به نام خدا

نام: فاطمه زهرا بخشنده **استاد: دکتر سعید پارسا**

شماره دانشجویی: 98522157 **درس اصول طراحی کامپایلر**

گزارش تمرین کلاسی 2:

سوال: گرفتن یک برنامه جاوا یا C# یا C++ و تبدیل کامنت های یک خطی به چند خطی، و افزودن نام و شماره دانشجویی به ابتدای همه کامنت ها.

کد این تمرین در refactor.py قرار دارد. گرامر های جاوا در پوشه grammars اضافه شدند و فایل های lexer و مابقی فایل های ساخته شده از گرامر نیز در پوشه gen موجود است.

ابتدا با استفاده از argparser آدرس فایل ورودی، آدرس فایل خروجی که کد ریفکتور شده را در آن می‌نویسیم، و نام و شماره دانشجویی را ورودی می گیریم. برای همه این مقادیر، مقدار دیفالت نیز قرار داده شده است، و help نیز ایجاد شده است.

```
arg_parser = argparse.ArgumentParser(description='Java Refactor Comments')
arg_parser.add_argument(
    '-n', '--file',
    help='input java file path',
    default=r"Game.java"
)
arg_parser.add_argument(
    '-out', '--out',
    help='output java file path',
    default=r"GameRefactored.java"
)
arg_parser.add_argument(
    '-name', '--name',
    help='your full name',
    default=name
)
arg_parser.add_argument(
    '-number', '--number',
    help='your student number',
    default=number
)
```

```
PS D:\uni\compiler\class\C2> python refactor.py -h
usage: refactor.py [-h] [-n FILE] [-out OUT] [-name NAME] [-number NUMBER]

Java Refactor Comments

optional arguments:
  -h, --help            show this help message and exit
  -n FILE, --file FILE  input java file path
  -out OUT, --out OUT   output java file path
  -name NAME, --name NAME
                        your full name
  -number NUMBER, --number NUMBER
                        your student number
```

سپس تابع refactor را با آرگیومنت های دریافت شده فراخوانی می کنیم. در تابع refactor با استفاده از حرکت روی token ها، کامنت های یک خطی line comments را تبدیل به کامنت چند خطی block comments می کنیم و به ابتدای همه کامنت ها (خط اول آن ها) نیز نام کامل و شماره دانشجویی وارد شده را اضافه می کنیم. که مقدار دیفالت آن نام و شماره دانشجویی خودم هست. کد refactor شده را نیز در آدرس فایل مقصد ذخیره می کنیم. همچنین محتوای کد ریفکتور شده را در خروجی چاپ می کنیم.

در انتها یک کد ریفکتور شده داریم که همه کامنت های آن از نوع block comment است، و خط اول همه آن ها نیز نام و شماره دانشجویی وارد شده هست.

عکس از نحوه ورودی دادن به برنامه:

```
PS D:\uni\compiler\class\C2> python refactoring.py -n Game.java -out GameRefactored.java -name 'FatemeZahra Bakhshande' -number 98522157
import java.util.Random;
 import java.util.ArrayList;

 /*FatemeZahra Bakhshande 98522157
 it was a line comment*/
 public abstract class Piece
 {
     protected String name;
     protected int x;
     protected int y;
     protected boolean[][] flag = new boolean[8][4];

     protected Piece(String name, int x, int y)
     {
         this.name = name;
         this.x = x;
         this.y = y;
     }

     protected abstract void move();
 }
```

آدرس فایل ورودی آن فایل Game.java است که در همین zip موجود است. آدرس فایل خروجی آن هم فایل GameRefactored.java است که در همین پوشه ایجاد شده است. و می بینیم که کد ریفکتور شده در آن قرار دارد.

خروجی کامل (محتوای فایل GameRefactored.java):

```java
import java.util.Random;
import java.util.ArrayList;

/*FatemeZahra Bakhshande 98522157
it was a line comment*/
public abstract class Piece
{
    protected String name;
    protected int x;
    protected int y;
    protected boolean[][] flag = new boolean[8][4];

    protected Piece(String name, int x, int y)
    {
        this.name = name;
        this.x = x;
        this.y = y;
    }

    protected abstract void move();
}

/* FatemeZahra Bakhshande 98522157
 it was
a block comment
*/

public class Castle extends Piece
{
    protected Castle(String name, int x, int y)
    {
        super(name, x, y);
        this.flag[this.y][this.x] = true;
    }

    /* FatemeZahra Bakhshande 98522157
      override method move
      use switch case
```

```
    move0
    move1
    move2
*/
@Override
protected void move()
{
    switch ((int) (Math.random() * 3))
    {
        case 0: move0(); break;
        case 1: move1(); break;
        case 2: move2(); break;
    }
}

private void move0()
{
    /*FatemeZahra Bakhshande 98522157
   check if y < 7*/
    if (this.y < 7)
    {
        System.out.printf("x:%d y:%d ==> ", this.x, this.y);
        this.y++;
        this.flag[this.y][this.x] = true;
        System.out.printf("x:%d y:%d\n", this.x, this.y);
    }
    else move();
}

/*FatemeZahra Bakhshande 98522157
   it was a line comment before move1*/
private void move1()
{
    if (this.x < 3)
    {
        System.out.printf("x:%d y:%d ==> ", this.x, this.y);
        this.x++;
        this.flag[this.y][this.x] = true;
        System.out.printf("x:%d y:%d\n", this.x, this.y);
    }
    else move();
}
/* FatemeZahra Bakhshande 98522157
   it was a block commrnt
   bellow is move2 method
```

```
    */
    private void move2()
    {
        if (this.x > 1)
        {
            System.out.printf("x:%d y:%d ==> ", this.x, this.y);
            this.x--;
            this.flag[this.y][this.x] = true;
            System.out.printf("x:%d y:%d\n", this.x, this.y);
        }
        else move();
    }
}
```