

# Övningar

## Del 1 – Klasser, objekt och metoder

1. Skapa en klass Person med namn och ålder. Skapa ett objekt och skriv ut dess värden.
2. Skapa en klass Rectangle med bredd och höjd. Lägg till en metod som beräknar arean.
3. Gör en klass Car med attribut för modell, årtal och färg. Skriv en metod printInfo().
4. Skapa en klass Book med titel, författare och antal sidor. Skapa en konstruktor med parametrar.
5. Skriv en metod isOlderThan() i klassen Person som jämför två personer.
6. Skapa en klass TemperatureConverter med statiska metoder för att konvertera mellan Celsius och Fahrenheit.
7. Gör en klass Circle som kan beräkna både area och omkrets.
8. Skapa en klass BankAccount med metoder för deposit(), withdraw() och getBalance().
9. Skriv en klass Counter som håller räkning på antalet skapade objekt (via en statisk variabel).
10. Gör en klass Student med en metod printGrade() som avgör betyg utifrån en poängsumma.
11. Skapa en klass Stopwatch med metoder för att starta, stoppa och beräkna hur lång tid som gått mellan start och stopp.
12. Gör en klass Angle som lagrar en vinkel i grader och har metoder för att lägga till, ta bort och omvandla till radianer.
13. Skapa en klass CharCounter som räknar hur många gånger en viss bokstav förekommer i en textsträng.
14. Gör en klass TimeSpan som representerar ett tidsintervall i timmar och minuter, med metoder för att lägga till och ta bort tid.
15. Skapa en klass Loan med attribut för lånebelopp, ränta och återbetalningstid. Skriv metoder för att räkna ut månadsbetalning och total kostnad.
16. Gör en klass Thermometer som kan ta emot temperaturmätningar och räkna ut medelvärdet av alla mätningar.
17. Skapa en klass SimpleStringBuffer som kan bygga upp en text med metoder som append() och appendLine().
18. Gör en klass RgbColor som lagrar färgvärden i RGB-format (0–255) och har metoder för att räkna ut ljusstyrka och konvertera till hex-format.
19. Skapa en klass LinItem som representerar en rad i ett kvitto med namn, pris och antal. Lägg till metod för att beräkna radens totalsumma.
20. Gör en klass Vector2 som representerar en punkt eller vektor i 2D med metoder för att beräkna längd, avstånd och lägga ihop vektorer.

## **Del 2 – Inkapsling och konstruktörer**

21. Gör alla attribut i BankAccount privata och lägg till getters/setters.
22. Skapa validering i withdraw() så att saldot inte kan bli negativt.
23. Skapa en klass Product med pris och namn. Lägg till metod applyDiscount(double percent).
24. Gör en överlagrad konstruktor till Product där priset sätts till 0 om det inte anges.
25. Skapa en klass Movie och en metod getAgeLimit() som returnerar 7, 11 eller 15 beroende på genre.
26. Skapa en klass Employee med lön. Lägg till en metod raiseSalary(double percent).
27. Skriv en klass Time med timmar och minuter, och en metod addMinutes(int m).
28. Gör en klass Address och använd den som attribut i klassen Person.
29. Skapa en klass Order som innehåller flera Product-objekt (utan att använda Collections).
30. Skriv en metod printReceipt() i Order som summerar alla produkters pris.

## **Del 3 – Arv och polymorfism**

31. Skapa en basklass Animal med metoden makeSound(). Låt Dog och Cat ärva och överskugga metoden.
32. Skapa en klass Shape med metoden area(). Låt Circle och Rectangle ärva och implementera.
33. Gör en klass Vehicle och låt Car och Bicycle ärva. Lägg till attribut som är specifika för varje fordon.
34. Skapa en klasshierarki Employee → Manager → Director med unika metoder.
35. Skapa en lista med Animal-objekt och låt alla anropa makeSound() (demonstrera polymorfism).

## **Del 4 – Felhantering, Array & Collections**

36. Modifiera BankAccount så att withdraw() kastar ett InsufficientFundsException vid för stort uttag.
37. Skapa en klass Calculator med metoder som kan kasta ArithmeticException.
38. Skapa en array av Student-objekt och skriv ut alla med betyg > 70.
39. Gör en ArrayList<Product> och skriv ut alla produkter dyrare än 100 kr.
40. Lägg till en metod som sorterar produkterna i listan efter pris.

## **AVANCERAD NIVÅ**

### **Del 5 – Abstraktion, interface och polymorfism**

41. Skapa ett interface Payable med metoden calculatePayment(). Implementera i Employee och Freelancer.
42. Skapa en abstrakt klass Shape med area() och perimeter(). Implementera i Triangle och Square.
43. Skapa ett interface Playable och implementera det i Game, Song och Video.
44. Gör en klasshierarki Vehicle → Car → ElectricCar med överskuggade metoder och super().
45. Skriv en metod som tar en List<Shape> och skriver ut total area för alla former.
46. Implementera Comparable<Product> så att produkter kan jämföras efter pris.
47. Skriv ett program där polymorfism används för att hantera både Dog, Cat och Bird i en lista.

### **Del 6 – Objektorienterad design och problemlösning**

48. Skapa en enkel klassdiagramskiss för ett bibliotekssystem (Book, Borrower, Loan). Implementera koden.
49. Bygg ett litet faktureringsystem med klasserna Customer, Invoice och Item.
50. Skapa ett litet registreringssystem där användaren kan lägga till och ta bort Student-objekt via konsolen.
51. Implementera ett enkelt medlemsregister där Member kan vara av typen Regular eller VIP.
52. Gör en klass ShoppingCart med metoder för att lägga till/ta bort produkter och beräkna totalpris.
53. Implementera en klass Tournament med spelare och resultat, och beräkna vinnaren.
54. Skapa en klass Logger som loggar händelser till en textfil.
55. Gör ett enkelt menyprogram där användaren kan skapa, lista och ta bort Book-objekt.

### **Del 7 – Sammanhängande och tankekrävande**

56. Implementera ett litet djursystem där djur har olika beteenden beroende på tid på dygnet.
57. Skapa en klass Deck som innehåller 52 Card-objekt och metoder för att dra kort slumpmässigt.
58. Implementera en klass BankSystem som hanterar flera konton och överföringar mellan dem.
59. Bygg ett enkelt bokningssystem för en biograf med klasserna Movie, Seat och Booking.
60. Gör en mini-applikation för en digital ordbok (Dictionary) med sök, lägg till, ta bort och lista ord.
61. Bibliotek v2: reservationskö – Utöka bibliotekssystemet med reservationskö per bok och notifiering när ett exemplar lämnas tillbaka.
62. Restaurangbokning – Klasser för Table, Reservation, Customer. Hantera krockar, maxsällskap och väntelista.

63. Parkeringshus – Hantera våningsplan, platstyper (handikapp/EV/vanlig), in-/utpassering och avgiftsberäkning per påbörjad timme.
64. Tågplanerare – Train, Route, Stop, Ticket. Kontrollera byten, restid, platsbokning och prissteg enligt zoner.
65. E-lärande-quiz – Course, Lesson, Question, Attempt. Beräkna poäng, tidstak, godkänthet och resultatrapport per student.
66. Huslån-simulator – Hantera flera lån med olika räntor, amorteringsmodeller och extraamorteringar. Visa betalplan och totalkostnad.
67. Förrådsinventering – Item, Batch, Location. Stöd för in-/utleverans, bäst-före, batchspårning och lågsaldovarningar.
68. Turneringsschema (round-robin) – Team, Match, Schedule. Generera schema utan dubbelbokning, tabellpoäng och målskillnad.
69. Energimätning i villa – Meter, Reading, Tariff. Timvärden, toppavgift vid hög belastning, månadsfaktura och jämförelse mot föregående månad.
70. Kollektivtrafikbiljett – Card, Trip, Validator. Zonavdrag, dagstak (cap), periodbiljettprioritet och spärr mot negativt saldo.