

Example exam questions Programming (2IP05)

1 Perfect numbers

Make for this problem a class `Perfect`. Submit this class as the file `Perfect.java`.

A *perfect number* is a natural number that is equal to the sum of its divisors, not including itself.

The operator `%` computes *remainder* after division.

1. Write a method `int[] trueDivisors(int n)` that calculates the *true divisors* of a positive integer n and returns them as an array. A true divisor of n is a positive divisor unequal to n . Example:

`trueDivisors(10)` returns the array: { 1, 2, 5 }

2. Write a method `int arraySum(int[] a)` that calculates and returns the sum of the elements in array `a`. The sum of an empty array is 0. You may assume that `a` is not `null`.
3. Write a method that determines whether a given positive number is a perfect number and returns this as a boolean. Use the methods of the previous parts.
4. Write a method that prints the perfect numbers between 1 and including n for a given integer n . Each perfect number should be printed on a new line. Nothing should be printed when there are no perfect numbers in the interval.
5. Give for each method pre- and postconditions.

2 Humor

A humorist wants to keep track of his jokes with the help of a Java program.

1. Write a class `Joke` with instance variables for the text of the joke (a `String`), the time it takes to tell the joke (in seconds), and a `double` that represents the funniness of the joke on a scale of 0 to 10. Override the method `String toString()` in order to give a well readable `String` representation of the joke.
2. Add and implement a class `Repository` that contains jokes. Include methods to add and remove jokes.
3. The humorist gives monologues (Dutch: *conferences*) where he basically tells a subset of the jokes in a certain order. Furthermore, he has two special types of jokes: dirty jokes and insulting jokes. Dirty jokes are only suited for people above a certain age, depending on the joke. Insulting jokes are jokes that are insulting to one or more population groups. When the humorist prepares a monologue for a certain audience, it is important to know for which groups the joke is insulting.

Complete the class design. Use inheritance. Give instance variables and methods and describe them. You do not have to give implementations of the methods (you can leave the bodies empty).
