interpreteri i kompajleri

bebić / rač @ petnica / mart 2024

kako ovo radi ?

```c
// main.c

int main() {
    printf("Hello, world!\n")
    return 0;
}
```

kako *ovo* radi ?

```
print("Hello, world!")
```

šta je zapravo `python` ?

kako radi interpreter ?

šta je zapravo `.py` fajl ?

```python
print("Hello, world!")
```

je zapravo

```
112 114 105 110 116 40 34  72
101 108 108 111 44  32 119 111
114 108 100 33  34  41
```

moramo da *razumemo* fajl

```
print("Hello, world!", 1 + 2)
```

```
'print' '(' '"Hello, world!"' ',' '1' '+' '2' ')'
```

```
CALL
    VARIABLE 'print'
    STRING   'Hello, world!'
    ADD
        NUMBER 1
        NUMBER 2
```

leksiranje

```java
public enum TokenType {
    IDENTIFIER, NUMBER, STRING,
    LPAREN, RPAREN, COMMA, PLUS,
    EOF
}

public class Token {
    public TokenType type;
    public String value;
}
```

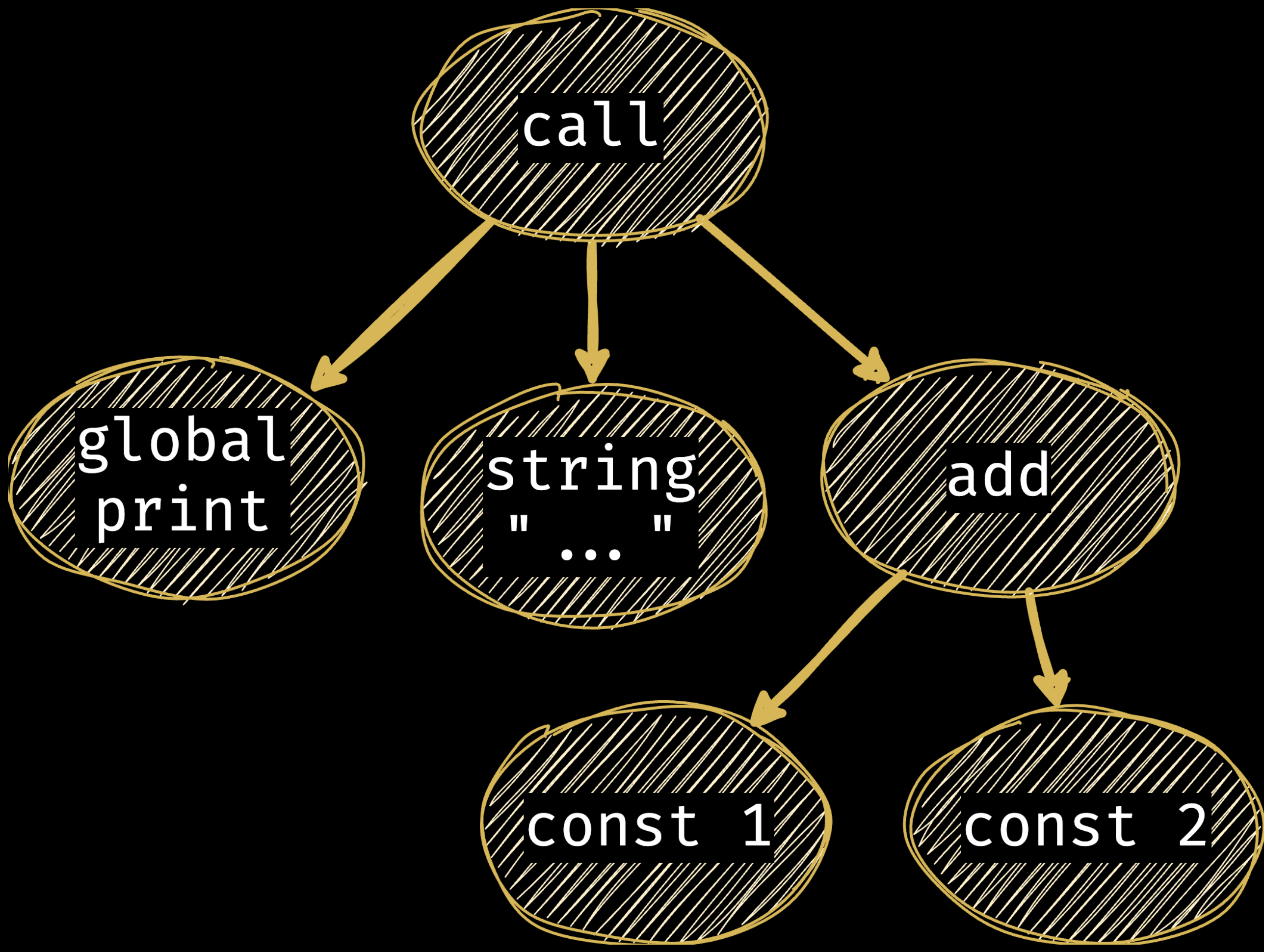
```
Token nextToken() {
    char currentChar = nextChar();
    if (currentChar == '(') {
        return new Token(TokenType.LPAREN, "(");
    // ...
// ...
if (isAlpha(currentChar)) {
    String ident = "" + currentChar;
    while (isAlphaNum(peekChar())) {
        ident += nextChar();
    }
}
// ...
```

parsiranje

```java
public Ast parseAdd() {
    Ast lhs = parseExpression();
    parseToken(TokenType.PLUS);
    Ast rhs = parseExpresion();

    return new AddNode(lhs, rhs);
}
```

ast



```java
class VariableNode extends Ast {
    private String name;
}
```

```java
class AddNode extends Ast {
    private Ast lhs;
    private Ast rhs;
}
```

šta sad ?

izvršavanje

```java
public interface Expression {
    Object execute(Context ctx);
}
```

```java
// class AddNode
public Object execute(Context ctx) {
    Object lhs = this.lhs.execute(ctx);
    Object rhs = this.rhs.execute(ctx);

    return (double)lhs + (double)rhs;
}
```

šta je `Context` ?

```java
public class VariableNode {
    private String name;

    public Object execute(Context ctx) {
        return ctx.getVariable(name);
    }
}
```

kako implementiramo `if` ?

```java
public class IfNode {
    private Expression condition;
    private BlockNode thenBlock;
    private BlockNode elseBlock;

    public Object execute(Context ctx) {
        if ((boolean)condition.execute(ctx)) {
            thenBlock.execute(ctx);
        } else {
            elseBlock.execute(ctx);
        }
    }
}
```

tipovi

kompajleri

šta je `.exe` zapravo ?

kako pravimo mašinski kod ?

alokacija registara

```
interface Expressions {
   Register generate(Context ctx);
}
```

```
class IntConstNode extends Ast {
    private int value;
    Register generate(Context ctx) {
        Register reg = ctx.allocateRegister();
        ctx.generate("mov", reg, value);
        return reg;
    }
}
```

```
class AddIntNode {
    private Expression lhs;
    private Expression rhs;

    Register generate(Context ctx) {
        Register lhs = this.lhs.generate(ctx);
        Register rhs = this.rhs.generate(ctx);
        ctx.generate("add", lhs, lhs, rhs);
        return lhs;
    }
}
```