Additional exercises

Sead Jahić
Teaching Assistant, Information Technologies

University of Primorska
Faculty of Mathematics, Natural Sciences and Information Technologies
(UP FAMNIT)

19. studenoga 2022.

Exercise

Implement program that will count binomial coefficient.

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}, \quad \binom{n}{0} = \binom{n}{n} = 1$$

Binomial coefficients appears in Newton-binom equation:

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

- create scanner, give opportunity to insert *n*,*k*.
- create functions:
  - factorial(input n)
  - binomial_coef(input n, k)
- use if statement to avoid input of negative number or zero

Exercise

Fibonacci sequence!

The Fibonacci Sequence is the series of numbers:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...$$

The next number is sum of the two numbers before it.
The sequence F_n of Fibonacci numbers is defined by the recurrence relation:

$$F_n = F_{n-1} + F_{n-2},$$

$$F_0 = 0, F_1 = 1.$$

for more information about Fibonacci sequence see `https://en.wikipedia.org/wiki/Fibonacci_number`.

Exercise

Another great thing to implement in JAVA as program is Horner's method!

Horner's method is used to calculate remainder of polynomial division (remainder of $f(x)$ on division by $x - \alpha, \alpha$ is integer).

	a_n	a_{n-1}	a_{n-2}	$\cdots$	a_0
α	b_n	b_{n-1}	b_{n-2}	$\cdots$	b_0
	$=$	$=$	$=$	$\cdots$	$=$
	a_n	$\alpha \cdot bn + a_{n-1}$	$\alpha \cdot b_{n-1} + a_{n-2}$	$\cdots$	$\underbrace{\alpha \cdot b_1 + a_0}_{R-remainder}$

For example: Find remainder of division polynomial $P(x) = 4x^4 + 3x^3 - 2x^2 + x + 2$ with $Q(x) = x - 2$.

$A[0] = B[0]$

	4	3	-2	1	2	=A
2	4	11	20	41	84	=B
= α		$B[1] = \alpha * B[0] + A[1]$				
		$11 = 2 * 5 + 3$				

Remainder can be also found using Bézout's theorem. It states that the remainder of the division of a polynomial $f(x)$ by a linear polynomial $x - r$ is equal to $f(r)$.
In our example it means:

$$P(2) = 4 \cdot 2^4 + 3 \cdot 2^3 - 2 \cdot 2^2 + 1 \cdot 2^1 + 2 \cdot 2^0 = 84.$$

Here is solution:

```
public static int Bezuot(int n, int alpha, int[] coeff){
    int sum=0;
    for (int i=0; i<=n; i++){
        sum = (int) (sum+coeff[i]*Math.pow(alpha,n-i));
    }
    return sum;
}
```