

Marcus Lund - 911118 1153  
Object-Oriented Programming and Design  
Håkan Jonsson

Lab 1 - D0010E

# Prog1

**package** Uppgifter;

/\*

\* Av: Marcus Lund

\* 911118-1153

\* 2015-01-20

\*/

**public** **class** Prog1 {

**public** **static** **void** main(String[] args) { // main method.

**int** arg = args.length; // If the input is ok, run the program!

System.*out*.println("Number of arguments: " + arg + "\n");

**if** (arg > 1) {

**for** (**int** o = 0; o < arg; o++){

**int** a = Integer.*parseInt*(args[o]);

System.*out*.print(String.*format*("The number %d gives: f1=%d, f2=%d, f4=%d, f8=%d, f16=%d, f32=%d \n",a, *f1*(a),*f2*(a),*f4*(a),*f8*(a),*f16*(a),*f32*(a)));

// Prints the requested numbers of times each function executes.

}

} **else** **if** (arg == 1) {

**int** a = Integer.*parseInt*(args[0]);

System.*out*.print(String.*format*("The number %d gives: f1=%d, f2=%d, f4=%d, f8=%d, f16=%d, f32=%d",a, *f1*(a),*f2*(a),*f4*(a),*f8*(a),*f16*(a),*f32*(a)));

// Prints the requested numbers of times each function executes.

} **else** {

System.*out*.println("There are no arguments!");

}

System.*out*.println("\n");

}

**public** **static** **int** f1(**int** a) {

/\*

\* This method holds the mathematical calculatioin forms.

\*/

**if** (a == 1) {

**return** a;

} **else** **if** (a % 2 == 0) {

**return** a/2;

} **else** {

**return** (3 \* a) +1;

}

}

/\*

\* All methods below gives the value after a certain sets of runs.

\*/

**public** **static** **int** f2(**int** a) {

**return** *f1*(*f1*(a));

}

**public** **static** **int** f4(**int** a) {

**return** *f2*(*f2*(a));

}

**public** **static** **int** f8(**int** a) {

**return** *f4*(*f4*(a));

}

**public** **static** **int** f16(**int** a) {

**return** *f8*(*f8*(a));

}

**public** **static** **int** f32(**int** a) {

**return** *f16*(*f16*(a));

}

}

Notes:

# Prog2

**package** Uppgifter;

/\*

\* Av: Marcus Lund

\* 911118-1153

\* 2015-01-23

\*/

**public** **class** Prog2 {

**public** **static** **void** main(String[] args) {

/\*

\* The main method checks how many arguments the program gets.

\* It loops through a set of two input datas and uses them.

\*/

**int** t = args.length;

**if** (args.length % 2 != 0) {

*message*();

} **else** {

**for** (**int** o = 0; o < t; o +=2) {

**int** p = Integer.*parseInt*(args[o]);

**int** q = Integer.*parseInt*(args[o+1]);

*iterate\_f*(p,q);

}

}

}

**public** **static** **void** iterate\_f(**int** a, **int** b) {

/\*

\* This method prints what value a start value reaches when going througt f1 a defined

numver of times.

\*/

**int** g = a;

**for** (**int** d = 0; d < b; d++) {

a = Prog1.*f1*(a);

}

System.*out*.println(String.*format*("The number %d gives the number %d when looping %d times.", g, a, b));

}

**public** **static** **void** message() {

System.*out*.println("The data input is uneven!");

}

}

Notes:

# Prog3

**package** Uppgifter;

/\*

\* Av: Marcus Lund

\* 911118-1153

\* 2015-01-23

\*/

**public** **class** Prog3 {

**public** **static** **void** main(String[] args) {

**int** arg = args.length; // If the input is ok, run the program!

*arg*(arg);

**if** (arg >= 1) {

**for** (**int** o = 0; o < arg; o++){

*iter\_life\_length*(Integer.*parseInt*(args[o]));

*rec\_print*(Integer.*parseInt*(args[o]));

}

} **else** {

*no\_input*();

}

}

**public** **static** **void** iter\_life\_length(**int** a) {

/\*

\* This method prints the life length of a value.

\*/

**int** counter = 0;

**int** u = a;

**while** (a > 1) {

counter += 1;

a = Prog1.*f1*(a);

}

System.*out*.print(String.*format*("The life length of %d is %d.\n", u, counter));

}

**public** **static** **int** rec\_life\_length(**int** a) {

/\*

\* This recursive method returnes the life length of a value.

\*/

**if** ( a == 1) {

**return** 0;

} **else** {

**return** 1 + *rec\_life\_length*(Prog1.*f1*(a));

}

}

**public** **static** **void** rec\_print(**int** a){

/\*

\* Prints the life length of the recursive method.

\*/

System.*out*.println(String.*format*("The life length of %d is %d.\n", a, *rec\_life\_length*(a)));

}

**public** **static** **void** no\_input() {

/\*

\* If String[] args is empty, print a message.

\*/

System.*out*.println("There are no input!");

}

**public** **static** **void** arg(**int** arg) {

System.*out*.println("Number of arguments: " + arg + "\n");

}

}

Notes:

# Prog4

**package** Uppgifter;

/\*

\* Av: Marcus Lund

\* 911118-1153

\* 2015-01-23

\*/

**public** **class** Prog4 {

**public** **static** **void** main(String[] args) {

/\*

\* The main method checks how many arguments the program gets.

\* It loops through a set of two input data and uses them.

\*/

**int** t = args.length;

**if** (args.length % 2 != 0) {

System.*out*.println("The data input is uneven!");

} **else** {

**for** (**int** o = 0; o < t; o +=2) {

**double** p = Double.*parseDouble*((args[o]));

**int** q = Integer.*parseInt*(args[o+1]);

System.*out*.println(*rec\_raise\_eff*(p, q));

}

}

}

**public** **static** **double** rec\_raise\_eff(**double** a, **int** b) {

/\*

\* A recursive function that calculates the result of a of the power to b.

\*/

**if** (b == 0) {

**return** 1;

} **else** **if** (b % 2 == 0) {

**double** i = *rec\_raise\_eff*(a, (**int**)Math.*floor*((b/2)));

**return** i \* i;

} **else** {

**double** i = *rec\_raise\_eff*(a, (**int**)Math.*floor*((b/2)));

**return** a \* i \* i;

}

}

}

Notes:

# Prog5

**package** Uppgifter;

/\*

\* Av: Marcus Lund

\* 911118-1153

\* 2015-01-23

\*/

**public** **class** Prog5 {

**public** **static** **void** main(String[] args) {

/\*

\* The main method checks how many arguments the program gets.

\* It loops through a set of two input datas and uses them.

\*/

**for** (**int** q = 1; q <= 50; q += 1) {

**double** p = Double.*parseDouble*((args[0]));

System.*out*.println("Rec\_raise\_eff: " + p + " to the power of " + q + " is " + *rec\_raise\_eff*(p, q) + ". It takes " + *counter* + " laps.");

// Calls and prints the result of rec\_raise\_eff(); function

*counter* = 0;

System.*out*.println("Rec\_raise: " + p + " to the power of " + q + " is " + *rec\_raise*(p, q) + ". It takes " + *counter* + " laps. \n");

// Calls and prints the result of rec\_raise(); function

*counter* = 0;

// Resets the value of the counter.

}

}

**static** **int** *counter* = 0;

**static** **public** **double** rec\_raise(**double** a, **int** b) {

/\*

\* This method is slower than the second one, and takes a long time to run depending on the

value of k.

\*/

**if** (b==0) {

**return** 1.0;

} **else** {

*counter* += 1;

**return** a \* *rec\_raise* (a, b-1);

}

}

**public** **static** **double** rec\_raise\_eff(**double** a, **int** b) {

/\*

\* This method is more efficient than the first one. It checks if the value is even or not

and does calculations depending on the input value.

\*/

**if** (b == 0) {

**return** 1.0;

} **else** **if** (b % 2 == 0) {

*counter* += 1;

**double** i = *rec\_raise\_eff*(a, (**int**)Math.*floor*((b/2)));

**return** i \* i;

} **else** {

*counter* += 1;

**double** i = *rec\_raise\_eff*(a, (**int**)Math.*floor*((b/2)));

**return** a \* i \* i;

}

}

}

Notes: