System.out.println((a.x+a.y)); }  
}
```

18. Дано метод і три програмних блоки, кожен з яких звертається до зазначеного методу. Слід перевірити правильність написання кожного з цих блоків; якщо він правильний – вказати, що буде надруковано.

```
static boolean flag (int a, int b){  
    return (4*b!=a+b)?true:false;  
}  
  
.  
.  
.  
1. int x=(flag (6,2))? 5:-5; System.out.println(x);  
2. int x=2; double y=8.; System.out.println(flag(x,y));  
3. int m=1; for(int i=3;i<6;i++)  
    System.out.println(flag (i,m));
```

19. Маючи клас з назвою Point, напишіть деякий код, який створює екземпляр цього класу, ініціалізує його дві змінні і потім виводить на екран значення кожної зі змінних.

```
public class Point {  
    public int x = 0;  
    public int y = 0;  
    public Point(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
}
```

20. Визначте клас Point, що задає точку на площині. Клас повинен містити кілька конструкторів. Реалізувати методи для обчислення відстані від точки до початку координат, відстані між двома точками, площі трикутника побудованого за трьома точками.
21. Визначте клас Vector, що дозволяє виконувати над векторами на площині такі операції: додавання, віднімання, множення на число і обчислення скалярного добутку.
22. Створіть клас Complex, який містить методи, що дозволяють виконувати над комплексними числами арифметичні операції додавання, віднімання, множення і виводити їх на екран.
23. Визначте клас Matrix розмірності (mхn). Клас повинен містити кілька конструкторів. Оголосити масив об'єктів. Передайте об'єкти в метод, що міняє місцями рядки з максимальним і мінімальним елементами k-го стовпця. Створіть метод, який змінює i-ту матрицю шляхом піднесення її до квадрата.
24. Опишіть рекурсивний метод, який для двох заданих натуральних чисел знаходить найбільший спільний дільник за алгоритмом Евкліда.
25. Підрахуйте, скільки разів буде потрібно повторно обчислити четвертий елемент послідовності Фібоначчі для обчислення п'ятнадцятого елемента.
26. Напишіть рекурсивний метод знаходження суми перших n членів арифметичної прогресії.
27. Напишіть рекурсивний метод виведення цифр цілого додатного числа n в зворотному порядку.
28. Опишіть рекурсивний метод знаходження суми чисел будь-якого натурального числа.
29. Визначте число n. Додайте всі цифри числа n, потім всі цифри знайденої суми і повторіть ці дії до тих пір, поки не отримаєте цифру, яка називається цифровим коренем числа. Напишіть рекурсивний метод обчислення цифрового кореня числа n.
30. Напишіть рекурсивний метод переведення натурального числа з десяткової системи числення в двійкову.

3.2. Модуль ЗМ-Л2 «Файлові структури даних. Динамічні структури даних. Фундаментальні алгоритми та їх побудова»

3.2.1. Повчання

Розділи модуля ЗМ-Л2 формують у студентів уявлення про структури даних, а саме: файлові та динамічні структури даних, алгоритми їх побудови.

При вивченні цих розділів необхідно звернути увагу на стеки, черги, списки та алгоритми їх оброблення, а також на файлові структури даних.

Програма ЗМ-Л2. Файлові структури даних. Динамічні структури даних. Фундаментальні алгоритми та їх побудова

- Робота з рядками. Методи класа String і StringBuilder. Клас StringTokenizer. Приклади рішення завдань із використанням алгоритмів обробки рядків [1, с.251-284]
- Файлове введення- виведення [1, ст.293-300]
- Робота з файлами в Java [1, с.288-293]
- Файлові структури даних [1, ст.288-290]
- Стеки, черги, списки та алгоритми їх оброблення [1, с.307-332]
- Бінарні дерева. Фундаментальні алгоритми на графах і деревах [4, с.467-490]
- Геометричні алгоритми [4, с.548-601]
- Криптографічні алгоритми [3, с.90-93]

3.2.2. Питання для самоперевірки

Запитання, що входять до тестів до модуля ЗМ-Л2 і являють собою необхідний мінімум знань, який потрібний для засвоєння дисципліни «Алгоритмізація та програмування+Теорія алгоритмів», наведені нижче:

1. За допомогою яких методів можна перетворити в рядок змінні примітивних типів і масив символів? Як перетворити з String в примітивний тип? Приведіть приклади.
2. Які методи класу String дозволяють провести заміну регістра букв?
3. У яких випадках використовується клас StringBuffer? Як можна встановити ємність буфера для об'єкту класу StringBuffer? Опишіть конструктори даного класу.
4. Дайте короткий опис конструкторам і методам класу StringTokenizer.
5. У чому відмінність між наступними фрагментами коду:

```
1. String s1 = "abc";  
String s2 = new String("abc");  
boolean result = (s1==s2);
```

```
2. String s1 = new String("abc");
```



```
String s2 = new String("abc");
boolean result = (s1.equals(s2));
```

6. Як в Java можна організувати файлові потоки введення-виведення?
7. Що собою уявляє стек. Де така структура організації даних використовується?
8. Дайте характеристику АДД «черга». Які особливості створення черги на основі масиву?
9. Поясніть відмінності однозв'язного і двозв'язного списків. Яким чином вони задаються?
- 10.Що собою представляє структура бінарне дерево?
- 11.Яким чином визначається глибина бінарного дерева?
- 12.Припустимо, з клавіатури було введено 45 57,8 789, а потім натиснута клавіша Enter. Чому дорівнюватимуть змінні після виконання наступного коду:

```
Scanner input = new Scanner(System.in);
int intValue = input.nextInt();
double doubleValue = input.nextDouble();
String line = input.nextLine();
```

- 13.Припустимо, що з клавіатури ввели 45, натиснули клавішу Enter, 57,8, натиснули клавішу Enter, 789, і натиснули клавішу Enter. Чому дорівнюватимуть змінні після виконання наступного коду:

```
Scanner input = new Scanner(System.in);
int intValue = input.nextInt();
double doubleValue = input.nextDouble();
String line = input.nextLine();
```

- 14.Відомо, що можна перевірити, чи є рядок підрядком іншого рядка за допомогою методу indexOf () класу String. Розробіть власний метод, що виконує цю функцію. Напишіть програму, яка пропонує користувачеві ввести два рядки, і перевіряє, чи є перший рядок підрядком другого.
- 15.Напишіть метод, який підраховує кількість букв в рядку (не використовуючи метод length) з таким заголовком:

```
public static int countLetters(String s)
```

Напишіть тестову програму, яка пропонує користувачеві ввести рядок і друкує кількість букв в рядку.
- 16.Деякі веб-сайти накладають певні правила на паролі, що вводяться користувачами. Напишіть метод, який перевіряє, чи є рядок допустимим паролем. Припустимо, що пароль повинен відповідати таким правилам:
 - містити не менше восьми символів;
 - складатися тільки з букв і цифр;
 - містити не менше двох цифр.

17. Напишіть програму, яка пропонує користувачеві ввести пароль і відображає Valid Password, якщо правила дотримані або Invalid Password в іншому випадку.
18. Напишіть метод, який підраховує кількості кожної цифри в рядку, використовуючи такий заголовок:
- ```
public static int[] count(String s)
```
- Метод підраховує, скільки разів цифра з'являється в рядку і повертає масив з десяти елементів, кожен з яких містить значення лічильника для відповідної цифри. Так, наприклад, після виконання `int[] counts = count("12203AB3")`, `counts[0] = 1`, `counts[1] = 1`, `counts[2] = 2`, `counts[3] = 2`.
- Напишіть тестову програму, яка пропонує користувачеві ввести рядок і відображає кількості входжень кожної цифри в рядок.
19. Напишіть метод, який повертає відсортований рядок і має такий заголовок:
- ```
public static String sort(String s)
```
- Наприклад, `sort("acb")` поверне `abc`. Напишіть тестову програму, яка пропонує користувачеві ввести рядок і виводить відсортований рядок.
20. Напишіть метод, який перевіряє, чи є два слова анаграмою. Два слова є анаграмою, якщо вони містять ті ж букви в будь-якому порядку. Так, наприклад, слова крона і норка є анаграмою. Заголовок методу:
- ```
public static boolean isAnagram(String s1, String s2)
```
- Напишіть тестову програму, яка пропонує користувачеві ввести два рядки і, якщо вони анаграми, відображає `two strings are anagrams` або `two strings are not anagrams` в іншому випадку.
21. Напишіть програму, яка зчитує з текстового файлу рядок і виводить на екран кількість великих букв в цьому рядку.
22. Напишіть рекурентний метод для видалення непорожнього каталогу на диску.

### **3.3. Модуль ЗМ-ЛЗ «Основи роботи з вікнами та графічними об'єктами»**

#### **3.3.1. Повчання**

Розділи модуля ЗМ-ЛЗ формують у студентів уявлення про роботу з віконними та графічними об'єктами в Java, а саме: малювання ліній, прямокутників, окружностей і овалів; установка кольору і режиму малювання; виведення у вікно тексту та використання шрифтів.

Програма ЗМ-ЛЗ. Основи роботи з вікнами та графічними об'єктами

- Створення фреймових вікон в Java [6, с.121-128]
- Малювання графічних примітивів [6, с.128-152]

- Установка кольору і режиму малювання [6, с.128-130]
- Робота з текстом та з шрифтом. Позиціонування тексту [6, с.131-137]

### 3.3.2. Питання для самоперевірки

Запитання, що входять до тестів до модуля ЗМ-ЛЗ і являють собою необхідний мінімум знань, який потрібний для засвоєння дисципліни «Алгоритмізація та програмування+Теорія алгоритмів», наведені нижче:

1. Дайте визначення фреймовому вікну. Наведіть програмний код його створення в Java.
2. Які функції потрібно застосувати для малювання прямокутників та овалів?
3. Яким чином можна задати власний колір?
4. Які класи та методи використовуються в Java для роботи зі шрифтом? Яким чином можна задати власний шрифт?
5. Які класи та методи використовуються в Java для позиціонування тексту? Наведіть приклади.
6. Використовуючи динамічну структуру «стек» для зберігання символів, визначте, чи є задана фраза паліндромом, тобто такою, що однаково читається в обох напрямках (зліва направо і справа наліво).
7. Використовуючи динамічну структуру список, збережіть ряд чисел. Видалити елементи, які повторюються.
8. Сформууйте динамічну структуру черга, елементами якої є цифри. Вилучаючи елементи з черги, надрукуйте їх двійкові еквіваленти.
9. Побудуйте динамічний список з елементів цілого типу за допомогою генератора випадкових чисел. Знайдіть середньо арифметичне значення елементів списку.
10. Задайте два динамічних списки. Перевірте їх на рівність.
11. Використовуючи динамічну структуру список, підрахуйте кількість додатних і від'ємних чисел в списку.
12. Використовуючи динамічну структуру «стек», зашифруйте зміст текстового файла.
13. Сформууйте чергу з елементів цілого типу. Парні елементи зведіть в квадрат. Роздрукуйте вихідну і результуючі черги.
14. Використовуючи динамічну структуру список, перевірте, чи є він упорядкованим набором чисел.
15. Використовуючи динамічну структуру черга, перевести введену послідовність цифр в слово, що складається з кодів Unicode.

### **3.4. Модуль ЗМ-П1 „Практичний модуль”**

При вивченні практичного модуля студенти набувають уміння програмування алгоритмів обробки двовимірних масивів, упорядкування двовимірних масивів методом вставки, вибору та лінійного сортування, а також роботи з рекурсивними та статичними методами в Java. [21-24].

При вивченні цього модуля необхідно звернути увагу на практичне застосування одержаних теоретичних знань про алгоритми обробки багатовимірних масивів, оголошення та виклик методів, особливості створення рекурсивних програм.

Перевірка якості засвоєних знань і одержаних навичок при вивченні цього модуля здійснюється викладачем під час проведення лабораторних занять шляхом усного опитування з наведених для теоретичного модуля ЗМ-Л1 питань і перевіркою якості виконання лабораторної роботи.

### **3.5. Модуль ЗМ-П2 „Практичний модуль”**

При вивченні практичного модуля студенти набувають уміння обробки масивів за допомогою методів класу Arrays та розробка власних методів, а також роботи з текстовими файлами та методами класів String, StringBuffer і StringTokenizer. [21-24].

При вивченні цього модуля необхідно звернути увагу на практичне застосування одержаних теоретичних знань про методи класів Arrays, String, StringBuffer і StringTokenizer.

Перевірка якості засвоєних знань і одержаних навичок при вивченні цього модуля здійснюється викладачем під час проведення лабораторних занять шляхом усного опитування з наведених для теоретичного модуля ЗМ-Л2 питань і перевіркою якості виконання лабораторної роботи.

### **3.6. Модуль ЗМ-П2 „Практичний модуль”**

При вивченні практичного модуля студенти набувають уміння роботи з динамічними типами даних в Java: списками, чергами, стеками, деревами та алгоритмами їх оброблення. [21-24].

При вивченні цього модуля необхідно звернути увагу на практичне застосування одержаних теоретичних знань про абстрактні типи даних: стек, черга, список, дерева та алгоритми їх створення та обробки.

Перевірка якості засвоєних знань і одержаних навичок при вивченні цього модуля здійснюється викладачем під час проведення лабораторних занять шляхом усного опитування з наведених для теоретичного модуля ЗМ-Л3 питань і перевіркою якості виконання лабораторної роботи.

## 4. ПИТАННЯ ДО ЗАХОДІВ ПОТОЧНОГО ТА ПІДСУМКОВОГО КОНТРОЛЮ

### 4.1. Тестові завдання до модульної контрольної роботи модуля ЗМ-Л1

1. Дайте визначення класу. Який формат має оголошення класу? [1, с.202-204]
2. Використовуючи поняття сигнатури методу, визначте, які методи несумісні один з одним в одному класі... [1, с.209-212]
3. Дано метод і три програмних блоку, кожен з яких звертається до зазначеного методу. Слід перевірити вірність написання кожного з цих блоків; якщо він вірний – вказати, що буде надруковано в якості результату. [1, с.208-213]
4. Визначте результат виклику наступного рекурентного методу... [1, с.229-251]
5. Напишіть програму, яка обчислює задану характеристику цілої прямокутної матриці `nхm` елементів. Програма повинна містити наступні методи:
  - метод заповнення матриці цілими двозначними числами (використовувати метод `Random ()` з пакету `java.util`);
  - метод виведення матриці на екран;
  - метод, який розраховує задану характеристику в одновимірному масиві. [1, с.202-229]
6. Для чого призначений модифікатор членів класу `final`? [1, с.213-220]
7. Коротко опишіть особливості зберігання даних в пам'яті. Які області пам'яті для цього виділені? [1, с.205-208]
8. Який метод класу `Arrays` призначений для заповнення масиву значенням? Наведіть формати його записи. [1, с.245-248]
9. Дайте визначення сигнатури методу. Які методи називаються перевантаженими? [1, с.208-213]
10. Дайте визначення статичного методу. Які особливості застосування статичних методів. [1, с.213-215]
11. Як організувати генератор випадкових чисел за допомогою методів класу `Random`? Наведіть приклади. [1, с.245-248]
12. Дайте коротке визначення основних понять об'єктно-орієнтованого програмування: об'єкту, класу і методу. [1, с.202-204]
13. Які значення за замовчуванням присвоюються членам класу примітивних типів? [1, с.213-218]
14. Дайте визначення членам класу. Наведіть приклади різних членів класу. [1, с.202-205]
15. Що таке «збирач сміття» в Java і для чого він призначений? [1, с.209-213]
16. Який метод класу `Arrays` призначений для порівняння масивів? Наведіть формат його записи. [1, с.245-248]

17. Дайте визначення поля класу. [1, с.202-205]
18. Для чого призначений модифікатор членів класу `private`? [1, с.213-218]
19. Що таке `return`-вираз і для чого воно призначене? [1, с.208-215]
20. Дайте визначення передачі параметрів в метод за значенням і за посиланням. Як відбувається передача параметрів в Java? [1, с.215-218]
21. Дайте визначення статичного методу. Які особливості застосування статичних методів. [1, с.213-218]
22. Який формат має заголовок методу `main ()`? [1, с.215-220]
23. Для чого призначений метод `System.arraycopy ()`? [1, с.245-248]
24. Що таке фактичні й формальні параметри? Яке відповідність повинно бути між ними? [1, с.215-218]
25. Як організувати бінарний пошук в відсортованому масиві за допомогою методів класу `Arrays`? [1, с.245-248]
26. Для чого призначений модифікатор членів класу `static`? [1, с.213-215]
27. Дайте визначення рекурсії. Який метод називається рекурсивним? [1, с.229-236]
28. Який формат має виклик методу? [1, с.212-213]
29. Який метод класу `Arrays` призначений для сортування масивів? Наведіть формати його записи. [1, с.245-248]
30. Опишіть всі етапи створення та ініціалізації об'єктів. [1, с.205-208]
31. Для чого призначені методи. Який формат має оголошення методу? [1, с.209-212]
32. Дайте визначення глобальної і локальної змінної. [1, с.204-205]
33. Як організувати генератор випадкових чисел за допомогою методів класу `Random`? [1, с.245-248]
34. Для чого призначений модифікатор членів класу `public`? [1, с.213-218]

#### **4.2. Тестові завдання до модульної контрольної роботи модуля ЗМ-Л2**

1. Дайте визначення конструктору. Його призначення і синтаксис запису. [1, с.220-224]
2. Опишіть яким чином можна організувати консольне введення даних за допомогою класу `Scanner`. [1, с.293-300]
3. Дайте характеристику класу `StringBuffer`. Його відмінність від класу `String`. [1, с.272-278]
4. Які методи дозволяють перетворити дані з типу `String` в примітивний числовий тип? [1, с.269-272]
5. Які класи і методи дозволяють здійснити форматне виведення даних? [1, с.293-300]
6. Дайте визначення конструктору. Що таке конструктор «за замовчуванням»? [1, с.220-224]
7. Дайте характеристику класу `String`. Його призначення і основні конструктори. [1, с.251-253]

8. Як організувати в Java введення даних з файлу? [1, с.293-300]
9. Опишіть етапи створення об'єкта. Призначення операції new. [1, с.205-208]
10. Опишіть класи пакету java.io, які використовуються для запису в текстовий файл. [1, с.293-300]
11. Дайте коротку характеристику класу StringBuffer. Його відмінність від класу String. [1, с.272-278]
12. Дайте визначення конструктору. Його подібності та відмінності від методів. [1, с.220-224]
13. Як правильно організувати консольне введення з використанням класу InputStreamReader? [1, с.284-300]
14. Для чого призначена операція new? [1, с.205-208]
15. Як організувати в Java виведення даних в файл? [1, с.293-300]
16. Дайте характеристику класу String. Його призначення і основні конструктори. [1, с.251-253]
17. Визначте, що буде виведено на екран в наступній програмі. [1, с.205-224]

```
class My {
 int x, y;
 My (int x, int y) {this.x = x; this.y = y;}
 My (int x) {this (x, -2);}
 My () {this (1,2);}
 public static void main (String [] args) {
 My a = new My (5);
 System.out.println ((a.x + a.y)); }}
```

18. Напишіть програму, в якій здійсніть введення з клавіатури за допомогою класу InputStreamReader наступних змінних: String name, long h, char [] c. [1, с.278-300]
  19. Яке значення отримає змінна s, після виконання наступного фрагмента програми. [1, с.253-278]
- ```
class Str {
    public static void main (String [] arg) {
        String s = "ABCDEF";
        String s1 = s.substring (2,4);
        s = new StringBuffer (s) .append (s1) .toString ();
        System.out.println (s); }}
```
20. Розробіть програму, в якій опишіть клас з ім'ям STUDENT, що містить наступні поля: прізвище; ім'я; батькові; курс і конструктор з параметрами, що дозволяє створити екземпляр класу STUDENT з ініціалізацією полів. Виведіть поля створеного екземпляра класу в текстовий файл. [1, с.202-229]

4.3. Тестові завдання до модульної контрольної роботи модуля ЗМ-ЛЗ

1. Як в Java можна організувати малювання у вікні? Який метод слід використовувати і які пакети імпортувати? [6, с.121-128]

2. Як в Java можна створити фреймове вікно? [6, с.121-128]
3. Які методи визначені в класі Graphics для малювання кіл і овалів? Наведіть приклад. [6, с.128-152]
4. Які методи визначені в класі Graphics для малювання ліній? Наведіть приклад. [6, с.128-152]
5. Які методи визначені в класі Graphics для малювання квадратів і прямокутників? Наведіть приклад. [6, с.128-152]
6. Як можна задати колір виведення зображень в Java? [6, с.128-130]
7. Як задати шрифт і стиль виведення тексту в Java? [6, с.131-137]
8. Опишіть призначення і основні методи класу FontMetrics. [6, с.131-137]
9. Якими є принципи побудови динамічної структури «стек»? Опишіть API-інтерфейс для стека. [1, с.310-318]
10. Яким є принцип побудови динамічної структури «список»? Як можна створити пов'язаний список? [1, с.325-332]
11. Якими є принципи побудови динамічної структури «чергу»? Як можна організувати чергу на основі масиву? [1, с.321-325]
12. Якими є принципи побудови динамічної структури «стек»? Як організувати стек на основі масиву? [1, с.310-318]
13. Яким є принцип побудови динамічної структури «стек»? Як організувати стек на основі пов'язаного списку? [1, с.310-318]
14. Опишіть алгоритми видалення і вставки елемента в зв'язний список. [1, с.325-332]
15. Який принцип побудови динамічної структури «список»? Як можна переглянути елементи списку? [1, с.325-332]
16. Якими є принципи побудови динамічної структури «чергу»? Як можна організувати чергу на основі зв'язного списку? [1, с.325-332]
17. Перерахуйте подібності та відмінності динамічних структур типу «лінійний список», «стек», «чергу». [1, с.309-332]
18. Опишіть, що собою являє пов'язаний список? У чому його переваги перед масивами? [1, с.325-332]
19. Якими є принципи побудови динамічної структури «черга»? Опишіть API - інтерфейс для реалізації черги. [1, с.325-332]
20. Напишіть програму виведення зазначеного нижче зображення і тексту у фреймове вікно. Пунктиром позначені допоміжні лінії (їх виводити не треба). Дано $x = 300$, $y = 300$, $w = 100$. Квадрат зафарбувати чорним кольором, лінії - білі. [6, с.121-137]
21. Створити програму-клієнт, яка поміщає в стек дані, обробляє їх за вказаним алгоритмом і виводить на екран. [1, с.307-334]
22. Створити програму-клієнт, яка поміщає в чергу дані, обробляє їх за вказаним алгоритмом і виводить на екран. [1, с.307-334]

4.4. Тестові завдання до іспиту

1. Який вигляд має заголовок головного методу `main()`? [1, с.208-212]
2. Визначте які методи несумісні в одному класі за сигнатурою... [1, с.208-212]
3. Дано метод та фрагмент коду програми, який звертається до указанного методу. Визначте, результат виконання програми... [1, с.208-220]
4. Яку команду слід використати, щоб імпортувати до класу пакет? [1, с.202-205]
5. Визначте результат виконання наступної програми... [1, с.202-229]
6. За допомогою якого із перелічених методів можна побудувати прямокутник [6, с.128-152]
7. Які значення приймуть елементи масиву `a[]` після виконання наступних методів... [1, с.245-248]
8. Визначте результат виклику наступного рекурентного методу... [1, с.229-251]
9. Яке значення отримає строкова змінна `s`, після виконання наступного фрагменту програми: [1, с.251-272]
10. Який заголовок буде мати метод, який обчислює в одномірному цілому масиві суму елементів більших заданого цілого числа? [1, с.208-220]
11. Як в Java здійснюється передача параметрів в метод? [1, с.215-218]
12. Який пакет необхідно імпортувати для організації файлового введення/ виводу? [1, с.251-270]
13. Який з перерахованих методів слід використовувати для побудови прямокутника за заданими координатами і довжинам сторін? [6, с.128-152]
14. Що використовується для опису поведінки об'єкта? [1, с.202-208]
15. Визначте який з даних методів викликається в наступному рядку: `MyClass.myMethod()`; [1, с.208-224]
16. Сигнатура методу визначається... [1, с.208-212]
17. Аргументи, які записані в заголовку методу називаються ... [1, с.215-218]
18. За допомогою методів якого класу можна провести розбір рядків на окремі лексеми(слова)? [1, с.282-286]
19. Для запису рядків в текстовий файл у якому порядку і які слід використовувати класи? [1, с.293-300]
20. Для явного посилання на об'єкт слід використовувати ключове слово... [1, с.205-226]
21. За допомогою якої операції можна створити екземпляр класу: [1, с.205-208]
22. Назвіть оператор переривання? [1, с.124-127]
23. За допомогою якого із перелічених методів можна побудувати зафарбоване коло? [6, с.128-152]

24. Який заголовок буде мати метод, який знаходить в одномірному дійсному масиві максимальний елемент? [1, с.202-229]
25. Назвіть пакет в якому розташований клас Scanner: [1, с.278-284]
26. Що використовується для створення об'єкта? [1, с.205-208]
27. Сигнатура методу визначається:... [1, с.209-212]
28. Аргументи, які передаються в метод при виклику називаються ... [1, с.218-220]
29. Об'єктами якого класу представлені в Java рядки- константи незмінної довжини і змісту? [1, с.251-278]
30. Для читання рядків з текстового файлу у якому порядку і які слід використовувати класи? [1, с.293-300]
31. Назвіть оператор, який дозволяє виконати переривання поточної ітерації циклу: [1, с.124-127]
32. Який модифікатор доданий до членів класу (полів чи методів) робить їх закритими для доступу з інших класів: [1, с.213-218]
33. За допомогою якого із перелічених методів можна побудувати лінію? [6, с.128-152]
34. Який заголовок буде мати метод, який заповнює одномірний дійсний масив випадковими числами? [1, с.202-229]
35. Який пакет необхідно імпортувати для організації файлового введення / виводу? [1, с.293-300]
36. Що використовується для опису властивостей об'єкта? [1, с.202-208]
37. Між формальними і фактичними параметрами методу повинна бути відповідність... [1, с.218-220]
38. Об'єктами якого класу представлені в Java рядки змінної довжини і змісту? [1, с.251-278]
39. Який модифікатор доданий до членів класу (полів чи методів) робить їх відкритими для доступу з інших класів: [1, с.213-218]
40. З якого службового слова починається оголошення класу: [1, с.202-205]
41. За допомогою якого із перелічених методів можна побудувати зафарбований прямокутник? [6, с.128-152]
42. Який заголовок буде мати метод, який виводить двовимірний дійсний масив на екран? [1, с.202-229]

5. ЛІТЕРАТУРА ДЛЯ ВИВЧЕННЯ ДИСЦИПЛІНИ

Основна

1. Кузніченко С.Д., Коваленко Л.Б. Основи алгоритмізації та програмування. Навчальний посібник – Одеса, ОДЕКУ, 2019 – 338 с.
2. Верлань А.Ф., Чмырь І.А., Кузніченко С.Д., Коваленко Л.Б. Императивное программирование и объектно-ориентированное

- моделирование: Java, UML, OCL // Учебное пособие для студентов высших учебных заведений. Одесса «Экология», 2013 г.– 432 с.
3. Горлова Т.М. Теорія алгоритмів. [Електронний ресурс]: конспект лекцій для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форм навчання / Т.М. Горлова, К.Є. Бобрівник, Н.В. Ліманська – К.: НУХТ, 2015. – 95 с.
 4. Седжвик Р., Уэйн К. Алгоритмы на Java, 4-е изд.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2013. – 848 с.: ил.
 5. Мелешко Є.В., Якименко М.С., Поліщук Л.І. М 47 Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.

Додаткова

6. А.Г.Горбань. Програмування в Java. К.:ДУТ, 2015. – 310 с. Режим доступу: http://www.dut.edu.ua/uploads/1_1364_67332457.pdf
7. Ахо, А. Структуры данных и алгоритмы / Ахо А., Хопкрофт Дж., Ульман Дж. – М.: Издательский дом «Вильямс», 2001. –384 с.
8. Вирт Н. Алгоритмы и структуры данных / Н. Вирт – 2-ое изд., испр. – СПб.: Невский диалект, 2001. – 352 с.
9. Ковалюк Т.В. Основы програмування.– К.: Видавнича група ВНУ, 2005. – 384 с.: іл.
10. Глушаков С.В. Программирование на Java 2: Изд.2-е.- Харьков: Фолио, 2003. – 536 с. – (Учебный курс).
11. Герберт Шилдт. Java. Полное руководство (Java SE 7, 8-е издание). – М.: Изд. Дом «Вильямс», 2012. – 1104с.: ил.
12. Ноутон О., Шилдт Г., Java 2: Пер. с англ. – СПб.: БХВ - Петербург, 2007. – 1072с.: ил.
13. Эккель Б. Философия Java.: ВНУ., СПб – 2001. 850с.
14. Морган М. Java 2. Руководство разработчика. – М.: Изд. дом «Вильямс», 2000.
15. Флэнэген Д. Java in a Nutshell. Полное руководство. – Киев: ВНУ, 1998.
16. Роганов Е.А. Основы информатики и программирования: Учебное пособие – М.: МГИУ, 2001. – 315 с.
17. Java 2. Библиотека профессионала, том 1. Основы. 7-е издание //Кей С. Хорстманн, Гари Корнелл. – Издательский дом «Вильямс», 2006, 896 с.: ил.
18. Быстрая разработка программ на Java и C++: принципы, примеры, практика Роберт С. Мартин - Издательский дом «Вильямс», 2003, 752 с.: ил.
19. Шень А. Программирование: теоремы и задачи. – М.: МЦНМО, 1995.
20. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.: МЦНМО, 2000.

Методичне забезпечення

21. Методичні вказівки до виконання лабораторних робіт з дисципліни “Основи програмування та алгоритмічні мови”, частина II, для студентів I курсу денної форми навчання. Напрямок підготовки - комп’ютерні науки / Кузніченко С.Д., Коваленко Л.Б.– Одеса, ОДЕКУ, 2009, 50 с.
22. Методичні вказівки до самостійної роботи студентів та виконання контрольних робіт з дисципліни “Основи програмування та алгоритмічні мови ”, для студентів 2 курсу заочної форми навчання. Напрямок підготовки - комп’ютерні науки / Кузніченко С.Д. , Коваленко Л.Б.– Одеса, ОДЕКУ, 2011 – 50 с.
23. Методичні вказівки до самостійної роботи студентів та виконання контрольних робіт з дисципліни “Основи програмування та алгоритмічні мови”, для студентів I курсу заочної форми навчання. Напрямок підготовки - комп’ютерні науки / Кузніченко С.Д. , Коваленко Л.Б.– Одеса, ОДЕКУ, 2011 – 52 с.
24. Методичні вказівки до виконання лабораторних робіт студентів з дисципліни “Алгоритмізація та програмування ”, частина 1, для студентів I курсу денної форми навчання. Напрямок підготовки – комп’ютерні науки / Укладачі: Кузніченко С.Д., Коваленко Л.Б. – Одеса, ОДЕКУ, 2012. - 51 с.
25. Методичні вказівки до виконання лабораторних робіт студентів з дисципліни “Алгоритмізація та програмування ” для студентів I курсу заочної форми навчання. Напрямок підготовки – комп’ютерні науки / Укладачі: Кузніченко С.Д., Коваленко Л.Б. – Одеса, ОДЕКУ, 2013. – 50 с. (ел.вар.)
26. Електронна бібліотека ОДЕКУ www.library-odeku.16mb.com