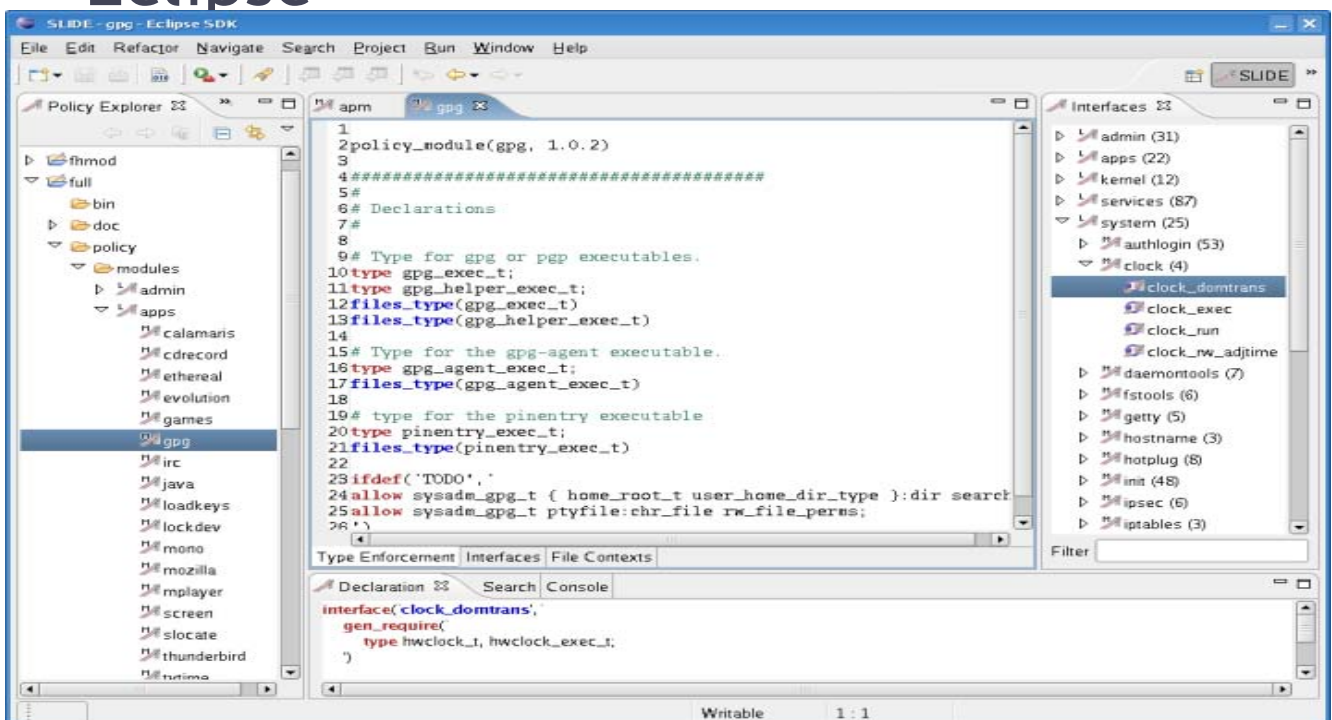


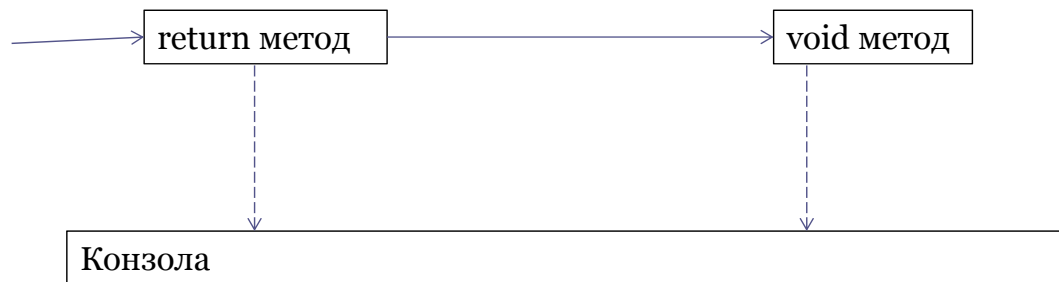
# Code@Feit

# Eclipse



# Return vs. void

- `return "feit";`
- `System.out.println("feit");`



# Од претходниот час

```
public static int faktorijal_iter(int n){
    int k=1;
    while (n>0){
        k=k*n;
        n--;
    }
}
```

```
public static int faktorijal_rec(int n){
    if (n==0) {
        return 1;
    } else {
        return n*faktorijal_rec(n-1);
    }
}
```

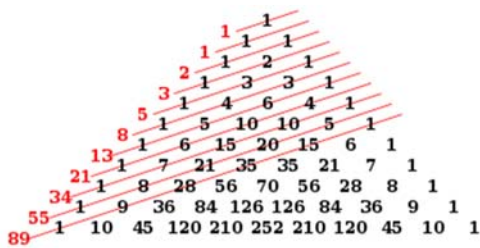
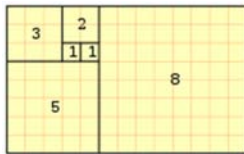
## Степен

```
public static int pow(int n, int i) {
    if (i == 0)
        return 1;
    else if (i == 1)
        return n;
    else {
        return n*pow(n, i-1);
    }
}
```

## Степен

```
public static int pow(int n, int i) {
    if (i == 0)
        return 1;
    else if (i == 1)
        return n;
    else {
        int partial = pow(n, i / 2);
        if (i % 2 == 0)
            return partial * partial;
        else
            return partial * partial * n;
    }
}
```

# Fibonacci



# Fibonacci итерација

```
int a,b,t;  
for (int i=2; i<n; i++){  
    if (n==0 || n==1) {  
        return n;  
    }  
    t=a+b;  
    a=b;  
    b=t;  
}
```

## Фібоначчі рекурзивно

```
public static int recFib(int n){
    if (n==0 || n==1) {
        return n;
    }
    else {
        return recFib(n-1)+recFib(n-2);
    }
}
```

## Квадратен корен (2)

```
public static int stepen(int n, int down, int up){
    if (down*down == n) {
        return down;
    } else if (up*up == n) {
        return up;
    } else if ((up-down)==1){
        return down;
    } else {
        int mid=(up+down)/2;
        if (mid*mid==n) {
            return mid;
        } else if (mid*mid > n) {
            return stepen(n, down, mid);
        } else {
            return stepen(n, mid, up);
        }
    }
}
```