

## Blatt 5

### Aufgabe 17

```
> restart:
```

```
(a)
```

```
> f := x -> cos(5 * arccos(x)) + x: 'f'(x) = f(x);  
> df := x -> diff(f(x), x): 'df'(x) = df(x);  
> d2f := x -> diff(f(x), x$2): 'd2f'(x) = d2f(x);  
> plot([ f(x), df(x) / 10, d2f(x) / 100 ], x = -1..1, y = -1.5..  
.1.5, color = [ blue, red, green ], legend = [ 'f', 'Df/10',  
'D^2*f/100' ]);
```

```
(b)
```

```
> g := x -> (x^4 - 7 * x^2 + 3) / (x^4 + 5): 'g'(x) = g(x);  
> dg := diff(g(x), x): 'dg'(x) = dg(x);  
> d2g := diff(g(x), x$2): 'd2g'(x) = d2g(x);  
> plot([ g(x), dg(x), d2g(x) ], x = -5..5, color = [ blue, red,  
green ], legend = [ 'g', 'Dg/10', 'D^2*g/100' ]);
```

### Aufgabe 18

```
> restart:
```

```
> print("Zahl", "Gerade", "Prim", "Natürlich", "Durch 3 Teilbar")  
;  
> for number in seq(kk, kk = -10..10) do  
    print(number, is(number, even), is(number, natural), is  
(number, prime), is(number/3, integer));  
end do;
```

### Aufgabe 19

```
> restart:
```

```
(a)
```

```
> ggt := proc (m::integer, n::integer)::integer;  
    description "Berechnet den größten gemeinsamen Teiler zweier  
natürlicher Zahlen";  
    local j := m, k := n, l := 1;  
    if not is(m, natural) or not is(n, natural) then  
        error("Erlaube für m und n nur natürliche Zahlen.");  
    end if;  
    while not l = 0 do  
        l := j mod k;  
        #print('j' = j, 'k' = k, 'j mod k' = l);  
        j := k;  
        k := l;  
    end do;
```

```

    # Die letzte Ausgabe ist die Ausgabe der gesamten Funktion
    j;
end proc:
> # # Test
> # 'ggT(2, 3)' = ggT(2, 3);
> # ggT(4, 2);
> # ggT(2, 4);
> # ggT(1, 2);
> # ggT(2, 1);
(b)
> ggT(19278, 3234);

```

## Aufgabe 20

```

> restart:
> ?rem
> ?degree
> ggTpoly := proc(p::polynom, q::polynom)::polynom;
    description "Berechnet den größten gemeinsamen Teiler zweier
    Polynome";
    local r := p, s := q, t := x;
    while degree(t, x) >= 0 do
        t := rem(r, s, x);
        #print(r, s, t);
        r := s;
        s := t;
    end do;
    # Ausgabe, normalisieren (Führkoeffizient 1)
    r := r / lcoeff(r);
end proc:
> # # Test
> # simplify(ggTpoly(x^3 - 1, x - I^(4/3)) - gcd(x^3 - 1, x - I^
(4/3)));
> # simplify(ggTpoly(x^3 - 1, x - I^(4/3)) - ggTpoly(x - I^(4/3),
x^3 - 1));
(b)
> p := x^4 + x^3 - x^2 + x + 2:
> q := x^3 + 2*x^2 + 2*x + 1:
> 'ggT'(p, q) = ggTpoly(p, q);
> p := x^4 + x^3 + x + 1:
> q := x^2 - 1:
> 'ggT'(p, q) = ggTpoly(p, q);

```