



КАФЕДРА ЦИФРОВОЇ ЕКОНОМІКИ ТА
БІЗНЕС-АНАЛІТИКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ

ЗАТВЕРДЖЕНО
на засіданні кафедри економічної кібернетики
протокол № 6 від "21" січня 2020 р.

Зав. кафедри _____ Шевчук І.Б.
(підпис)

ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ
СТУДЕНТА І МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
ЩОДО ЇХ ВИКОНАННЯ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Об'єктно-орієнтоване програмування

(назва навчальної дисципліни)

галузь знань: _____ 05 «Соціальні та поведінкові науки»
(шифр та найменування галузі знань)

спеціальність: _____ 051 "Економіка"
(код та найменування спеціальності)

спеціалізація: _____ Інформаційні технології в бізнесі
(найменування спеціалізації)

освітній ступінь: _____ бакалавр
(бакалавр/магістр)

Укладач:

_____ Шевчук І.Б., доцент, д.е.н., доцент
(ПІБ, посада, науковий ступінь, вчене звання)

ЛЬВІВ 2020

1. ЗАГАЛЬНІ МЕТОДИЧНІ РЕКОМЕНДАЦІЇ З ВИКОНАННЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТА

Самостійна робота студентів (СРС) займає провідне місце у системі сучасної вищої освіти. З усіх видів навчальної діяльності СРС значною мірою забезпечує формування самостійності як провідної риси особистості студента.

Самостійна робота завершує завдання усіх інших видів навчальної діяльності. Адже знання, що не стали об'єктом власної діяльності, не можуть вважатися дійсним надбанням людини. Тому СРС має навчальне, особисте та суспільне значення.

СРС – це багатоаспектне та поліфункціональне явище з двоєдиністю цілей:

- формування самостійності студента;
- розвиток здібностей, вмінь, знань та навичок студентів.

Завдяки СРС відбувається перехід від переважно виконавчої репродуктивної діяльності студентів до пошукового, творчого начала на всіх етапах навчання у ВНЗ.

Самостійна робота з дисципліни “Об’єктно-орієнтоване програмування” припускає її здійснення в наступних видах: самостійне вивчення теоретичного матеріалу, написання програм мовою Java в інтегрованому середовищі IntelliJ IDEA.

Метою виконання самостійної роботи є більше глибоке вивчення мови програмування Java та набуття практичних навичок написання нею прикладних програм.

Правильна організація самостійної роботи необхідна для оволодіння дисципліною “Об’єктно-орієнтоване програмування”, оскільки обсяг аудиторних занять не дозволяє розглянути на лекціях і лабораторних усі основні аспекти програмування мовою Java. Крім того, успішність самостійної роботи багато в чому визначає успішність здачі заліку й наступної практичної діяльності, тому що тільки в рамках виконання самостійної роботи студент одержує навички практичної діяльності.

У самостійній роботі реалізуються наступні компетенції студента:

1. Соціально-особистісні:

- 1.1. Уміння коректно й переконливо представити свою позицію, сприймати критику, досягати компромісу;
- 1.2. Готовність до постійного саморозвитку, вміння будувати стратегію особистого й професійного навчання й розвитку;
- 1.3. Адаптивність і комунікабельність;
- 1.4. Наполегливість у досягненні мети;
- 1.5. Креативність, здатність до системного мислення.

2. Загальнонаукові:

- 2.1. Розуміння й використання основних категорій алгоритмізації та програмування;
- 2.2. Застосування методів наукового пізнання.

3. Інструментальні:

- 3.1. Здатність до самоорганізації, організації й планування;
- 3.2. Навички роботи з комп'ютером, уміння використовувати сучасні інформаційні технології (довідкові системи, Інтернет і ін.) для одержання доступу до джерел інформації, зберігання й обробки даних.
4. Загальнопрофесійні:
 - 4.1. Володіння основними навичками опису етапів розробки програм;
 - 4.2. Уміння спроектувати, написати, перевірити та дослідити результати виконання простих програм;
 - 4.3. Використання різних способів і стилів програмування.
5. Спеціальні професійні:
 - 5.1. Здатність розробляти алгоритми та структури даних для програмних продуктів;
 - 5.2. Володіння основами методів та технологій програмування в інтегрованому середовищі IntelliJ IDEA;
 - 5.3. Здатність розв'язувати математичні та економічні задачі шляхом створення відповідних застосувань;
 - 5.6. Використання Інтернет-ресурсів для розв'язання експериментальних і практичних завдань з професійної діяльності.

Самостійна робота виконується студентами під керівництвом викладача, який здійснює аудиторну роботу в навчальній групі.

Самостійна робота студентів повинна мати такі головні ознаки:

- бути виконаною особисто студентом;
- бути закінченою розробкою, де розкриваються й аналізуються актуальні проблеми з певної теми або її окремих аспектів;
- демонструвати достатню компетентність автора в розкритті питань, що досліджуються;
- мати навчальну, наукову, й/або практичну спрямованість і значимість;
- містити певні елементи новизни;
- самостійна письмова робота оформляється відповідно до вимог кафедри.

При виконанні самостійної роботи необхідно дотримуватись НАСТУПНИХ ПРАВИЛ:

1. Перед виконанням самостійної роботи потрібно повністю ознайомитися зі змістом завдання, підібрати потрібну літературу, визначити усі параметри виконання індивідуального завдання на комп'ютері.
2. Результатом виконання самостійної роботи є програма, написана в інтегрованому середовищі IntelliJ IDEA, та звіт, який виконується з використанням комп'ютерної техніки та надрукований на папері формату А4. Оформлення звіту: шрифт – Times New Roman; розмір шрифту – 14 кегель; інтервал між рядками – півтора; абзац – 12,5 мм, поля: верхнє і нижнє – 20 мм, ліве – 25 мм, праве – 15 мм; нумерація сторінок – по центру нижнього поля. Зразок оформлення титульної сторінки наведено у додатку А.
3. Після перевірки кожного завдання викладачем студент зобов'язаний усунути допущені помилки, інакше він не допускається до виконання наступного завдання.

Усі види самостійної роботи повинні бути здані у встановлений графіком термін. Викладач фіксує факт здачі кожної роботи та виставляє оцінку в журнал.

Поради із планування й організації часу, необхідного для виконання самостійної роботи

Раціональне планування і організація самостійної роботи студентів є найважливішою умовою її ефективності.

Планування самостійної роботи направлено на формування логічно вибудованої, прозорої, зрозумілої, доступної і ефективної системи організації самостійної роботи та її оцінки.

При цьому необхідно пам'ятати, що самостійна робота студентів виконує в навчальному процесі кілька функцій:

- розвиваючу (підвищення культури розумової праці, привчання до творчих видів діяльності, вдосконалення інтелектуальних здібностей студентів);
- інформаційно-навчальну (навчальна діяльність на аудиторних заняттях, не підкріплена самостійною роботою, стає мало результативною);
- орієнтуючу і стимулюючу (процесу навчання надається прискорення і мотивація);
- виховну (формується і розвиваються професійні якості фахівця);
- дослідницьку (новий рівень професійно-творчого мислення).

В основі самостійної роботи студентів лежать наступні принципи: розвиток творчої діяльності, цільове планування, особистісно-діяльнісний підхід.

Самостійну роботу можна назвати ефективною тільки в тому випадку, якщо вона організована і реалізується в освітньому процесі як цілісна система на всіх етапах навчання.

Можна виділити кілька об'єктивних закономірностей організації самостійної роботи студентів:

- творча складова самостійної роботи зростає в міру навчання;
- в процесі організації самостійної роботи виникає потреба в методичному забезпеченні;
- застосування інформаційних технологій стає частиною організації і моніторингу самостійної роботи студентів на всіх її етапах.

У процесі самостійної роботи студент набуває навиків самоорганізації, самоконтролю, самоврядування, саморефлексії і стає активним самостійним суб'єктом навчальної діяльності.

Самостійна робота повинна давати важливий вплив на формування особистості майбутнього фахівця. Кожен, хто навчається самостійно планує режим своєї роботи з урахуванням часу роботи бібліотеки, профільних лабораторій, комп'ютерних класів і т.п. Він виконує самостійну роботу за особистим індивідуальним планом, в залежності від його підготовки, часу та інших умов.

Першим завданням в організації позааудиторної самостійної роботи є складання розкладу, що відображає час занять і їх характер, перерви на обід, вечір, відпочинок, сон, проїзд і т.п. Із самого початку студенту не потрібно прагнути робити відразу найважчу її частину. Доцільно вибрати щось середнє за складністю.

Після цього, перейти до більш важкої роботи, легке залишивши наостанок. Розумову працю необхідно не тільки правильно організувати, а й стимулювати. Важливо вміти підтримувати стійку увагу до досліджуваного матеріалу. Вироблення уваги вимагає значних вольових зусиль від студента. Стійка увага з'являється тоді, коли людина ставиться до справи з інтересом.

Слід правильно організувати свої заняття за часом: 50 хвилин – робота, 5-10 хвилин – перерва, після 3 годин роботи перерва – 20-25 хвилин. Інакше наростаюча втома спричинить нестійкість уваги. Організація активного відпочинку передбачає чергування розумової та фізичної діяльності, що відновлює працездатність людини.

Опис послідовності дій студента при виконанні самостійної роботи

Організацію самостійної роботи можна умовно розділити на три етапи:

- планування навчальної діяльності та її методична підготовка;
- здійснення цієї діяльності та її супровід;
- контроль, аналіз результатів (з можливими змінами в плануванні самостійної роботи).

Рекомендації щодо використання матеріалів навчально-методичного комплексу навчальної дисципліни

Зміст вивчення дисципліни “Об’єктно-орієнтоване програмування” визначено її робочою програмою.

Інформативну частину навчання складають навчальні посібники, конспекти лекцій у паперовій та електронній формі, план, зміст та методичні рекомендації до проведення лабораторних занять, методичні рекомендації до виконання самостійної та індивідуальної науково-дослідної роботи, перелік рекомендованої до вивчення літератури, ресурси мережі Інтернет.

У рекомендаціях до проведення лабораторних занять з дисципліни “Об’єктно-орієнтоване програмування” міститься план занять, завдання для виконання лабораторних робіт та перелік питань для самостійного опрацювання матеріалу. Також зазначається короткий теоретичний коментар до кожної теми, що допомагає студентові ознайомитися із сутністю питань, на основі яких базується виконання завдань лабораторних робіт. Окрім цього у даних методичних рекомендаціях можна ознайомитися з питаннями, що виносяться на обговорення та списком літератури, необхідної для цілеспрямованої роботи студента при підготовці до наступного лабораторного заняття.

Рекомендації щодо роботи з літературою

Найважливішим інформаційним джерелом вивчення навчальної дисципліни “Об’єктно-орієнтоване програмування” є ресурси мережі Інтернет. Основна частина матеріалу в Інтернеті розрахована на професіоналів, тому при вивченні навчальної дисципліни спочатку необхідно користуватися літературою навчального характеру.

При опрацюванні матеріалу потрібно дотримуватись таких правил:

1. Зосередитися на тому, що читаєш.
2. Виділити головну думку автора.
3. Виділити основні питання тексту від другорядних.
4. Зрозуміти думку автора чітко і ясно, що допоможе виробити власну думку.
5. Уявити ясно те, що читаєш.

У процесі роботи над темою тлумачення незнайомих слів і спеціальних термінів слід знаходити у фаховій літературі, термінологічних словниках. Незрозумілі місця, фрази, вирази доречно перечитувати декілька разів, щоб зрозуміти їх зміст.

Після прочитання тексту необхідно:

1. Усвідомити зв'язок між теоретичними положеннями і практикою.
2. Закріпити прочитане у свідомості.
3. Пов'язати нові знання з попередніми у даній галузі.
4. Перейти до заключного етапу засвоєння і опрацювання – записам.

Записи необхідно починати з назви теми та посібника, прізвища автора, року видання та назви видавництва. Якщо це журнал, то рік і номер видання, заголовок статті. Після чого скласти план, тобто короткий перелік основних питань тексту в логічній послідовності теми.

Складання плану, або тез логічно закінченого за змістом уривка тексту, сприяє кращому його розумінню. План може бути простий або розгорнутий, тобто більш поглиблений, особливо при опрацюванні додаткової літератури за даною темою. Записи необхідно вести розбірливо і чітко. Вони можуть бути короткі або розгорнуті залежно від рівня знань студента, багатства його літературної і професійної лексики, навичок самостійної роботи з книгою.

Для зручності користування записами необхідно залишати поля для заміток і вільні рядки для доповнень. Записи не повинні бути одноманітними. В них необхідно виділяти важливі місця, головні слова, які акцентуються різним шрифтом або різним кольором шрифтів, підкреслюванням, замітками на полях, рамками, стовпчиками тощо. Записи можуть бути у вигляді конспекту, простих або розгорнутих тез, цитат, виписок, систематизованих таблиць, графіків, діаграм, схем.

Поради із підготовки до поточного, проміжного та підсумкового контролю

Контрольні заходи включають поточний і підсумковий контроль знань студентів. Поточний контроль є органічною частиною навчального процесу і проводиться під час лекцій та лабораторних занять.

Форми поточного контролю:

- усна співбесіда за матеріалами розглянутої теми на початку лабораторного заняття з оцінкою відповідей студентів (5-10 хв.);
- письмове фронтальне опитування студентів на початку чи в кінці лабораторного заняття (5-10 хв.). Відповіді перевіряються і оцінюються викладачем у поза аудиторний час;
- перевірка виконання завдань лабораторних робіт;
- тестова перевірка знань студентів;
- модульний контроль;
- інші форми.

При кредитно-модульній системі навчання теми самостійної роботи входять у модуль, який контролюються після закінчення логічно завершеної частини лекцій та інших видів занять з дисципліни “Об’єктно-орієнтоване програмування” та їх результати враховуються при виставленні підсумкової оцінки.

Приклад модульного контрольного завдання:

I рівень завдання з вибором відповіді – тестові завдання.	
Завдання з вибором відповіді вважається виконаним правильно, якщо студентом вказана вірна відповідь. За кожне правильно виконане завдання виставляється 0,2 бали	
1.	Властивістю алгоритмів є: а. Визначеність б. Скінченність в. Правильність г. Всі відповіді правильні
2.	Перекладає команди програми в машинні коди по черзі й відразу їх виконує: а. Транслятор б. Компілятор в. Інтерпретатор г. Немає правильної відповіді
3.	Над елементами масиву можна виконувати такі дії: а. додавати б. віднімати в. множити г. всі відповіді правильні
4.	Які імена правильно написані мовою C? а. 1Name б. name_ в. +name г. {name}
5.	Який із керуючих символів в команді cout << здійснює перехід до наступної позиції табуляції? а. \n б. \t в. \b г. \\\
6.	Змінна, яка визначена в середині блоку {} є: а. локальною б. глобальною в. видимою г. формальною
7.	Відзначте правильне визначення константи: а. const int DOZEN = 12; б. const double; в. double COFF = (const)1.2e10; г. немає правильної відповіді
8.	Що таке cout? а. об’єкт типу iostream б. клас, що виводить дані на термінал в. змінна, котру програміст повинен створити для виведення даних
9.	У якому із записів використано префіксний інкремент? а. int d; ++d; б. int d; d++;

	в. int d; - -d; г. int d; d--;
10.	Результат операції «&&» є «істина» (1), якщо: а. обидва її операнди правдиві (не рівні 0); б. хоча б один з її операндів є «істина»; в. його операнд є «істина» (1); г. обидва її операнди неправдиві (0).
Другий рівень – завдання з короткою відповіддю. Завдання з короткою відповіддю вважається виконаним правильно, якщо студент дав вірні визначення, посилання, тлумачення, короткі коментарі. За кожне правильно виконане завдання виставляється 0,5 бали	
1.	Що таке інкремент і декремент?
2.	Оголошення і використання змінних і констант
Третій рівень – завдання із вибором відповіді – тестові завдання практичної направленості Завдання з вибором відповіді на практичне питання вважається виконаним правильно, якщо в картці тестування записана правильна відповідь. За правильно виконане завдання виставляється 0,5 бали.	
1.	Якщо у функції main() виконується: <pre> ... int a=9; // у глобальному просторі void f() { int a; a = 4; } ... cout << a;</pre> то що буде виведено на екран? а. 9 б. 0 в. 4 г. інший
2.	Що буде на екрані після виконання наступного фрагмента коду? <pre> int a=2; int ar[10]={a-=a, a-=a , a-=a , a-=a }; for (int i=0; i<10; i++){ cout<<ar[i]<<' '; }</pre> а. 4 1 1 1 0 0 0 0 0 0 б. 4 18 16 32 0 0 0 0 0 0 в. 4 16 256 65536 0 0 0 0 0 0 г. 0 0 0 0 0 0 0 0 0 0
3.	Якщо i = 5, то який буде результат? <pre> do { cout << (++i)++ << " "; } while (i>=5 && i < 8);</pre> а. 6 б. 6 8 в. 6 7 г. 6 7 8
4.	Що буде на екрані після виконання наступного фрагмента коду? <pre> double a[10] = {6, 7.8, 8.0, -6, -5.7, 5, 5.7}; cout << a[6 % 8/4 + 21 % 9];</pre> а. -5.7 б. -5 в. -6 г. 5
РАЗОМ:	
5 балів	

Список рекомендованої літератури

1. Java : Учебник для начинающих программистов : [Электрон. ресурс]. - Режим доступа: <http://proglang.su/java>
2. Блинов И.Н., Романчик В. С. Java. Методы программирования : уч.-мет. пособие / И. Н. Блинов, В. С. Романчик. – Минск : издательство "Четыре четверти", 2013. - 896 с.
3. Васильев А. Н. Java. Объектно-ориентированное программирование: Учебное пособие. – СПб.: Питер, 2011. – 400 с.
4. Гамма, Э., Хелм, Р., Джонсон, Р., Влиссидес, Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб. : Питер, 2007. – 366 с.
5. Давыдов В.Г. Программирование и основы алгоритмизации: Учеб. пособие. / В.Г. Давыдов. – М.: Высш. шк., 2003. – 447 с.
6. Інженерія якості програмного забезпечення: навч. посібник / Г.В Табунщик, Р.К. Кудерметов, Т.І. Брагіна. - Запоріжжя: ЗНТУ, 2013. - 180 с.
7. Кингсли-Хьюджес Э., Кингсли-Хьюджес К. Справочник программиста. - М.: ООО "ИД Вильямс", 2007.
8. Популярные технологии программирования в 2017 году : [Электрон. ресурс]. - Режим доступа: <https://vc.ru/dev/21483-what-language-2017>
9. Программирование на Java: [Электрон. ресурс]. - Режим доступа: <https://www.intuit.ru/studies/courses/16/16/info>
10. Рудаков А. В. Технология разработки программных продуктов : учебник для студ. сред. проф. образования / А. В. Рудаков. – 7-е изд., стер. – М. : Издательский центр "Академия", 2012. - 208 с.
11. Топ-5 полезных видеокурсов по Java : [Электрон. ресурс]. - Режим доступа: <https://javarush.ru/groups/posts/528-top-5-poleznychkh-videokursov-po-java-chastjh-1-->
12. Уроки по основам языка программирования JAVA для начинающих : [Электрон. ресурс]. - Режим доступа: <https://www.fandroid.info/tutorial-po-osnovam-yazyka-programirovaniya-java-dlya-nachinayushhih/>

2. ГРАФІК ВИКОНАННЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТА

№ розділу, теми	Назва розділу, теми	Кількість годин СРС	Форма контролю	Термін виконання СРС (сем./тиж.)
Тема 1	Загальні уявлення про мову Java	4	Програма, написана в інтегрованому середовищі IntelliJ IDEA, звіт.	3/12-15
Тема 2	Основи програмування на Java	4		
Тема 3	Робота з рядками	4		
Тема 4	Концепція об'єктно-орієнтованого програмування	9		
Тема 5	Обробка виключень	9		
Разом годин самостійної роботи студента		30		

3. ЗМІСТ САМОСТІЙНОЇ РОБОТИ СТУДЕНТА І МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ЩОДО ЇЇ ВИКОНАННЯ

Структура класів, що описують реальні об'єкти, і приклади їх використання

У програмуванні на мовах високого рівня використовується структурований призначений для користувача іменований тип даних, який представляє собою запис, який складається із сукупності елементів різних типів та має загальне ім'я і призначення. Використовувати даний тип можна для зберігання різнорідної інформації про конкретні об'єкти. Структура типу задається розробником.

Наприклад, для роботи з даними про студентів, де представлено прізвище, ім'я, по батькові, рік народження, група в якості призначеного для користувача типу даних можна створити запис **Student**, в якому будуть оголошені відповідні поля *fam, ima, otch, god, grupper*.

Запис на мовах C / C ++ використовує ключове слово `struct`:

```
struct Student { /* Student - ім'я запису */
    char fam [15]; /* 1-е поле запису - прізвище студента */
    char ima [10]; /* 2-е поле запису - ім'я студента */
    char otch [15]; /* 3-є поле запису - по батькові студента */
    int god; /* 4-е поле запису - рік народження */
    int grupper; /* 5-е поле запису - група */
};
```

У мові Java поняття запису відсутнє, і ці структури реалізуються за допомогою класів користувачів з відповідними полями.

Приклад 1.1. Ввести інформацію про співробітників фірми (фамілія, ім'я, по батькові, посада, оклад). Вивести на екран інформацію в табличному вигляді.

Нижче наведено код реалізації.

```
package firma;
import java.util.Scanner;

class Sotrudnik{
    String fam, im, otch, doljnost;
    int oklad;
}

public class Firma { // головний клас
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in,"cp1251");

        System.out.println("Введіть кількість співробітників => ");
        int kol=sc.nextInt();
```

```
sc.nextLine(); // очищення буфера після введення числа
```

```
Sotrudnik []sotr=new Sotrudnik[kol]; // отримана посилання на масив
// співробітників
```

```
System.out.println("Введіть інформацію про кожного співробітника");
for (int i = 0; i < sotr.length; i++) {
```

```
sotr[i]=new Sotrudnik();// отримано посилання на i-тій елемент
```

```
// Присвоєння значень полям
```

```
System.out.print("Введіть прізвище"+(i+1)+" співробітника => ");
sotr[i].fam= sc.nextLine();
```

```
System.out.print("Введіть його ім'я => ");
sotr[i].im=sc.nextLine();
```

```
System.out.print("Введіть його по батькові => ");
sotr[i].otch=sc.nextLine();
```

```
System.out.print("Введіть його посаду => ");
sotr[i].doljnost=sc.nextLine();
```

```
System.out.print("Введіть його оклад => ");
sotr[i].oklad=sc.nextInt();
```

```
sc.nextLine(); //очищення буфера
} // end for
```

```
// Виведення інформації про співробітників
```

```
System.out.println( "\n Співробітники фірми: \n фам \t ім'я \t по
бат +\t посада \tОклад");
```

```
for (int i = 0; i < sotr.length; i++) {
```

```
System.out.println(s.fam+ "\t"+s.im + "\t"+s.otch
+ "\t "+s.doljnost + "\t\t"+s.oklad);
```

```
}
```

```
}
```

```
}
```

Можливий альтернативний варіант запису
цикла **for** для роботи з елементами масиву

```
for (Sotrudnik s : sotr) {
    System.out.print(s.fam+ "\t"+s.im + "\t"+s.otch
        + "\t "+s.doljnost + "\t\t"+s.oklad+ "\n");
}
```

У класах, що представляють собою структуру даних реальних об'єктів, полями можу бути змінні, самі які є структурами даних. Розглянемо трохи ускладнений варіант попереднього завдання, де використовуються дві структури: Співробітник і Дитина.

Приклад 1.2. Ввести інформацію про співробітників фірми (фамілія, ім'я, по батькові, посада) і їх дітей (ім'я, вік). Вивести на екран інформацію в табличному вигляді.

Укрупнений алгоритм.

1. Ввести кількість співробітників.
2. Для кожного співробітника
 - а) ввести прізвище, ім'я, по батькові, посаду та кількість дітей;
 - б) якщо кількість дітей не дорівнює нулю, то для кожної дитини ввести ім'я і вік.
3. Для кожного співробітника вивести інформацію про нього і його дітей.
4. Закінчити.

Програма, що реалізує даний алгоритм.

```
package firma2;
```

```
import java.util.Scanner;
```

```
class Sotrudnik{
    String fam,im,otch, doljnost ;
    int kolDet;
    Rebenok []reb=null;
}
```

```
class Rebenok{ String imaR; int vozrastR;
```

```
полям класу Sotrudnik є масив об'єктів класу Rebenok
}
```

```
public class RecSotr1 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in,"cp1251");
        System.out.println("Введіть кількість співробітників => ");
        int kol=sc.nextInt();
        sc.nextLine(); // очистка буфера
        Sotrudnik []sotr=new Sotrudnik[kol];
        System.out.println("Введіть інформацію про кожного співробітника: ");
        for (int i = 0; i < sotr.length; i++) {
```

```

sotr[i]=new Sotrudnik();
System.out.print("Введіть фамілію "+(i+1)+" співробітника => ");
sotr[i].fam= sc.nextLine();

System.out.print("Введіть ім'я => ");
sotr[i].im=sc.nextLine();

System.out.print("Введіть по батькові => ");
sotr[i].otch=sc.nextLine();

System.out.print("Введіть посаду => ");
sotr[i].doljnost=sc.nextLine();

System.out.print("Введіть кількість дітей => ");
sotr[i].kolDet=sc.nextInt();
sc.nextLine(); // очистка буфера

if (sotr[i].kolDet!=0) {
    sotr[i].reb=new Rebenok[sotr[i].kolDet];
    System.out.println("Діти => ");
    for (int j = 0; j < sotr[i].reb.length; j++) {
        sotr[i].reb[j]=new Rebenok();

        System.out.print("Введіть ім'я "+(i+1)+" дитини => ");
        sotr[i].reb[j].imaR = sc.nextLine();

    } } }
System.out.print("Введіть її вік => ");
sotr[i].reb[j].vozzrastR = sc.nextInt();
sc.nextLine(); // очистка буфера

// Виведення інформації про співробітників
System.out.println("\nСпівробітники фірми \n фам \t ім'я \t по бат \t посада");
for (Sotrudnik s : sotr) {
    System.out.print(s.fam+ "\t"+s.im + "\t"+ s.otch  + "\t "+ s.doljnost);
    System.out.println("\n    Діти: ");
    for (Rebenok r : s.reb)
        System.out.println("\t\t"+ r.imaR+ "\t\t"+ r.vozrastR);
    System.out.println("");
}
} }

```

Подальші приклади наведені для демонстрації типових алгоритмів роботи з масивами об'єктів. Вони схожі з роботою зі звичайними числовими або рядковими

масивами. Використовується сортування за вказаними полями, пошук максимуму (мінімуму), середнього значення та ін.

Приклад 1.3. Ввести масив із заданою користувачем кількістю країн світу, що містить інформацію виду: Назва, площа. Вивести його в табличному вигляді з відповідними заголовками колонок.

Далі для введених країн:

- визначити країну з максимальною площею;
- провести сортування країн по зростанню площі;
- провести сортування країн за назвою в алфавітному порядку;
- підрахувати середню площу введених країн і визначити країни з площею більше середньої;
- знайти країну за назвою, введеною користувачем.

Вся отримана інформація повинна виводитися на екран з відповідними поясненнями.

Загальний алгоритм:

1. Ввести кількість країн.
2. Ввести характеристики країн.
3. Вивести характеристики країн в табличному вигляді.
4. Знайти країну з максимальною площею.
5. Вивести інформацію про країну з максимальною площею.
6. Впорядкувати країни за площею.
7. Вивести упорядкований по площі список.
8. Впорядкувати країни за назвою.
9. Вивести упорядкований за назвою список.
10. Ввести назву потрібної країни.
11. Визначити, чи є шукана країна в списку, і вивести відповідну інформацію.
12. Закінчити.

Програма, що реалізує даний алгоритм.

```
package recordcountry;
```

```
import java.util.Scanner;
```

```
class Strana{
```

```
String name; // назва країни
```

```
double square; // площа країни
```

```
}
```

```
public class RecordCountry {
```

```
public static void main(String[] args) {
```

```
Scanner sc=new Scanner(System.in,"cp1251");
```

```

// Виведення інформації про країни
System.out.println("Введіть кількість країн ");
int n=sc.nextInt();          // кількість країн
Strana country[]=new Strana[n];
System.out.println("Введіть інформацію про країни: ");
for (int i = 0; i < country.length; i++) {
    sc.nextLine();          // очистка буфера
    country[i]=new Strana();
    System.out.print("Назва "+(i+1)+"-ої країни");
    country[i].name=sc.nextLine();
    System.out.print("Площа "+(i+1)+"-ої країни =>");
    country[i].square=sc.nextDouble();
}

// Виведення отриманої інформації
System.out.println("\nХарактеристики країн:");
for (int i = 0; i < country.length; i++) {
    System.out.println(""+country[i].name+"\t"+country[i].square
        +" МЛН КВ. КМ");
}

// Країна з максимальною площею
int nommax=0;                // номер елемента для країни з максимальною
                            // площею (початкове значення)
double    max=country[nommax].square;    // максимальна площа
                                                // (початкове значення)

for (int i = 0; i < country.length; i++)
    if (country[i].square>max) {
        max= country[i].square;
        nommax=i;
    }
System.out.println("\nКраїна з максимальною площею:");
System.out.println(""+country[nommax].name+"\t"+country[nommax].square
    +"МЛН КВ. КМ");

// Сортування країн за площею
for (int i = 0; i < country.length-1; i++)
    for (int j = 0; j < country.length-1-i; j++)
        if (country[j].square>country[j+1].square){
            Strana rab=country[j]    // rab – робоча змінна для перестановки
            country[j]=country[j+1];
            country[j+1]=rab;
        }
}

```

```

System.out.println("\nВідсортований список по площі:");//
for (int i = 0; i < country.length; i++) {
    System.out.println(""+country[i].name+"\t"+country[i].square+"млн кв. км");
}

// Сортування країн за назвою
for (int i = 0; i < country.length-1; i++)
    for (int j = 0; j < country.length-i-1; j++)
        if(country [j].name.compareTo(country[i+1].name)>0){
            Страна rab=country[j]; //rab – робоча змінна для перестановки
            country[j]=country[j+1];
            country[j+1]=rab;
        }
System.out.println("\nВідсортований список за назвами:");
for (int i = 0; i < country.length; i++) {
    System.out.println(""+country[i].name +"\t"+country[i].square
        +" млн кв. км");
}

// Площа більша за середню
double s=0; // сумарна площа
for (int i = 0; i < country.length; i++)
    s+=country[i].square;
double sr=s/country.length; ; // середня площа
System.out.println("Середня площа =" +sr);
System.out.println("\nКраїни з площею більшою за середню");
for (int i = 0; i < country.length; i++) {
    if (country[i].square>sr)
        System.out.println(country[i].name +"\t"+country[i].square +" млн кв. км");
}

// Пошук по назві
sc.nextLine(); // очистка буфера
System.out.println("Введіть назву країни для пошуку");
String name=sc.nextLine();
int nom=-1; // -1 – країна з такою назвою відсутня

for (int i = 0; i < country.length; i++)
    if (name.equalsIgnoreCase(country[i].name))
        nom=i;

if (nom!= -1) {
    System.out.println("Така країна є в списку. Це"
        +country[nom].name+" "+country[nom].square+" млн кв. км");
}

```



```

    }else System.out.println("Такої країни нема в списку");
}
}

```

Приклад 1.4. Для завдання, розглянутого в попередньому прикладі, розробити методи для виконання кожної підзадачі:

- 1) введення масиву країн - метод setCountry (...);
- 2) виведення інформації про країни в табличному вигляді - метод show- Array (...);
- 3) виведення інформації про одну країну - метод showCountry (...);
- 4) визначення індексу країни з максимальною площею - метод pomMax (...);
- 5) сортування країн по зростанню площі - sortSquare (...);
- 6) сортування країн за назвою в алфавітному порядку – метод sortName (...);
- 7) визначення середньої площі - метод avgSquare (...);
- 8) визначення країни з площею більшою за середню - метод bigger (...);
- 9) пошук країни за назвою - метод findForName (...).

Код програми з використанням зазначених методів.

```

package recordCountry2;

```

```

import java.util.Scanner;

```

```

class Strana{

```

```

    String name;

```

```

    double square;

```

```

}

```

```

public class RecordCountry2{

```

```

// Методи для роботи з об'єктами класу Strana

```

```

    public static Strana [] setCountryArr(int k){ // Ведення масиву країн

```

```

        Scanner sc=new Scanner(System.in,"cp1251");

```

```

        Strana country[]=new Strana[k];

```

```

        System.out.println("Введіть інформацію про країни: ");

```

```

        for (int i = 0; i < country.length; i++) {

```

```

            country[i]=new Strana();

```

```

            System.out.print("Назває "+(i+1)+"-ої країни");

```

```

            country[i].name=sc.nextLine();

```

```

            System.out.print("Площа "+(i+1)+"-ої країни");

```

```

            country[i].square=sc.nextDouble();

```

```

            sc.nextLine(); // очистка буфера

```

```

    }
return country;
}

```

```

public static void showArray(Strana [] cntr){

```

```

// Виведення масиву

```

```

    for (int i = 0; i < cntr.length; i++) {

```

```

        System.out.println("" + cntr[i].name + " \t" + cntr[i].square + " млн кв. км");

```

```

    } }

```

```

// Інформація про одну країну

```

```

public static void showCountry(Strana cntr){

```

```

System.out.println(""+cntr.name+"\t"+cntr.square+" млн кв. км"); }

```

```

// Номер країни з максимальною площею

```

```

public static int NomMax(Strana []st){

```

```

    int nommax=0; // номер елемента країни з макс. площею

```

```

// (початкове значення)

```

```

    double max=st[nommax].square; // макс. площа (початкове значення)

```

```

    for (int i = 0; i < st.length; i++)

```

```

        if (st[i].square>max) {

```

```

            max= st[i].square;

```

```

            nommax=i;

```

```

        }

```

```

    return nommax;

```

```

}

```

```

// Сортування країн за площею

```

```

public static void sortSquare(Strana []cntr){

```

```

for (int i = 0; i < cntr.length-1; i++)

```

```

    for (int j = 0; j < cntr.length-1-i; j++)

```

```

        if (cntr[j].square>cntr[j+1].square){

```

```

            Strana rab=cntr[j]; // rab – робоча змінна для перестановки

```

```

            cntr[j]=cntr[j+1];

```

```

            cntr[j+1]=rab;

```

```

        }

```

```

}

```

```

// Сортування країн за назвою

```

```

public static void sortName(Strana [] cntr){

```

```

for (int i = 0; i < cntr.length-1; i++)
  for (int j = 0; j < cntr.length-i-1; j++)
    if(cntr[j].name.compareTo(cntr[i+1].name)>0){
      Strana rab=cntr[j];    // rab – робоча змінна для перестановки
      cntr[j]=cntr[j+1];
      cntr[j+1]=rab;
    }
}

```

```

public static double avgSquare( Strana [] cntr){ // Середня площа
  double s=0; // сумарна площа
  for (int i = 0; i < cntr.length; i++)
    s+=cntr[i].square;
  double sr=s/cntr.length; // середня площа
  return sr;
}

```

// Країни з площею більшою за середню

```

public static Strana [] Bigger(Strana cntr[]){
  double sred=avgSquare(cntr);
  int kol=0; // кількість країн
  for (int i = 0; i < cntr.length; i++) {
    if (cntr[i].square>sred)
      ++kol;
  }

  if (kol != 0){
    Strana [] bigCountry=new Strana[kol];
    int n=-1;
    for (int i = 0; i < cntr.length; i++)
      if (cntr[i].square>sred) {
        bigCountry[++n]=cntr[i];
      }
    return bigCountry;
  } else return null;
}

```

// Пошук за назвою

```

public static Strana findForName(Strana cntr[], String name){

```

```

int n=-1;           // -1 – країна із заданою назвою відсутня
for (int i = 0; i < cntr.length; i++)
    if (name.equals(cntr[i].name))
        n=i;
if (n!= -1) {
    return cntr[n];
}else return null;
}

public static void main(String[] args) { // Головний клас з методами
    Scanner sc=new Scanner(System.in,"cp1251");
    System.out.print("Введіть кількість країн");
    int n=sc.nextInt();                // кількість країн
    Strana country[]= setCountryArr(n); // введіть інформацію про країни

    System.out.println("\nХарактеристики країн:");
    showArray(country);                // виведення отриманої інформації

// країна з максимальною площею
int nommax=NomMax(country);
System.out.println("\nКраїна з максимальною площею:");
    showCountry(country[nommax]);

// сортування країн за площею
    sortSquare(country);
    System.out.println("\nВідсортований список за площею");
    showArray(country);

// сортування країн за назвою
    sortName(country);
    System.out.println("\nВідсортований список за назвою");
    showArray(country);

// середня площа
    System.out.println("Середня площа =" +avgSquare(country));

// площа більша за середню
    System.out.println("\nКраїни з площею більшою за середню");
    Strana [] larger=Bigger(country);

```

```
showArray(larger);
```

```
// пошук країни
```

```
System.out.println("\nПошук країни \n Введіть назву країни ");
sc.nextLine();
String sname=sc.nextLine();
Strana sfind =findForName(country, sname);
if (sfind!=null) {
    showCountry(sfind);
} else {
    System.out.println("Такої країни немає в списку!");
}}}
```

Завдання 1.1. Використовуючи розглянуті приклади 1.3 і 1.4, виконати завдання згідно заданому варіанту у вигляді двох проєктів:

1-й проєкт – без використання методів;

2-й проєкт – зі створенням методів для кожної підзадачі.

Варіант 1

Розробити клас для зберігання інформації про автомобілі:

Модель, реєстр_номер, рік випуску, пробіг, вартість

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найдешевший автомобіль;
- упорядкувати масив по спаданню року випуску;
- організувати пошук за реєстраційним номером, виправлення одного з полів і виведення повної інформації про автомобіль після редагування.

Варіант 2

Розробити клас для зберігання інформації про фільми:

Назва_фільму, рік випуску, країна, жанр, вартість прокату

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити фільми, вартість прокату яких перевищує середню;
- вивести інформацію про фільм самого раннього року випуску;
- упорядкувати масив за назвами фільмів в алфавітному порядку;
- організувати пошук за назвою, виправлення одного з полів і виведення повної інформації про фільм після редагування.

Варіант 3

Розробити клас для зберігання інформації про людей:

Прізвище, Ім'я, По батькові, вік, розмір одягу

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити людину з найбільшим розміром одягу;

- визначити середній розмір одягу для людей старших 40 років;
- упорядкувати масив по зростанню розміру одягу;
- організувати пошук за прізвищем, виправлення одного з полів і виведення повної інформації про людину після редагування.

Варіант 4

Розробити клас для зберігання інформації про товари:

Найменування товару, виробник, кількість одиниць, ціна

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити товари з максимальною кількістю одиниць;
- визначити середню ціну товарів і кількість товарів з ціною нижчою за середню;
- упорядкувати список за зменшенням цін товарів;
- організувати пошук по найменуванню товару, виправлення одного з полів і виведення повної інформації про товар після редагування.

Варіант 5

Розробити клас для зберігання інформації про футбольні команди:

Назва, місто, місце в лізі, кількість перемог

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити команду з найбільшою кількістю перемог;
- визначити команди з кількістю перемог більшою за середню;
- упорядкувати масив по спаданню місць в лізі;
- організувати пошук за назвою команди, виправлення одного з полів і виведення повної інформації про команду після редагування.

Варіант 6

Розробити клас для зберігання інформації про гори:

Назва, місце розташування, висота

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найвищу вершину;
- визначити вершини з висотою понад 1000 м;
- упорядкувати масив по зростанню висот;
- організувати пошук за назвою вершини, виправлення одного з полів і виведення повної інформації про вершині після редагування.

Варіант 7

Розробити клас для зберігання інформації про людей:

Прізвище, Ім'я, стать, зріст

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити середнє зростання жінок і середній зріст чоловіків;
- визначити найвищого чоловіка;
- упорядкувати список за зростанням росту;
- організувати пошук за прізвищем, виправлення одного з полів і виведення повної інформації про людину після редагування.

Варіант 8

Розробити клас для зберігання інформації про людей:

Прізвище, Ім'я, рік народження, місяць народження

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найстарішу людину;
- визначити середній вік (використовувати тільки рік народження) і людей старших за середній вік;
- упорядкувати масив за прізвищами в порядку, зворотному алфавітному;
- організувати пошук за прізвищем, виправлення одного з полів і виведення повної інформації про людину після редагування.

Варіант 9

Розробити клас для зберігання інформації про співаків:

Виконавець, рейтинг, кількість випущених альбомів

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найпопулярнішого виконавця;
- визначити середню кількість випущених альбомів і виконавців з кількістю випущених альбомів більших середнього;
- упорядкувати масив за прізвищами виконавців в порядку, зворотному алфавітному;
- організувати пошук за прізвищем, виправлення одного з полів і виведення повної інформації про пісню після редагування.

Варіант 10

Розробити клас для зберігання інформації про озера:

Назва, місце розташування, площа

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найбільше озеро;
- підрахувати кількість озер з площею меншою за середню;
- упорядкувати список за назвами;
- організувати пошук за назвою озера, виправлення одного з полів і виведення повної інформації про озеро після редагування.

Варіант 11

Розробити клас для зберігання інформації про театри:

Назва, художній керівник, адреса, рейтинг

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити театр з найбільшим рейтингом;
- упорядкувати масив за назвами в зворотному порядку;
- організувати пошук за прізвищем художнього керівника, виправлення одного з полів і виведення повної інформації про театр після редагування;
- вивести інформацію про театри з більш високим рейтингом, ніж рейтинг театру з назвою, заданою користувачем.

Варіант 12

Розробити клас для зберігання інформації про річки:

Назва, місце розташування, довжина

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найкоротшу річку;
- вивести інформацію про річки з довжиною більшою за середню;
- упорядкувати список річок за назвами в алфавітному порядку;
- організувати пошук за назвою річки, виправлення одного з полів і виведення повної інформації про річку після редагування.

Варіант 13

Розробити клас для зберігання інформації про мотоцикли:

Модель, реєстр_номер, рік_випуску, пробіг, вартість

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найдорожчий мотоцикл;
- упорядкувати масив по спаданню пробігу;
- організувати пошук за реєстраційним номером, виправлення одного з полів і виведення повної інформації про мотоцикл після редагування.

Варіант 14

Розробити клас для зберігання інформації про мультфільми:

Назва_мультфільму, рік випуску, країна, жанр, вартість прокату

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити мультфільми, вартість прокату яких не перевищує середню;
- вивести інформацію про мультфільми самого раннього року випуску;
- упорядкувати масив за назвами мультфільмів в алфавітному порядку;
- організувати пошук за назвою, виправлення одного з полів і виведення повної інформації про мультфільм після редагування.

Варіант 15

Розробити клас для зберігання інформації про тварин:

Кличка, Порода, Вид, Вік, Розмір

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити тварину найбільшого розміру;
- визначити середній розмір для тварин молодших 5 років;
- упорядкувати масив по зростанню віку;
- організувати пошук за кличкою, виправлення одного з полів і виведення повної інформації про тварину після редагування.

Варіант 16

Розробити клас для зберігання інформації про послуги:

Назва послуги, компанія, кількість задіяних осіб, вартість

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити послуги для надання яких задіяно максимальну кількість осіб;
- визначити середню ціну послуг і кількість послуг з вартістю вищою за середню;

- упорядкувати список за зменшенням вартості послуг;
- організувати пошук за назвою послуг, виправлення одного з полів і виведення повної інформації про послугу після редагування.

Варіант 17

Розробити клас для зберігання інформації про баскетбольні команди:

Назва, місто, місце в турнірній таблиці, кількість перемог

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити команду з найбільшою кількістю перемог;
- визначити команди з кількістю перемог більшою за середню;
- упорядкувати масив по спаданню місць у турнірній таблиці;
- організувати пошук за назвою команди, виправлення одного з полів і виведення повної інформації про команду після редагування.

Варіант 18

Розробити клас для зберігання інформації про рівнини:

Назва, місце розташування, висота над рівнем моря

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найвищу рівнину над рівнем моря;
- визначити рівнину з висотою понад 100 м над рівнем моря;
- упорядкувати масив по зростанню висот;
- організувати пошук за назвою рівнини, виправлення одного з полів і виведення повної інформації про рівнини після редагування.

Варіант 19

Розробити клас для зберігання інформації про коней:

Кличка, порода, стать, висота, вартість, власник

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити середній ріст коней чоловічої та жіночої статей;
- визначити найвищого коня;
- упорядкувати список за зростанням вартості коней;
- організувати пошук за кличкою, виправлення одного з полів і виведення повної інформації про коня після редагування.

Варіант 20

Розробити клас для зберігання інформації про людину:

Прізвище, Ім'я, рік народження, місяць народження

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити наймолодшу людину;
- визначити середній вік (використовувати тільки рік народження) і людей молодших за середній вік;
- упорядкувати масив за прізвищами в алфавітному порядку;
- організувати пошук за прізвищем, виправлення одного з полів і виведення повної інформації про людину після редагування.

Варіант 21

Розробити клас для зберігання інформації про акторів театру:

Актор, гонорар, кількість вистав

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найпопулярнішого актора театру;
- визначити середню кількість зіграних вистав і акторів з кількістю вистав меншою за середню;
- упорядкувати масив за прізвищами авторів в порядку, зворотному алфавітному;
- організувати пошук за прізвищем, виправлення одного з полів і виведення повної інформації про актора після редагування.

Варіант 22

Розробити клас для зберігання інформації про моря:

Назва, місце розташування, площа

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найменше море;
- підрахувати кількість морів з площею меншою за середню;
- упорядкувати список за назвами;
- організувати пошук за назвою моря, виправлення одного з полів і виведення повної інформації про море після редагування.

Варіант 23

Розробити клас для зберігання інформації про цирки:

Назва, художній керівник, адреса, рейтинг

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити цирк з найбільшим рейтингом;
- упорядкувати масив за назвами в зворотному порядку;
- організувати пошук за прізвищем художнього керівника, виправлення одного з полів і виведення повної інформації про цирк після редагування;
- вивести інформацію про цирки з більш високим рейтингом, ніж рейтинг цирку з назвою, заданою користувачем.

Варіант 24

Розробити клас для зберігання інформації про течії:

Назва, місце розташування, тип, потужність

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найпотужнішу течію;
- вивести інформацію про течію з потужністю більшою за середню;
- упорядкувати список течій за назвами в алфавітному порядку;
- організувати пошук за назвою течії, виправлення одного з полів і виведення повної інформації про течію після редагування.

Варіант 25

Розробити клас для зберігання інформації про велосипеди:

бренд, клас, діаметр колеса, матеріал рами, вартість

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити найдорожчий велосипед;
- упорядкувати масив по спаданню діаметра коліс;
- організувати пошук за брендом, виправлення одного з полів і виведення повної інформації про велосипед після редагування.

Варіант 26

Розробити клас для зберігання інформації про кліпи:

Назва_кліпу, автор, рік випуску, країна, співак, вартість прокату

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити кліпи, вартість прокату яких перевищує середню;
- вивести інформацію про кліп самого раннього року випуску;
- упорядкувати масив за співаками в алфавітному порядку;
- організувати пошук за назвою, виправлення одного з полів і виведення повної інформації про кліп після редагування.

Варіант 27

Розробити клас для зберігання інформації про людей:

Прізвище, Ім'я, По батькові, вік, розмір взуття, вага, зріст

і реалізувати алгоритм роботи з масивом даних об'єктів, в якому потрібно:

- визначити людину з найбільшим розміром ноги;
- визначити середню вагу для людей старших 35 років;
- упорядкувати масив по зростанню зросту;
- організувати пошук за прізвищем, виправлення одного з полів і виведення повної інформації про людину після редагування.

Список рекомендованої літератури

1. Java : Учебник для начинающих программистов : [Електрон. ресурс]. - Режим доступу: <http://proglang.su/java>
2. Блинов И.Н., Романчик В. С. Java. Методы программирования : уч.-мет. пособие / И. Н. Блинов, В. С. Романчик. – Минск : издательство "Четыре четверти", 2013. - 896 с.
3. Васильев А. Н. Java. Объектно-ориентированное программирование: Учебное пособие. – СПб.: Питер, 2011. – 400 с.
4. Популярные технологии программирования в 2017 году : [Електрон. ресурс]. - Режим доступу: <https://vc.ru/dev/21483-what-language-2017>
5. Программирование на Java: [Електрон. ресурс]. - Режим доступу: <https://www.intuit.ru/studies/courses/16/16/info>
6. Топ-5 полезных видеокурсов по Java : [Електрон. ресурс]. - Режим доступу: <https://javarush.ru/groups/posts/528-top-5-poleznychkh-videokursov-po-java-chastjh-1-->

7. Уроки по основам языка программирования JAVA для начинающих : [Електрон. ресурс]. - Режим доступу: <https://www.fandroid.info/tutorial-po-osnovam-yazyuka-programmirovaniya-java-dlya-nachinayushhih/>

Питання для самоконтролю

1. Які форми запису має умовний оператор if?
2. Які оператори використовуються для організації циклів в java?
3. Які з циклів є циклами з передумовою, а які з післяумовою?
4. Які три операції виконуються в циклі for?
5. Які керуючі оператори використовуються в циклах?
6. Що таке вкладені цикли?
7. Чи обов'язково повинні бути присутні всі гілки оператора if ?
8. Чи допускається вкладеність операторів if ?
9. Яку роль виконує оператор switch ?
10. Який тип виразу має “селектор” оператора switch ?
11. Скільки міток може містити кожна з гілок оператора switch ?
12. Як виконується робота оператора switch ?
13. Коли ефективно використовувати оператор switch ?
14. Який пріоритет мають логічні операції порівняно з операціями відношення?
15. Що називається циклом ?
16. Для чого потрібний лічильник циклу ?
17. Загальний формат оператора for ?
18. Як виконується робота оператора while ?
19. Яку роль виконує оператор do while ?
20. Яка відмінність між while та do while ?
21. Як виконується робота оператора break ?
22. Як виконується робота оператора continue ?
23. Як називаються такі логічні операції як &&, ||, ! ?
24. Загальний формат оператора while ?
25. Загальний формат оператора do while.
26. Яким чином використовуються логічні операції такі як &&, ||, ! в операторах for, do while, while.
27. Що таке масив?
28. Що таке одновимірний масив?
29. Які є правила оформлення масивів?
30. Для чого використовується масив типу char?
31. Якими способами можна ініціалізувати елементи масивів?
32. Коли і для чого використовується ключове слово static?
33. Як звернутися до елемента масиву ?
34. Як ініціалізувати масив при оголошенні?
35. Що таке багатомірний масив?
36. Для чого використовуються багатомірні масиви?
37. Скільки індексів може мати багатомірний масив?
38. Як ініціалізується багатовимірний масив?

39. Як ініціалізувати двовимірний масив при оголошенні?
40. Що таке клас? Для чого використовують класи?
41. Що таке поле?
42. Що таке об'єкт?
43. Що таке конструктор?
44. Що таке оператор new?
45. Що таке модифікатор static?
46. Що таке модифікатор protected?
47. Що таке модифікатор void?
48. Що таке модифікатор private?
49. Що таке модифікатор abstract?
50. Що таке модифікатор final?
51. Що таке метод?
52. Що таке заголовок методу?
53. Що таке тіло методу?
54. Що таке аргументи (або параметри)?
55. Що таке тип значень, які повертаються?
56. Що таке абстрактний метод?
57. Що таке статистичний метод?
58. Що таке статичне поле?
59. Яка різниця між статичним та нестатичним методами?
60. Як здійснюється доступ до членів класу?
61. Що таке ієрархія?
62. Яка суть принципу ООП - успадкування?
63. Що таке абстрактний клас?
64. Що таке абстрактний метод?
65. Як заборонити успадкування?
66. Який клас називається батьківським?
67. Який клас називається нащадком?
68. Які є види успадкування?
69. Як відбувається багаторівневе успадкування?
70. Як відбувається множинне успадкування?
71. Як відбувається ієрархічне успадкування?
72. Як відбувається одиночне успадкування?
73. Що таке динамічна диспетчеризація методів?

Порядок оформлення та захисту самостійної роботи

1. Звіт про виконання самостійної роботи подається у друкованому форматі на папері формату А4.

Оформлення звіту: шрифт – Times New Roman; розмір шрифту – 14 кегель; інтервал між рядками – півтора; абзац – 12,5 мм, поля: верхнє і нижнє – 20 мм, ліве – 25 мм, праве – 15 мм; нумерація сторінок – по центру нижнього поля.

Звіт повинен мати наступну структуру:

- титульна сторінка (додаток А).
- формулювання завдання.
- лістинг програми.
- вікно результатів виконання програми.
- література.

2. Самостійна робота подається викладачу не пізніше ніж за 1 тиждень до заліку в електронному (програма, написана в інтегрованому середовищі IntelliJ IDEA) та друкованому вигляді (звіт).

3. Оцінка за самостійну роботу виставляється на заключному занятті з навчальної дисципліни на основі попереднього ознайомлення викладача зі змістом самостійної роботи. Можливий захист завдання шляхом демонстрації роботи програми у середовищі IntelliJ IDEA.

4. Оцінка за самостійну роботу є обов'язковою складовою підсумкової оцінки з навчальної дисципліни.

4. КРИТЕРІЇ ОЦІНЮВАННЯ

Результати самостійної роботи оцінюються викладачем згідно з чинною шкалою оцінювання.

№ з/п	Види робіт. Критерії оцінювання знань студентів	Бали рейтингу	Максимальна кількість балів
Самостійна робота студентів (СРС)			
Критерії оцінювання		5 балів	
завдання зроблене повністю та здане вчасно, якісно оформлено звіт		5	
завдання зроблене, але є незначні помилки в процесі побудови коду програми або неналежно оформлений звіт;		4	
завдання зроблене, але є незначні помилки в процесі побудови коду програми та немає звіту;		3	
завдання зроблене, але є суттєві помилки в процесі побудови коду програми та немає звіту;		2	
завдання не виконане або тільки розпочато процес побудови коду програми.		0-1	

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Кафедра економічної кібернетики

Самостійна робота № 1

з дисципліни

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

тема «_____»

Виконав(ла):

Перевірила: