Question 1:

/\*\*

\* The simple program

\* Author: Adrian Pailler

\* Date: 22/1/21

\* Version: 1.0

\* Description: uses GUI to display user input. Focus on use of methods.

\*/

**package** Methods;

**import** javax.swing.\*;

**public** **class** Q1 {

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

String Title = *Title*(); // see 17-25

String something = JOptionPane.*showInputDialog*(**null**, "Enter something: ",Title,0); // asks for user input

JOptionPane.*showMessageDialog*(**null**, "You entered: "+something,Title, 0); // displays the same user input "something"

}

/\*\*

\* Creates the title for the GUI input dialog

\* no parameters

\* returns the title

\*/

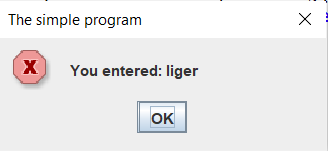
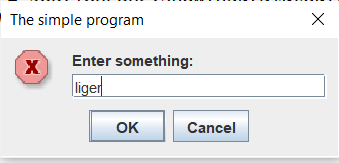
**private** **static** String Title() {

**final** String Title = "The simple program";

**return** Title;

}

}



Question 2:

/\*\*

\* power calculator

\* Author: Adrian Pailler

\* Date: 22/1/21

\* Version: 1.0

\* Description: calculates the power of a base to an exponent even with negative powers

\*/

**package** Methods;

**import** javax.swing.\*;

**public** **class** Q2 {

**static** **final** String ***title*** = "Powers and Exponents"; // since title is reused, creating it as a string is more efficient

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

**int** flag = 0; // flag to efficiently break out

**while**(flag==0) {

JOptionPane.*showMessageDialog*(**null**, "Welcome to Powers and Exponents",***title***,0);

**double** ans = *power*(Double.*parseDouble*(JOptionPane.*showInputDialog*(**null**, "Enter a base: ",***title***,0)),Integer.*parseInt*(JOptionPane.*showInputDialog*(**null**,"Enter an exponent: ",***title***,0)));

// call power (see 27-40) and gets values for base and exponent (in order)

JOptionPane.*showMessageDialog*(**null**, "The answer is: "+ans); // displays the answer

flag =JOptionPane.*showConfirmDialog*(**null**, "whould you like to play again?",***title***,0); // uses flag to decide to repeat program and stay in loop or not

}

**if**(flag==1) {

JOptionPane.*showMessageDialog*(**null**, "Thank you for playing powers and exponents!",***title***,0); // uses flag to initiate code at end

}

}

/\*\*

\* calculates the power

\* two double/integer values (base and exponent) parameters which are set on line 32

\* is recursive therefore returns and calls itself

\*/

**public** **static** **double** power(**double** base, **int** exponent) {

**if**(exponent > 0){ // finds whether exponent is +ve

**return** base\**power*(base, exponent-1); // multiplies base by itself until exponent is 1

}**else** **if**(exponent < 0){ // finds whether exponent is -ve

**return** 1/*power*(base, -exponent); // changes the sign of the exponent and puts it under 1/. after return it goes through 33-34

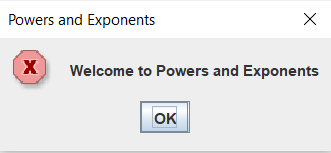
}**else**{ // exponent is 0

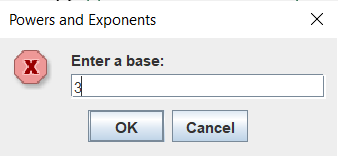
**return** 1;

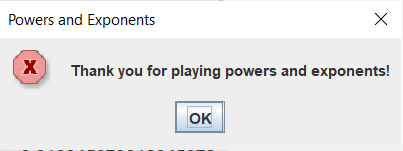
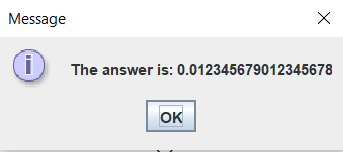
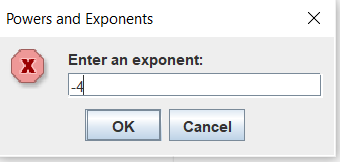
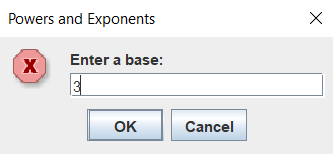
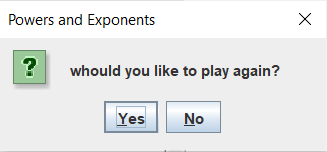
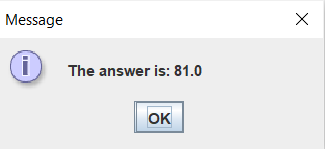
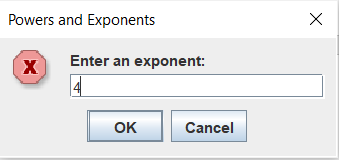
}

}

}







Question 3:

/\*\*

\* Volume calculator

\* Author: Adrian Pailler

\* Date: 23/1/21

\* Version: 1.0

\* Description: Uses user input to calculate the volume of a cylinder

\*/

**package** Methods;

**import** javax.swing.\*;

**public** **class** Q3 {

**static** **final** String ***title*** = "Volume Calculator"; // since title is reused, creating it as a string is more efficient

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

**int** flag = 0;// flag to efficiently break out

**while**(flag==0) {

JOptionPane.*showMessageDialog*(**null**,"Welcome to Volume Calculator",***title***,0);

**double** ans = *input*(); // calls input and receives the answer ()

JOptionPane.*showMessageDialog*(**null**, "The volume is: "+ans);// displays the answer

flag =JOptionPane.*showConfirmDialog*(**null**, "whould you like to play again?",***title***,0); // uses flag to decide to repeat program and stay in loop or not

}**if**(flag==1) {

JOptionPane.*showMessageDialog*(**null**, "Thank you for using volume calculator",***title***,0); // uses flag to initiate code once user does not wish to continue

}

}

/\*\*

\* gets the users inputs

\* no parameters

\* returns double answer

\* calls calculator method (42-50)

\*/

**public** **static** **double** input() {

JTextField one = **new** JTextField();

JTextField two = **new** JTextField();

Object [] constants = {

"radius: ",one,

"height: ",two,

}; // creates two input dialogs

JOptionPane.*showConfirmDialog*(**null**,constants,"Input the radius and the height: ", 0);

**double** radius = Double.*parseDouble*(one.getText());// gets and converts values in the same line

**double** height = Double.*parseDouble*(two.getText());

**double** ans = *calculator*(radius,height); // could be more clearer and direct but just for the sake of using methods

**return** ans; // returns to main

}

/\*\*

\* calculates the volume

\* two doubles (radius and height)

\* returns the answer from the calculation

\*/

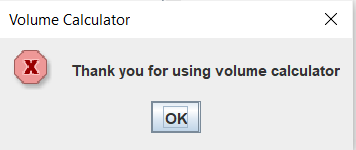
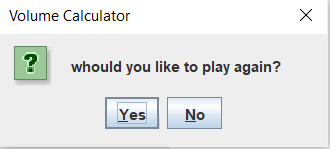
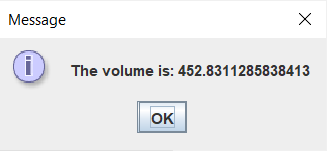
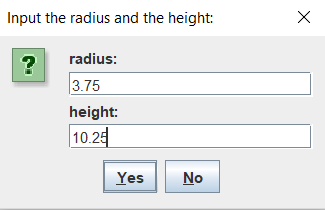
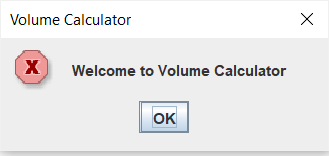
**public** **static** **double** calculator(**double** r, **double** h) {

**double** ans = r\*r\*h\*Math.***PI***; // calculates the volume

**return** ans; // returns to input()

}

}



Question 4:

/\*\*

\* Format with commas

\* Author: Adrian Pailler

\* Date: 23/1/21

\* Version: 1.0

\* Description: Takes an Integer as a parameter (up to 10 digits)

\* and returns a string which is the number with properly places commas

\*/

**package** Methods;

**import** javax.swing.JOptionPane;

**public** **class** Q4 {

**static** StringBuffer *str* = **new** StringBuffer();

**static** **final** String ***title*** = "Formatting Numbers"; // since title is reused, creating it as a string is more efficient

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

**int** flag = 0;// flag to efficiently break out

**while**(flag==0) {

JOptionPane.*showMessageDialog*(**null**,"Welcome to Formatting with Numbers",***title***,0);

StringBuffer ans = *formatwithcommas*(*str*.append(JOptionPane.*showInputDialog*(**null**,"Enter an Integer: "))); // calls method(27-44) and gets user input integer

JOptionPane.*showMessageDialog*(**null**, "The formatted number is: "+ans);// displays the answer

flag =JOptionPane.*showConfirmDialog*(**null**, "whould you like to play again?",***title***,0); // uses flag to decide to repeat program and stay in loop or not

*str*.delete(0, *str*.length());

}**if**(flag==1) {

JOptionPane.*showMessageDialog*(**null**, "Thank you for using number formatter",***title***,0); // uses flag to initiate code once user does not wish to continue

}

}

/\*\*

\* formats the integer with commas. Does this from the end to index 0

\* uses stringBuffer

\* returns the stringBuffer answer

\*/

**public** **static** StringBuffer formatwithcommas(StringBuffer num) {

**int** i = num.length();

**while**(i!=0){

**if** (i-3>0) { // makes sure the index is not below 0

num.insert(i-3, ","); // inserts a comma every 3 indexes

i=i-3;

}**else** { // breaks out of loop

i=0;

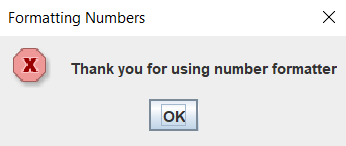
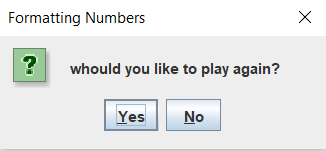
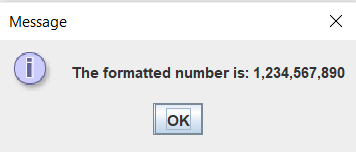
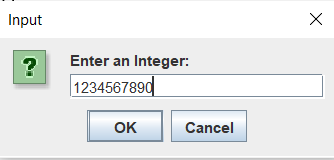
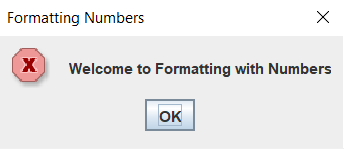
}

}

**return** num;

}

}



Question 6:

/\*\*

\* Fraction Reducer

\* Author: Adrian Pailler

\* Date: 25/1/21

\* Version: 1.0

\* Description: fraction is reduced to numerator and denominator

\*/

**package** Methods;

**import** javax.swing.\*;

**public** **class** Q6 {

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

**final** String title = "Fraction reducer"; // since title is reused, creating it as a string is more efficient

**int** flag = 0; // flag to break efficiently

**while**(flag==0) {

JOptionPane.*showMessageDialog*(**null**,"Welcome to Fraction reducer",title,0);

String input = JOptionPane.*showInputDialog*(**null**,"Enter the fraction: ",title,0);

**if**(input.contains("/")) { // makes sure input is a fraction

**int** denominator = *getDenominator*(input); // calls getdenominator(77-87)

**int** numerator = *getNumerator*(input); // calls getnumerator(65-75)

String reduction = *reduce*(numerator,denominator); // calls reduction (33-63)

JOptionPane.*showMessageDialog*(**null**, "The numerator is: "+numerator+"\n The denominator is: "+denominator+"\n The reduced fraction is: "+reduction,title,0);

} **else** { // if it is not a fraction

JOptionPane.*showMessageDialog*(**null**, "That is not fraction! Your interllect is questionable...",title,0);

}

flag = JOptionPane.*showConfirmDialog*(**null**, "would you like to try again?",title,0); // loops back to 16 player chooses yes

}

**if**(flag==1) {

JOptionPane.*showMessageDialog*(**null**,"Bye!",title,0);

}

}

/\*\*

\* reduces the fraction to its smallest form

\* takes in two int values (numerator and denominator of fraction)

\* returns the answer of the reduction

\*/

**public** **static** String reduce(**int** numerator, **int** denominator) {

**int** largest;

**int** n = numerator, d = denominator;

String ans;

**if** (numerator < 0) {

n = -numerator;

}

**if** (n > d) { // checks if it is a improper fraction

largest = n;

} **else** {

largest = d;

}

**int** gcd = 0;

**for** (**int** i = largest; i >= 2; i--) {

**if** (numerator % i == 0 && denominator % i == 0) {

gcd = i; // gets the greatest common factor

i=0;

}

}

**if** (gcd != 0) { // divides both by the greatest common factor to reduce them

numerator /= gcd;

denominator /= gcd;

}

ans= numerator+"/"+denominator; // puts both back together

**return** ans;

}

/\*\*

\* gets the numerator from the fraction string

\* takes a String for the fraction

\* returns the numerator

\*/

**public** **static** **int** getNumerator(String fraction) {

**int** numerator;

**int** index = fraction.indexOf("/");

numerator = Integer.*parseInt*(fraction.substring(0,index)); // gets the value before the '/'

**return** numerator;

}

/\*\*

\* gets the denominator from the fraction string

\* takes a String for the fraction

\* returns the denominator

\*/

**public** **static** **int** getDenominator(String fraction) {

**int** denominator;

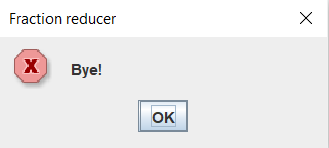
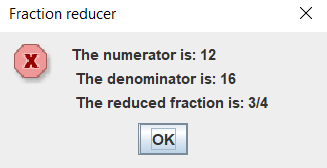
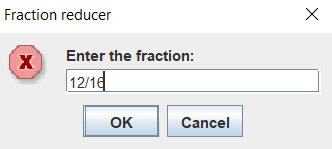
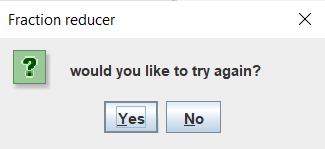
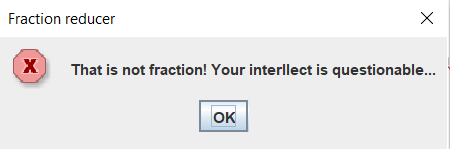
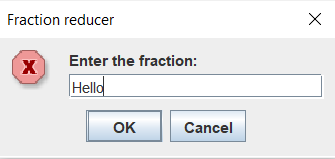
**int** index = fraction.indexOf("/")+1;

denominator = Integer.*parseInt*(fraction.substring(index,fraction.length()));// gets the value after the '/'

**return** denominator;

}

}



Question 7:

/\*\*

\* Drivers education

\* Author: Adrian Pailler

\* Date: 25/1/21

\* Version: 1.0

\* Description: converts time into a standard time format

\*/

**package** Methods;

**import** javax.swing.\*;

**public** **class** Q7 {

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

**final** String title = "Driver's education department"; // since title is reused, creating it as a string is more efficient

**int** flag = 0; // flag to break efficiently

**while**(flag==0) {

JOptionPane.*showMessageDialog*(**null**,"Welcome to driver's education",title,0);

**int**[] arr =*readTime*();

*displayTime*(arr[0],arr[1]);

flag = JOptionPane.*showConfirmDialog*(**null**, "would you like to try again?",title,0); // loops back to 16 player chooses yes

}

**if**(flag==1) {

JOptionPane.*showMessageDialog*(**null**,"Bye!",title,0);

}

}

/\*\*

\* gets the users input minutes and seconds

\* no parameters

\* no returns

\*/

**public** **static** **int**[] readTime() {

SpinnerModel limits = **new** SpinnerNumberModel(1,0,10000000,1);

SpinnerModel limit = **new** SpinnerNumberModel(1,0,10000000,1);// initial, min, max, step

JSpinner minutes = **new** JSpinner(limits); // creates spinner

JSpinner seconds = **new** JSpinner(limit);

Object [] spinners = {

"Enter minutes: ",minutes,

"Enter seconds: ",seconds,

};

JOptionPane.*showConfirmDialog*(**null**,spinners);

**int** min = (**int**) minutes.getValue();// gets and converts values in the same line

**int** sec = (**int**) seconds.getValue();

**int**[] arr = **new** **int**[2];

arr[0]=min; arr[1]=sec; // can only return one object

**return** arr;

}

/\*\*

\* puts the time in the correct format

\* int int for the minutes and seconds

\* returns the answer

\*/

**public** **static** **void** displayTime(**int** min, **int** sec) {

**int** m\_remainder=min;

**int** s\_remainder=sec;

**int** hours=00;

**if**(min>=60) {

hours = min/60;

m\_remainder = min%60; // converts min to hour

}

**if**(sec>=60) {

m\_remainder = sec/60+m\_remainder;

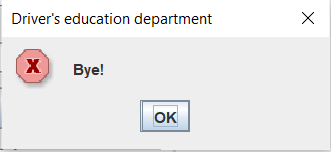
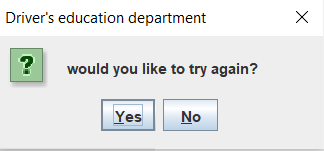
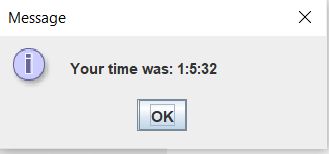
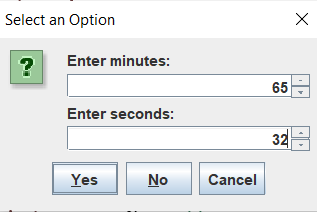
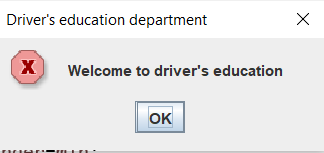
s\_remainder = sec%60; // converts sec to min

}

JOptionPane.*showMessageDialog*(**null**,"Your time was: "+hours+":"+m\_remainder+":"+s\_remainder);

}

}



Question 8:

/\*\*

\* C-RAP

\* Author: Adrian Pailler

\* Date: 26/1/21

\* Version: 1.0

\* Description: does unit conversions for user

\*/

**package** Methods;

**import** javax.swing.\*;

**public** **class** Q8 {

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

**final** String title = "C-RAP converter"; // since title is reused, creating it as a string is more efficient

**int** flag = 0; // flag to break efficiently

**while**(flag==0) {

JOptionPane.*showMessageDialog*(**null**,"Welcome to C-RAP converter",title,0);

**double**[] choice = *Choice*(); // calls choice(25-42)

*Converter*(choice[0],choice[1]); // calls converter(43-89)

flag = JOptionPane.*showConfirmDialog*(**null**, "would you like to try again?",title,0); // loops back to 16 player chooses yes

}

**if**(flag==1) {

JOptionPane.*showMessageDialog*(**null**,"Thank you for using C-RAP conversion",title,0);

}

}

/\*\*

\* gets the users input for what type of conversion they want to use and also the values they need converted

\* no parameters

\* returns the choice of conversion (choice[0]) and the value to be converted (choice[1])

\*/

**public** **static** **double**[] Choice() {

SpinnerModel limit = **new** SpinnerNumberModel(1,0,10000000,0.5);// initial, min, max, step

JSpinner input = **new** JSpinner(limit); // creates spinner

**double**[] choice= **new** **double**[2];

String [] buttons = { // creates the user input buttons

"1- Inches to Centimeters","2- Feet to Centimeters","3- Yards to Meters","4- Miles to Kilometers","5- Centimeters to Inches",

"6- Centimeters to Feet","7- Meters to Yards","8- Kilometers to Miles"};

choice[0] = JOptionPane.*showOptionDialog*(**null**,"Click a button","What would you like to convert",JOptionPane.***DEFAULT\_OPTION***,JOptionPane.***INFORMATION\_MESSAGE***,**null**,buttons,buttons[0]);

JOptionPane.*showConfirmDialog*(**null**,input);

choice[1] = (**double**) input.getValue(); // cannot return 1 value hence the array

**return** choice;

}

/\*\*

\* Converts the input to whichever unit the user chose out of the 8 options

\* double double, the first being the type of conversion and the second is the amount of the value being converted all retrieved from choice()

\* no returns

\*/

**public** **static** **void** Converter(**double** choice, **double** amount) {

String initial; // both initial and last are used to display the name of the unit based on the type of conversion

String last;

**double** ans;

**if**(choice==0) { // finds the choice option the user picked

initial = "Inches";

last = "Centimeters";

ans=amount\*2.54;

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

initial = "Feet";

last = "Centimeters";

ans=amount\*30.48;

}**else** **if**(choice==2) {

initial = "Yards";

last = "Meters";

ans=amount/1.0936133;

}**else** **if**(choice==3) {

initial = "Miles";

last = "Kilometers";

ans=amount\*1.609344;

}**else** **if**(choice==4) {

initial = "Centimeters";

last = "Inches";

ans=amount/2.54;

}**else** **if**(choice==5) {

initial = "Centimeters";

last = "Feet";

ans=amount/30.48;

}**else** **if**(choice==6) {

initial = "Meters";

last = "Yards";

ans=amount\*1.0936133;

}**else** {

initial = "Kilometers";

last = "Miles";

ans=amount/1.609344;

}

JOptionPane.*showMessageDialog*(**null**,amount+" "+initial+" Is equal to: "+ans+" "+last); /\* amount: value of unit to be converted

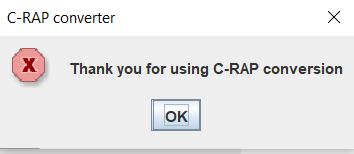
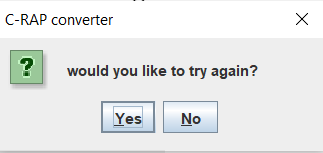
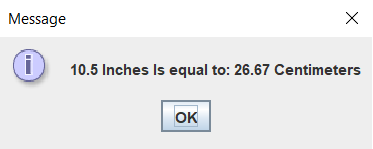
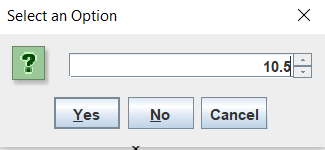
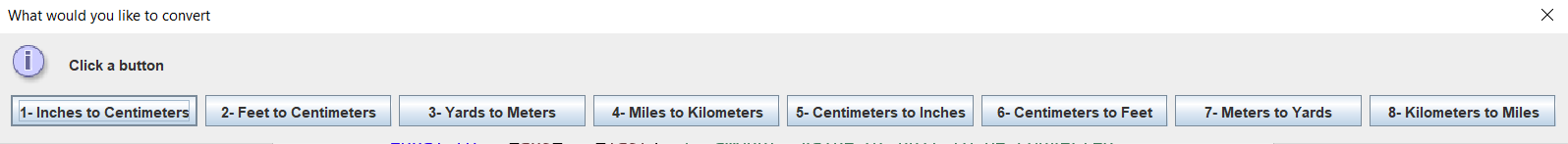
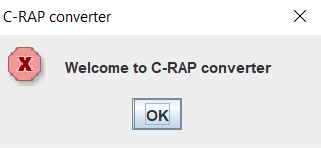
initial: name of unit to be converted

ans: value of unit after conversion

last: name of unit after conversion \*/

}

}



Question 14:

/\*\*

\* Mr. Teasdale's Rövarspråket (Robbers language)

\* Author: Adrian Pailler

\* Date: 28/1/21

\* Version: 1.0

\* Description: Converts normal English into robbers language

\* Rules: 1. if letter is consonant leave it

\* 2. get the closest vowel from consonant and put it in front of it

\* 3. get next consonant of the original consonant

\* 4. if the letter is a vowel, leave it

\*/

**package** Methods;

**import** javax.swing.\*;

**public** **class** Q14 {

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

**final** String title = "Rövarspråket"; // since title is reused, creating it as a string is more efficient

**int** flag = 0; // flag to break efficiently

**while**(flag==0) {

JOptionPane.*showMessageDialog*(**null**,"Welcome to Rövarspråket",title,0);

String input = *input*();

*toRövarspråket*(input);

flag = JOptionPane.*showConfirmDialog*(**null**, "would you like to try again?",title,0); // loops back to 16 player chooses yes

}

**if**(flag==1) {

JOptionPane.*showMessageDialog*(**null**,"Thank you for using Rövarspråket",title,0);

}

}

/\*\*

\* gets the users input

\* no parameters

\* returns the word to be translated (String)

\*/

**public** **static** String input() {

String english = JOptionPane.*showInputDialog*(**null**,"Enter a word to translate: ");

**if**(english.matches("[a-zA-Z]+")) { // makes sure input is a word

**return** english.toLowerCase();

}**else** {

JOptionPane.*showMessageDialog*(**null**, "That is not a word...");

**return** **null**;

}

}

/\*\*

\* translates the English word to Rövarspråket

\* String (English) takes the English input of the user to translate

\* no return

\*/

**public** **static** **void** toRövarspråket(String english) {

String consonants = "bcdfghjklmnpqrstvwxyz";

String collector = ""; // collects the individual chars of the translated word

**char** a='a'; // gets the nearest vowel

**int** index = 0; // gets index of next consonant

**for**(**int** i=0;i<english.length();i++) {

**if**(english.substring(i,i+1).matches("[^aeiou]")) { // checks if the letter at index i is a consonant

**if**(english.charAt(i)<='c') { // if it isn't a consonant || 55-65 gets the nearest vowel

a='a';

} **else** **if**(english.charAt(i)<='g') {// finds the closest vowel

a='e';

} **else** **if**(english.charAt(i)<='l') {//

a='i';

} **else** **if**(english.charAt(i)<='r') {//

a='o';

} **else** **if**(english.charAt(i)<='z') {//

a='u';

} // 66-70 gets next consonant

**if**(english.charAt(i)=='z') { // if consonant is 'z', get 'b' the first consonant in the alphabet

index=consonants.indexOf('a');

} **else** { // if it is not 'z'

index = consonants.indexOf(english.charAt(i));

}

collector = collector+english.charAt(i);// adds the orginal consonant of the string

collector = collector+a;// adds the nearest vowel after

collector = collector+consonants.charAt(index+1);// adds the next consonant last

} **else** {// if it is a consonant, leaves it

collector = collector+english.charAt(i);

}

}

JOptionPane.*showMessageDialog*(**null**, "The word "+english+" in Rövarspråket is "+collector);

}

}

