A2

2. Download eclipse from http://www.eclipse.org/downloads/ (Links to an external site.) and install it

3.

a. Create a new Java project in eclipse

b. Create a new Java file (see the video “Shoe”)

c. Implement a class Cat – the cat has a name.

d. Write a program where the user can enter the name and assign the name to a cat.

e. Print the name of the cat

4. Design a use case diagram for a photographer. The photographer will take pictures and change flash cards – can you think of other things a photographer might do?

Write an application that inputs three integers from the user and displays the sum, average, product, smallest, and largest of the numbers. To determine the largest and smallest number use Math.min(x, y) and Math.max(x,y) – see https://examples.javacodegeeks.com/core-java/math/find-minimum-of-numbers-with-math-min/ (Links to an external site.) for more details on those.

5. Write an application that inputs one number consisting of five digits from the user, separate the number into its individual digits and prints the digits separated from one another by three spaces each. For example, if the user types in the number 42339, the program should print 4 2 3 3 9 – You will need to use:

a. The modulo operator % which is nothing more than “remainder after division”, for instance 12 % 10 would be 2

b. Integer division /, for instance 12 / 10 would be 1

A3

2.

a. Explain the purpose of an instance variable.

b. Why should instance variables be initialized (set to a value) in the constructor? (If you don’t know the answer try it out and don’t initialize the variable and print it out)

3. Solve this:

The Account class models a bank account. Identify which type of variable

(temporary, parameter, or instance) should be used in each of the following values.

Justify your answers using Table 6-4.

a. The bank account’s balance.

b. The amount to deposit in the account.

c. The account’s current interest rate.

d. The amount of interest earned in the last month.

表格

描述已自动生成

4. Design a UML Diagram for a classes Cat and Fish. The Cat has a name, can miow, catch the fish, and sleep. The fish has a name, can swim, and eats. What do those classes have in common? Design an abstract class Pet they can inherit from.

5. Consider the Shoe and RunningShoe example.

a. Add two new classes DressShoe and WorkingBoot inheriting from Shoe. Write a main method in each which asks the size from the user.

b. Add a new attribute/instance variable color of type String to Shoe. Change the constructors so you can initialize each shoe with its color.

6. Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables – a part number (type String), a part description (type String), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a method named getInvoiceAmount that calculates the invoicd amount (i.e. multiplies the quantity by the price per item), then prints the amount as a double value. If the quantity is not positive, it should be set to 0 (consider the Math.max method – see http://www.java-examples.com/find-maximum-two-numbers-using-math.max (Links to an external site.))If the price per item is not positive, it should be set to 0.0.Write a test application named InvoiceTest that demonstrates the Invoice’s capabilities. Make sure to create this Java file in the same directory as Invoice. You can then refer to Invoice like you would in Invoice.java, e.g. Invoice invoice = new Invoice()

A4

2. Write a program with several functions as shown in the example videos and call them from a main method for testing:

a. f(x) = 2x +3

b. f(x) = x^2 + 3x + 9

c. f(x) = x^3 + 4x + 7

3. Create a class Giraffe which stores the height, age, and name of the Giraffe. Add acessor methods (getters/setters) which show those values but also let you set them. Write another method walk which takes a number of steps and prints “Giraffe is walking X steps”.

4. Create a class ColaVendingMachine. This class is simulating a cola vending machine. It keeps track of how many cola bottles are in the class and how much one bottle costs. There should be a method sellBottle which sells one bottle to a customer, decreases the amount of bottles left. There also is a method restock which sets the number of bottles to the number it is restocked to. Write a main method to test the functionality of the ColaVendingMachine machine.

A5

2. Submit a Truth table for the NAND function which is basically NOT AND. (see <https://en.wikipedia.org/wiki/Sheffer_stroke>)

3. Write a class Dog which has an instance variable size. Add a method bark which prints the bark. The bark is dependent on size:

0<size<=5 -> Yip! Yip!

5<size<=10 -> Ruff! Ruff!

10<size -> Woof! Woof!

4. Write a program (main method) that advises the user on programming language. So if a user for instance enters “Java” the program might say “awesome” or if the user enters “Ruby” it might say “are you sure?” . Make the program comment on at least 5 programming languages.

A6

2. Look at the binary search algorithm. Explain how it fulfills each of the five algorithmic properties (see slides).

3. Write a program which counts from your birthday the years until today. For example if you are born in 2015 it would output:

2015

2016

4. Write a simple guessing game:

a. Generate a random number between 1 and 10 by using java.util.Random:

Random r = new Random()

int i = r.nextInt(10)+1

b. Ask the user for his guess

c. Use a while loop to loop until the user has guessed the correct number

5. Implement the simple search (see slides – the one which uses a loop) and search for the first letter a in the text “This class is awesome”. Utilize charAt (see http://www.tutorialspoint.com/java/java\_string\_charat.htm (Links to an external site.) ) to obtain a character to test.

A7

2. Explain the difference between a list and an Array. Give examples when you would use one or the other.

3. Explain the difference between the while- and for-loop. Give examples when you would use one or the other.

4. Describe why looking at ADT can help with deciding which one is best for the task.

5. Write a program to store a positive number a user inputs until he enters 0. Print all the numbers entered at the end. The dialog might look like:

Enter Number: 1

Enter Number: 10

Enter Number: 0

You entered numbers 1, 10

Select an appropriate data structure for this program.

6. Simple search: Search for the number 7 in an array of 10 random numbers. You will need to declare an array, use a loop to assign a random number and then use another loop to search for the number 7 in the array by investigating each array element.

A8

2. Write a program to simulate a login. Create a class user which has a user name, a password, and a role. Initialize it with 3 users: Bob, role user; Jim role user; Liz role super user. Select appropriate passwords. Now, ask the user to input his username and password. If they are correct print “Welcome <username>!” otherwise “Access Denied!”. A dialog might look like:

Login: Bob

Password: <password>

Welcome Bob!

Alternatively:

Login: Bob

Password: <wrong password>

Access Denied!

Use a Map to solve this task.

3.Modify the program from (2) to use Exceptions. If the user enters the wrong password throw an AccessDenied exception.

4.Implement the Factorial (see https://en.wikipedia.org/wiki/Factorial (Links to an external site.) ) as a recursive Java function. For example factorial(2) should return 2

5. Explain the difference between stack and heap.

A9

2. Review the definitions for Interface, Abstract Class, and Concrete Class – give an example when you would use each of them.

3. Solve exercise 12.10 on p. 692 of “Introduction to programming with Robots”

Some courses assign letter grades, whereas other courses assign a percentage between 0 and 100. Still others assign a pass/fail grade.

Write an interface named Grade. The toPercent method returns the grade as an integer percentage between 0 and 100 percent. The toString method prints the grade in its “native” format (a percentage, a letter grade, or either “Pass” or “Fail”). The isPass method returns true for a passing grade, false otherwise. The includeInAverage returns true for letter and numeric grades, but false for pass/fail grades.

Write three classes that implement Grade: LetterGrade, PercentageGrade, and PassFailGrade. Write a main method that fills an array with grades. For each grade, print on one line the native format, “Pass” or “Fail” (as appropriate), and the percentage (if it can be included in an average). After the list of grades, print the average grade as a percentage.

Use your school’s mapping between letter grades and numeric grades, if it has one. Otherwise, make up something like A+ is 95%, A is 90%, etc.

4. Based on what we learned in chapter 11 about tests consider the DressShoe class from A3(A3那个问题):

a. Develop a few test scenarios

b. Implement an extra test class which will be running those scenarios against the DressShoe class