

Генерация на код

Цел на упражнението

Упражнението представя структурата на генератора на код в учебния компилатор.

На входа на генератора се получава семантично коректно дърво и символната таблица. На базата на тези структури той формира инструкции за изпълнение от виртуалната машина.

Пример за генерация на код при извикване на функция:

```
public void visit(FunctionCall node) {
    List<AssignableNode> params = node.getActualParameters()
                                   .getParameters();

    int paramsCount = 0;
    if (params != null) {
        paramsCount = params.size();
        params.forEach(p -> p.accept(this));
    }
    code[pointer++] = OperationCode.ICONST.ordinal();
    code[pointer++] = paramsCount;
    code[pointer++] = OperationCode.CALL.ordinal();
    code[pointer++] =
helper.getFunctionAddress(node.getToken().getText());
}
```

Примери за аритметични операции:

- събиране

```
public void visit(AdditionNode node) {
    node.getChildNodes().forEach(n -> n.accept(this));
    code[pointer++] = OperationCode.IADD.ordinal();
}
```

- изваждане

```
public void visit(SubtractionNode node) {
    node.getChildNodes().forEach(n -> n.accept(this));
    code[pointer++] = OperationCode.ISUB.ordinal();
}
```

- умножение

```
public void visit(MultiplicationNode node) {
    node.getChildNodes().forEach(n -> n.accept(this));
    code[pointer++] = OperationCode.IMUL.ordinal();
}
```

- деление

```
public void visit(DivisionNode node) {
    node.getChildNodes().forEach(n -> n.accept(this));
    code[pointer++] = OperationCode.IDIV.ordinal();
}
```

- деление с остатък

```
public void visit(ModNode node) {
    node.getChildNodes().forEach(n -> n.accept(this));
    code[pointer++] = OperationCode.IREM.ordinal();
}
```

Примери за логически операции:

- логическо И (AND)

```
public void visit(AndNode node) {
    node.getChildNodes().forEach(n -> n.accept(this));
}
```

```
    code[pointer++] = OperationCode.IAND.ordinal();
}
```

- логическо ИЛИ (OR)

```
public void visit(OrNode node) {
    node.getChildNodes().forEach(n -> n.accept(this));
    code[pointer++] = OperationCode.IOR.ordinal();
}
```

Пример за оператор while:

```
public void visit(WhileStatementNode node) {
    int gotoExpression = pointer;
    node.getExpression().accept(this);
    code[pointer++] = OperationCode.IF_FALSE.ordinal();
    int jumpAddr = pointer;
    code[pointer++] = -1;
    node.getBlock().accept(this);
    code[pointer++] = OperationCode.GOTO.ordinal();
    code[pointer++] = gotoExpression;
    code[jumpAddr] = pointer;
}
```

Задачи:

1. Да се генерира код за програма на STUDENT, която извежда на екран символен низ „Hello world!“.
2. Да се генерира код за програма на STUDENT, която извежда сумата на целите числа от 1 до 10.
3. Да се генерира код за програма на STUDENT, която извежда стойността на факториел за число, подадено като параметър на функция, като се използва итеративен алгоритъм.

4. Да се генерира код за програма на STUDENT, която извежда стойността на факториел за число, подадено като параметър на функция, като се използва рекурсивен алгоритъм.

5. Да се генерира код за програма на STUDENT, която извежда на екран 10 цели числа, записани в масив.

6. Да се генерира код за програма на STUDENT, която сортира масив от 10 цели числа във възходящ ред.