**“Obyekt-yönlü proqramlaşdırma (Java)” fənni üzrə**

**TƏDRİS-TEMATİK PLAN**

| **№** | **Tarix** | **Mövzular** | **Tədris materialının məzmunu və sərbəst işlər** | | | **Saat** | **Qeyd** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **I BÖLMƏ. Obyekt-yönlü proqramlaşdırmanın konsepsiyaları (Object-Oriented Programming Concepts).** | | | | | | | |
|  |  | **Introduction to Object Oriented Programming (OOP).** | | | **Giriş. Obyekt-yönlü proqramlaşdırma (OYP) fənninin predmeti, məqsəd və vəzifələri.** |  |  |
|  |  | **Machine Languages, Assembly Languages and High-Level Languages.** | | |  |  |  |
|  |  | **History of Java.** | | |  |  |  |
|  |  | **Introduction to Java Programming.** | | |  |  |  |
|  |  | **Introduction to Object Technology** | | |  |  |  |
|  |  | * **The Automobile as an Object** * **Methods and Classes** * **Instantiation** * **Reuse** * **Messages and Method Calls** * **Attributes and Instance Variables** * **Encapsulation and Information Hiding** * **Inheritance** * **Interfaces** * **Object-Oriented Analysis and Design (OOAD)** * **The UML (Unified Modeling Language)** | | |  |  |  |
|  |  | **Basic Principles of Object Oriented Programming.** | | | **Obyekt-yönlü proqramlaşdırmanın əsas prinsipləri, mahiyyəti.** |  |  |
|  |  | **Issues with Procedure Oriented Programming.** | | | **Prosedur-yönlü proqramlaşdırmanın problemləri.** |  |  |
|  |  | **Procedure Oriented versus Object Oriented Programming.** | | | **Prosedur-yönlü və obyekt-yönlü proqramlaşdırmanın müqayisəsi.** |  |  |
|  |  | **Programming Languages** | | |  |  |  |
|  |  | **Java** | | |  |  |  |
|  |  | **A Typical Java Development Environment** | | |  |  |  |
|  |  | **Test-Driving a Java Application** | | |  |  |  |
|  |  | **Internet and World Wide Web**   * **The Internet: A Network of Networks** * **The World Wide Web: Making the Internet User-Friendly** * **Web Services and Mashups** * **Ajax** * **The Internet of Things** | | |  |  |  |
|  |  | **Software Technologies** | | |  |  |  |
|  |  | **Keeping Up-to-Date with Information Technologies** | | |  |  |  |
|  |  | **Introduction to Java Applications: Input/Output and Operators** | | |  |  |  |
|  |  | **Your First Program in Java: Printing a Line of Text** | | |  |  |  |
|  |  | **Modifying Your First Java Program** | | |  |  |  |
|  |  | **Displaying Text with printf** | | |  |  |  |
|  |  | **Another Application: Adding Integers**   * **import Declarations** * **Declaring Class Addition** * **Declaring and Creating a Scanner to Obtain User Input from the Keyboard** * **Declaring Variables to Store Integers** * **Prompting the User for Input** * **Obtaining an int as Input from the User** * **Prompting for and Inputting a Second int** * **Using Variables in a Calculation** * **Displaying the Result of the Calculation** * **Java API Documentation** | | |  |  |  |
|  |  | **Memory Concepts** | | |  |  |  |
|  |  | **Arithmetic** | | |  |  |  |
|  |  | **Decision Making: Equality and Relational Operators** | | |  |  |  |
|  |  | **(Optional) GUI and Graphics Case Study: Using Dialog Boxes** | | |  |  |  |
|  |  | **Basic of Object Oriented Programming.** | | |  |  |  |
|  |  | **Concept of Object Oriented Programming:**   * **Object** * **Class** * **Abstraction** * **Encapsulation** * **Inheritance** * **Polymorphism** | | | **Obyekt-yönlü proqramlaşdırma konsepsiyaları:**   * **Obyekt** * **Sinif** * **Abstraksiya** * **İnkapsulyasiya** * **Varislik** * **Polumorfizm** |  |  |
|  |  | **Example of Some Object Oriented Languages.** | | | **Obyekt-yönlü proqramlaşdırma dilləri nümunələri.** |  |  |
|  |  | **Advantages and Disadvantages of OOP.** | | | **OYP-nin üstünlükləri və çatışmazlıqlar.** |  |  |
|  |  | **Questions and Exercises:**   * **What Is an Object?** * **What Is a Class?** * **What Is Inheritance?** * **What Is an Interface?** * **What Is a Package?** | | | **Suallar və tapşırıqlar:**   * **Obyekt nədir?** * **Sinif nədir?** * **Varislik nədir?** * **İnterfeys nədir?** * **Paket nədir?** |  |  |
| **II BÖLMƏ. İdarəetmə operatorları (Control Statements).** | | | | | | | |
|  |  | **Introduction: Part 1. Assignment, ++ and -- Operators** | |  | |  |  |
|  |  | **Algorithms** | |  | |  |  |
|  |  | **Pseudocode** | |  | |  |  |
|  |  | **Control Structures** | |  | |  |  |
|  |  | **if Single-Selection Statement** | |  | |  |  |
|  |  | **if…else Double-Selection Statement** | |  | |  |  |
|  |  | **while Repetition Statement** | |  | |  |  |
|  |  | **Formulating Algorithms: Counter-Controlled Repetition** | |  | |  |  |
|  |  | **Formulating Algorithms: Sentinel-Controlled Repetition** | |  | |  |  |
|  |  | **Formulating Algorithms: Nested Control Statements** | |  | |  |  |
|  |  | **Compound Assignment Operators** | |  | |  |  |
|  |  | **Increment and Decrement Operators** | |  | |  |  |
|  |  | **Primitive Types** | |  | |  |  |
|  |  | **(Optional) GUI and Graphics Case Study: Creating Simple Drawings** | |  | |  |  |
|  |  | **Introduction: Part 2. Logical Operators** | |  | |  |  |
|  |  | **Essentials of Counter-Controlled Repetition** | |  | |  |  |
|  |  | **for Repetition Statement** | |  | |  |  |
|  |  | **Examples Using the for Statement** | |  | |  |  |
|  |  | **do…while Repetition Statement** | |  | |  |  |
|  |  | **switch Multiple-Selection Statement** | |  | |  |  |
|  |  | **break and continue Statements** | |  | |  |  |
|  |  | **Logical Operators** | |  | |  |  |
|  |  | **Structured Programming Summary** | |  | |  |  |
|  |  | **(Optional) GUI and Graphics Case Study: Drawing Rectangles and Ovals** | |  | |  |  |
| **III BÖLMƏ. Metodlar (Methods).** | | | | | | | |
|  |  | **Program Modules in Java** | |  | |  |  |
|  |  | **static Methods, static Variables and Class Math** | |  | |  |  |
|  |  | **Declaring Methods** | |  | |  |  |
|  |  | **Notes on Declaring and Using Methods** | |  | |  |  |
|  |  | **Method-Call Stack and Stack Frames** | |  | |  |  |
|  |  | **Argument Promotion and Casting** | |  | |  |  |
|  |  | **Java API Packages** | |  | |  |  |
|  |  | **Case Study: Secure Random-Number Generation** | |  | |  |  |
|  |  | **Case Study: A Game of Chance; Introducing enum Types** | |  | |  |  |
|  |  | **Scope of Declarations** | |  | |  |  |
|  |  | **Method Overloading** | |  | |  |  |
|  |  | **(Optional) GUI and Graphics Case Study: Colors and Filled Shapes** | |  | |  |  |
| **IV BÖLMƏ. Massivlər və Massivlərin Siyahısı (Arrays and ArrayLists).** | | | | | | | |
|  |  | **Primitive Types vs Reference Types** | |  | |  |  |
|  |  | **Arrays** | |  | |  |  |
|  |  | **Declaring and Creating Arrays** | |  | |  |  |
|  |  | **Examples Using Arrays**   * **Creating and Initializing an Array** * **Using an Array Initializer** * **Calculating the Values to Store in an Array** * **Summing the Elements of an Array** * **Using Bar Charts to Display Array Data Graphically** * **Using the Elements of an Array as Counters** * **Using Arrays to Analyze Survey Results** | |  | |  |  |
|  |  | **Exception Handling: Processing the Incorrect Response**   * **The try Statement** * **Executing the catch Block** * **toString Method of the Exception Parameter** | |  | |  |  |
|  |  | **Enhanced for Statement** | |  | |  |  |
|  |  | **Passing Arrays to Methods** | |  | |  |  |
|  |  | **Pass-By-Value vs Pass-By-Reference** | |  | |  |  |
|  |  | **Multidimensional Arrays** | |  | |  |  |
|  |  | **Variable-Length Argument Lists** | |  | |  |  |
|  |  | **Using Command-Line Arguments** | |  | |  |  |
|  |  | **Class Arrays** | |  | |  |  |
|  |  | **Introduction to Collections and Class ArrayList** | |  | |  |  |
|  |  | **(Optional) GUI and Graphics Case Study: Drawing Arcs** | |  | |  |  |
| **V BÖLMƏ. Siniflərə və Obyektlərə Giriş (Introduction to Classes and Objects).** | | | | | | | |
|  |  | **Instance Variables, set Methods and get Methods**   * **Account Class with an Instance Variable, a set Method and a get Method** * **AccountTest Class That Creates and Uses an Object of Class Account** * **Compiling and Executing an App with Multiple Classes** * **Account UML Class Diagram with an Instance Variable and set and get Methods** * **Additional Notes on This Example** * **Software Engineering with private Instance Variables and public set and get Methods** | |  | |  |  |
|  |  | **Default and Explicit Initialization for Instance Variables** | |  | |  |  |
|  |  | **Account Class: Initializing Objects with Constructors** | |  | |  |  |
|  |  | **Declaring an Account Constructor for Custom Object Initialization** | |  | |  |  |
|  |  | **Class AccountTest: Initializing Account Objects When They’re Created** | |  | |  |  |
|  |  | **Account Class with a Balance: Floating-Point Numbers**  **Account Class with a balance Instance Variable of Type double**  **AccountTest Class to Use Class Account** | |  | |  |  |
|  |  | **Case Study: Card Shuffling and Dealing Simulation** | |  | |  |  |
|  |  | **Case Study: Class GradeBook Using an Array to Store Grades** | |  | |  |  |
|  |  | **Case Study: Class GradeBook Using a Two-Dimensional Array** | |  | |  |  |
| **VI BÖLMƏ. Siniflər və Obyektlər: Dərin yanaşma (Classes and Objects: A Deeper Look).** | | | | | | | |
|  |  | **Time Class Case Study** | |  | |  |  |
|  |  | **Controlling Access to Members** | |  | |  |  |
|  |  | **Referring to the Current Object’s Members with the this Reference** | |  | |  |  |
|  |  | **Time Class Case Study: Overloaded Constructors** | |  | |  |  |
|  |  | **Default and No-Argument Constructors** | |  | |  |  |
|  |  | **Notes on Set and Get Methods** | |  | |  |  |
|  |  | **Composition** | |  | |  |  |
|  |  | **enum Types** | |  | |  |  |
|  |  | **Garbage Collection** | |  | |  |  |
|  |  | **static Class Members** | |  | |  |  |
|  |  | **static Import** | |  | |  |  |
|  |  | **final Instance Variables** | |  | |  |  |
|  |  | **Package Access** | |  | |  |  |
|  |  | **Using BigDecimal for Precise Monetary Calculations** | |  | |  |  |
|  |  | **(Optional) GUI and Graphics Case Study: Using Objects with Graphics** | |  | |  |  |
| **VII BÖLMƏ. Obyekt-yönlü proqramlaşdırma: Varislik (Object-Oriented Programming: Inheritance).** | | | | | | | |
|  |  | **Superclasses and Subclasses** | |  | |  |  |
|  |  | **protected Members** | |  | |  |  |
|  |  | **Relationship Between Superclasses and Subclasses**   * **Creating and Using a CommissionEmployee Class** * **Creating and Using a BasePlusCommissionEmployee Class** * **Creating a CommissionEmployee–BasePlusCommissionEmployee Inheritance Hierarchy** * **CommissionEmployee–BasePlusCommissionEmployee Inheritance Hierarchy Using protected Instance Variables** * **CommissionEmployee–BasePlusCommissionEmployee Inheritance Hierarchy Using private Instance Variables** | |  | |  |  |
|  |  | **Constructors in Subclasses** | |  | |  |  |
|  |  | **Class Object** | |  | |  |  |
|  |  | **(Optional) GUI and Graphics Case Study: Displaying Text and Images Using Labels** | |  | |  |  |
| **VIII BÖLMƏ. Polimorfizm və İnterfeyslər (Object-Oriented Programming: Polymorphism and Interfaces).** | | | | | | | |
|  |  | **Polymorphism Examples** | |  | |  |  |
|  |  | **Demonstrating Polymorphic Behavior** | |  | |  |  |
|  |  | **Abstract Classes and Methods** | |  | |  |  |
|  |  | **Case Study: Payroll System Using Polymorphism**   * **Abstract Superclass Employee** * **Concrete Subclass SalariedEmployee** * **Concrete Subclass HourlyEmployee** * **Concrete Subclass CommissionEmployee** * **Indirect Concrete Subclass BasePlusCommissionEmployee** * **Polymorphic Processing, Operator instanceof and Downcasting** | |  | |  |  |
|  |  | **Allowed Assignments Between Superclass and Subclass Variables** | |  | |  |  |
|  |  | **final Methods and Classes** | |  | |  |  |
|  |  | **A Deeper Explanation of Issues with Calling Methods from Constructors** | |  | |  |  |
|  |  | **Creating and Using Interfaces**   * **Developing a Payable Hierarchy** * **Interface Payable** * **Class Invoice** * **Modifying Class Employee to Implement Interface Payable** * **Modifying Class SalariedEmployee for Use in the Payable Hierarchy** * **Using Interface Payable to Process Invoices and Employees Polymorphically** * **Some Common Interfaces of the Java API** | |  | |  |  |
|  |  | **Java SE 8 Interface Enhancements**   * **default Interface Methods** * **static Interface Methods** * **Functional Interfaces** | |  | |  |  |
|  |  | **GUI and Graphics Case Study: Drawing with Polymorphism** | |  | |  |  |
| **IX BÖLMƏ. İstisnaların işlənilməsi: Dərin yanaşma (Exception Handling: A Deeper Look).** | | | | | | | |
|  |  | **Example: Divide by Zero without Exception Handling** | |  | |  |  |
|  |  | **Example: Handling ArithmeticExceptions and InputMismatchExceptions** | |  | |  |  |
|  |  | **When to Use Exception Handling** | |  | |  |  |
|  |  | **Java Exception Hierarchy** | |  | |  |  |
|  |  | **finally Block** | |  | |  |  |
|  |  | **Stack Unwinding and Obtaining Information from an Exception Object** | |  | |  |  |
|  |  | **Chained Exceptions** | |  | |  |  |
|  |  | **Declaring New Exception Types** | |  | |  |  |
|  |  | **Preconditions and Postconditions** | |  | |  |  |
|  |  | **Assertions** | |  | |  |  |
|  |  | **try-with-Resources: Automatic Resource Deallocation** | |  | |  |  |
| **X BÖLMƏ. İstifadəçinin Qrafiki İnterfeysi Komponentləri (GUI Components: Part 1).** | | | | | | | |
|  |  | **Java’s Nimbus Look-and-Feel** | |  | |  |  |
|  |  | **Simple GUI-Based Input/Output with JOptionPane** | |  | |  |  |
|  |  | **Overview of Swing Components** | |  | |  |  |
|  |  | **Displaying Text and Images in a Window** | |  | |  |  |
|  |  | **Text Fields and an Introduction to Event Handling with Nested Classes** | |  | |  |  |
|  |  | **Common GUI Event Types and Listener Interfaces** | |  | |  |  |
|  |  | **How Event Handling Works** | |  | |  |  |
|  |  | **JButton** | |  | |  |  |
|  |  | **Buttons That Maintain State**   * **JCheckBox** * **JRadioButton** | |  | |  |  |
|  |  | **JcomboBox: Using an Anonymous Inner Class for Event Handling** | |  | |  |  |
|  |  | **JList** | |  | |  |  |
|  |  | **Multiple-Selection Lists** | |  | |  |  |
|  |  | **Mouse Event Handling** | |  | |  |  |
|  |  | **Adapter Classes** | |  | |  |  |
|  |  | **JPanel Subclass for Drawing with the Mouse** | |  | |  |  |
|  |  | **Key Event Handling** | |  | |  |  |
|  |  | **Introduction to Layout Managers** | |  | |  |  |
|  |  | **FlowLayout** | |  | |  |  |
|  |  | **BorderLayout** | |  | |  |  |
|  |  | **GridLayout** | |  | |  |  |
|  |  | **Using Panels to Manage More Complex Layouts** | |  | |  |  |
|  |  | **JTextArea** | |  | |  |  |
|  |  | **Graphics and Java D** | |  | |  |  |
|  |  | **Introduction** | |  | |  |  |
|  |  | **Graphics Contexts and Graphics Objects** | |  | |  |  |
|  |  | **Color Control** | |  | |  |  |
|  |  | **Manipulating Fonts** | |  | |  |  |
|  |  | **Drawing Lines, Rectangles and Ovals** | |  | |  |  |
|  |  | **Drawing Arcs** | |  | |  |  |
|  |  | **Drawing Polygons and Polylines** | |  | |  |  |
|  |  | **Java D API** | |  | |  |  |
| **XI BÖLMƏ. Sətirlər, Simvollar və requlyar ifadələr (Strings, Characters and Regular Expressions).** | | | | | | | |
|  |  | **Fundamentals of Characters and Strings** | |  | |  |  |
|  |  | **Class String**   * **String Constructors** * **String Methods length, charAt and getChars** * **Comparing Strings** * **Locating Characters and Substrings in Strings** * **Extracting Substrings from Strings** * **Concatenating Strings** * **Miscellaneous String Methods** * **String Method valueOf** | |  | |  |  |
|  |  | **Class StringBuilder**   * **StringBuilder Constructors** * **StringBuilder Methods length, capacity, setLength and ensureCapacity** * **StringBuilder Methods charAt, setCharAt, getChars and reverse** * **StringBuilder append Methods** * **StringBuilder Insertion and Deletion Methods** | |  | |  |  |
|  |  | **Class Character** | |  | |  |  |
|  |  | **Tokenizing Strings** | |  | |  |  |
|  |  | **Regular Expressions, Class Pattern and Class Matcher** | |  | |  |  |
| **XII BÖLMƏ. Fayllar, Axınlar və obyektlərin seriallaşdırılması (Files, Streams and Object Serialization).** | | | | | | | |
|  |  | **Files and Streams** | |  | |  |  |
|  |  | **Using NIO Classes and Interfaces to Get File and Directory Information** | |  | |  |  |
|  |  | **Sequential-Access Text Files**   * **Creating a Sequential-Access Text File** * **Reading Data from a Sequential-Access Text File** * **Case Study: A Credit-Inquiry Program** * **Updating Sequential-Access Files** | |  | |  |  |
|  |  | **Object Serialization**   * **Creating a Sequential-Access File Using Object Serialization** * **Reading and Deserializing Data from a Sequential-Access File** | |  | |  |  |
|  |  | **Opening Files with JFileChooser** | |  | |  |  |
|  |  | **(Optional) Additional javaio Classes**   * **Interfaces and Classes for Byte-Based Input and Output** * **Interfaces and Classes for Character-Based Input and Output** | |  | |  |  |
| **XIII BÖLMƏ. Universal Kolleksiyalar (Generic Collections).** | | | | | | | |
|  |  | **Collections Overview** | |  | |  |  |
|  |  | **Type-Wrapper Classes** | |  | |  |  |
|  |  | **Autoboxing and Auto-Unboxing** | |  | |  |  |
|  |  | **Interface Collection and Class Collections** | |  | |  |  |
|  |  | **Lists** | |  | |  |  |
|  |  | **ArrayList and Iterator** | |  | |  |  |
|  |  | **LinkedList** | |  | |  |  |
|  |  | **Collections Methods**   * **Method sort** * **Method shuffle** * **Methods reverse, fill, copy, max and min** * **Method binarySearch** * **Methods addAll, frequency and disjoint** | |  | |  |  |
|  |  | **Stack Class of Package javautil** | |  | |  |  |
|  |  | **Class PriorityQueue and Interface Queue** | |  | |  |  |
|  |  | **Sets** | |  | |  |  |
|  |  | **Maps** | |  | |  |  |
|  |  | **Properties Class** | |  | |  |  |
|  |  | **Synchronized Collections** | |  | |  |  |
|  |  | **Unmodifiable Collections** | |  | |  |  |
|  |  | **Abstract Implementations** | |  | |  |  |
| **XIV BÖLMƏ. Java SE 8 Lambdaları və Axınlar (Java SE 8 Lambdas and Streams).** | | | | | | | |
|  |  | **Functional Programming Technologies Overview**   * **Functional Interfaces** * **Lambda Expressions** * **Streams** | |  | |  |  |
|  |  | **IntStream Operations**   * **Creating an IntStream and Displaying Its Values with the forEach Terminal Operation** * **Terminal Operations count, min, max, sum and average** * **Terminal Operation reduce** * **Intermediate Operations: Filtering and Sorting IntStream Values** * **Intermediate Operation: Mapping** * **Creating Streams of ints with IntStream Methods range and rangeClosed** | |  | |  |  |
|  |  | **Stream<Integer> Manipulations**   * **Creating a Stream<Integer>** * **Sorting a Stream and Collecting the Results** * **Filtering a Stream and Storing the Results for Later Use** * **Filtering and Sorting a Stream and Collecting the Results** * **Sorting Previously Collected Results** | |  | |  |  |
|  |  | **Stream<String> Manipulations**   * **Mapping Strings to Uppercase Using a Method Reference** * **Filtering Strings Then Sorting Them in Case-Insensitive Ascending Order** * **Filtering Strings Then Sorting Them in Case-Insensitive Descending Order** | |  | |  |  |
|  |  | **Stream<Employee> Manipulations**   * **Creating and Displaying a List<Employee>** * **Filtering Employees with Salaries in a Specified Range** * **Sorting Employees By Multiple Fields** * **Mapping Employees to Unique Last Name Strings** * **Grouping Employees By Department** * **Counting the Number of Employees in Each Department** * **Summing and Averaging Employee Salaries** | |  | |  |  |
|  |  | * **Generating Streams of Random Values** * **Lambda Event Handlers** * **Additional Notes on Java SE Interfaces** * **Java 8 SE and Functional Programming Resources** | |  | |  |  |
| **XV BÖLMƏ. Rekursiyalar (Recursion).** | | | | | | | |
|  |  | **Recursion Concepts** | |  | |  |  |
|  |  | **Example Using Recursion: Factorials** | |  | |  |  |
|  |  | **Reimplementing Class FactorialCalculator Using Class BigInteger** | |  | |  |  |
|  |  | **Example Using Recursion: Fibonacci Series** | |  | |  |  |
|  |  | **Recursion and the Method-Call Stack** | |  | |  |  |
|  |  | **Recursion vs Iteration** | |  | |  |  |
|  |  | **Towers of Hanoi** | |  | |  |  |
|  |  | **Fractals**   * **Koch Curve Fractal** * **(Optional) Case Study: Lo Feather Fractal** | |  | |  |  |
|  |  | **Recursive Backtracking** | |  | |  |  |
| **XVI BÖLMƏ. Arama, Çeşidləmə və Böyük O (Searching, Sorting and Big O).** | | | | | | | |
|  |  | **Linear Search** | |  | |  |  |
|  |  | **Big O Notation**   * **O(1) Algorithms** * **O(n) Algorithms** * **O(n2) Algorithms** * **Big O of the Linear Search** | |  | |  |  |
|  |  | **Binary Search**   * **Binary Search Implementation** * **Efficiency of the Binary Search** | |  | |  |  |
|  |  | **Sorting Algorithms** | |  | |  |  |
|  |  | **Selection Sort**   * **Insertion Sort Implementation** * **Efficiency of the Insertion Sort** | |  | |  |  |
|  |  | **Merge Sort**   * **Merge Sort Implementation** * **Efficiency of the Merge Sort** | |  | |  |  |
|  |  | **Big O Summary for This Chapter’s Searching and Sorting Algorithms** | |  | |  |  |
| **XVII BÖLMƏ. Ümumi siniflər və metodlar (Generic Classes and Methods).** | | | | | | | |
|  |  | **Motivation for Generic Methods** | |  | |  |  |
|  |  | **Generic Methods: Implementation and Compile-Time Translation** | |  | |  |  |
|  |  | **Additional Compile-Time Translation Issues: Methods That Use a Type Parameter as the Return Type** | |  | |  |  |
|  |  | **Overloading Generic Methods** | |  | |  |  |
|  |  | **Generic Classes** | |  | |  |  |
|  |  | **Raw Types** | |  | |  |  |
|  |  | **Wildcards in Methods That Accept Type Parameters** | |  | |  |  |
| **XVIII BÖLMƏ. Özəl Ümumi Verilənlərin Strukturu (Custom Generic Data Structures).** | | | | | | | |
|  |  | **Self-Referential Classes** | |  | |  |  |
|  |  | **Dynamic Memory Allocation** | |  | |  |  |
|  |  | **Linked Lists**   * **Singly Linked Lists** * **Implementing a Generic List Class** * **Generic Classes ListNode and List** * **Class ListTest** * **List Method insertAtFront** * **List Method insertAtBack** * **List Method removeFromFront** * **List Method removeFromBack** * **List Method print** * **Creating Your Own Packages** | |  | |  |  |
|  |  | **Stacks** | |  | |  |  |
|  |  | **Queues** | |  | |  |  |
|  |  | **Trees** | |  | |  |  |
| **XIX BÖLMƏ. İstifadəçinin Qrafiki İnterfeysi Komponentləri (GUI Components: Part 2).** | | | | | | | |
|  |  | **JSlider** | |  | |  |  |
|  |  | **Understanding Windows in Java** | |  | |  |  |
|  |  | **Using Menus with Frames** | |  | |  |  |
|  |  | **JPopupMenu** | |  | |  |  |
|  |  | **Pluggable Look-and-Feel** | |  | |  |  |
|  |  | **JDesktopPane and JInternalFrame** | |  | |  |  |
|  |  | **JTabbedPane** | |  | |  |  |
|  |  | **BoxLayout Layout Manager** | |  | |  |  |
|  |  | **GridBagLayout Layout Manager** | |  | |  |  |
| **XX BÖLMƏ. Paralellik (Concurrency).** | | | | | | | |
|  |  | **Thread States and Life Cycle**   * **New and Runnable States** * **Waiting State** * **Timed Waiting State** * **Blocked State** * **Terminated State** * **Operating-System View of the Runnable State** * **Thread Priorities and Thread Scheduling** * **Indefinite Postponement and Deadlock** | |  | |  |  |
|  |  | **Creating and Executing Threads with the Executor Framework** | |  | |  |  |
|  |  | **Thread Synchronization**   * **Immutable Data** * **Monitors** * **Unsynchronized Mutable Data Sharing** * **Synchronized Mutable Data Sharing—Making Operations Atomic** | |  | |  |  |
|  |  | **Producer/Consumer Relationship without Synchronization** | |  | |  |  |
|  |  | **Producer/Consumer Relationship: ArrayBlockingQueue** | |  | |  |  |
|  |  | **(Advanced) Producer/Consumer Relationship with synchronized, wait, notify and notifyAll** | |  | |  |  |
|  |  | **(Advanced) Producer/Consumer Relationship: Bounded Buffers** | |  | |  |  |
|  |  | **(Advanced) Producer/Consumer Relationship: The Lock and Condition Interfaces** | |  | |  |  |
|  |  | **Concurrent Collections** | |  | |  |  |
|  |  | **Multithreading with GUI: SwingWorker**   * **Performing Computations in a Worker Thread: Fibonacci Numbers** * **Processing Intermediate Results: Sieve of Eratosthenes** | |  | |  |  |
|  |  | **sort/parallelSort Timings with the Java SE Date/Time API** | |  | |  |  |
|  |  | **Java SE : Sequential vs Parallel Streams** | |  | |  |  |
|  |  | **(Advanced) Interfaces Callable and Future** | |  | |  |  |
|  |  | **(Advanced) Fork/Join Framework** | |  | |  |  |
| **XXI BÖLMƏ. JDBC ilə Verilənlər Bazasına əlyetərlik (Accessing Databases with JDBC).** | | | | | | | |
|  |  | **Relational Databases** | |  | |  |  |
|  |  | **A books Database** | |  | |  |  |
|  |  | **SQL**   * **Basic SELECT Query** * **WHERE Clause** * **ORDER BY Clause** * **Merging Data from Multiple Tables: INNER JOIN** * **INSERT Statement** * **UPDATE Statement** * **DELETE Statement** | |  | |  |  |
|  |  | **Setting up a Java DB Database**   * **Creating the Chapter’s Databases on Windows** * **Creating the Chapter’s Databases on Mac OS X** * **Creating the Chapter’s Databases on Linux** | |  | |  |  |
|  |  | **Manipulating Databases with JDBC**   * **Connecting to and Querying a Database** * **Querying the books Database** | |  | |  |  |
|  |  | **RowSet Interface** | |  | |  |  |
|  |  | **PreparedStatements** | |  | |  |  |
|  |  | **Stored Procedures** | |  | |  |  |
|  |  | **Transaction Processing** | |  | |  |  |
| **XXII BÖLMƏ. JavaFX İstifadəçinin Qrafiki İnterfeysi (JavaFX GUI: Part 1).** | | | | | | | |
|  |  | **JavaFX Scene Builder and the NetBeans IDE** | |  | |  |  |
|  |  | **JavaFX App Window Structure** | |  | |  |  |
|  |  | **Welcome App—Displaying Text and an Image**   * **Creating the App’s Project** * **NetBeans Projects Window—Viewing the Project Contents** * **Adding an Image to the Project** * **Opening JavaFX Scene Builder from NetBeans** * **Changing to a VBox Layout Container** * **Configuring the VBox Layout Container** * **Adding and Configuring a Label** * **Adding and Configuring an ImageView** * **Running the Welcome App** | |  | |  |  |
|  |  | **Tip Calculator App—Introduction to Event Handling**   * **Test-Driving the Tip Calculator App** * **Technologies Overview** * **Building the App’s GUI** * **TipCalculator Class** * **TipCalculatorController Class** | |  | |  |  |
| **XXII BÖLMƏ. JavaFX İstifadəçinin Qrafiki İnterfeysi (JavaFX GUI: Part 2).** | | | | | | | |
|  |  | **JavaFX Graphics and Multimedia** | |  | |  |  |
|  |  | **Networking** | |  | |  |  |
|  |  | **Java Persistence API (JPA)** | |  | |  |  |
|  |  | **JavaServer™ Faces Web Apps: Part** | |  | |  |  |
|  |  | **JavaServer™ Faces Web Apps: Part** | |  | |  |  |
|  |  | **REST-Based Web Services** | |  | |  |  |
|  |  | **(Optional) ATM Case Study, Part : Object-Oriented Design with the UML** | |  | |  |  |
|  |  | **(Optional) ATM Case Study, Part : Implementing an Object-Oriented Design** | |  | |  |  |
| **CƏMİ:** | | | | | | **75** | **30M/45L** |
| **ƏDƏBİYYAT SİYAHISI** | | | | | |  |  |
| **1. References :**   1. **Robert Lafore, “Object Oriented Programming in C++”, 4th Edition 2002, Sams Publication** 2. **Daya Sagar Baral and Diwakar Baral, “The Secrets of Object Oriented Programming in C++”, 1st Edition 2010, Bhundipuran Prakasan** 3. **Harvey M. Deitel and Paul J. Deitel, “C++ How to Program”, 3rd Edition 2001, Pearson Education Inc.** 4. **Harvey M. Deitel and Paul J. Deitel, “C++ How to Program”, Fifth Edition** 5. **D. S. Malik, “C++ Programming”, 3rd Edition 2007, Thomson Course Technology** 6. **Herbert Schildt, “C++: The Complete Reference”, 4th Edition 2003, Tata McGraw Hill** 7. [**http://www.ioenotes.edu.np/ioe-syllabus/object-oriented-programming-oop-cpp-439**](http://www.ioenotes.edu.np/ioe-syllabus/object-oriented-programming-oop-cpp-439) 8. [**https://docs.oracle.com/javase/tutorial/java/concepts/index.html**](https://docs.oracle.com/javase/tutorial/java/concepts/index.html) 9. [**http://znc.es/Prentice Hall C++ How to Program 5th Edition.pdf**](http://znc.es/Prentice%20Hall%20C++%20How%20to%20Program%205th%20Edition.pdf) 10. [**http://www.ioenotes.edu.np/ioe-bct-syllabus**](http://www.ioenotes.edu.np/ioe-bct-syllabus) 11. [**https://codescracker.com/**](https://codescracker.com/) 12. [**http://www.deitel.com/books/cpphtp5/index.html**](http://www.deitel.com/books/cpphtp5/index.html)   **2. Following web sites may be referred:**   1. [**http://www.osborne.com**](http://www.osborne.com)   **3. The required material for practical can be downloaded from the site:**   1. [**http://www.sun.java.com**](http://www.sun.java.com) | | | | | |  |  |