StringBuilder

Dateioperation

Dateien lesen und schreiben: import java.io.[File,InputStream,InputStreamReader,IOException,BufferedReader]; import java.nio.file.[Files,Paths]; String datei="Dateiname"; BufferedReader eingang=new BufferedReader(new InputStreamReader(new FileInputStream(datei))); while((zeile=eingang.readLine())!=null){inhalt.append(zeile);inhalt.append(System.lineSeparator());} String inhaltAlsString=inhalt.toString(); c.replace(...,...)); Files.write(Paths.get("Dateiname"),inhaltAlsString.getBytes()); alt: Scanner scanner=new Scanner(new File(datei),"UTF-8"); while(scanner.hasNext()){wortholen=scanner.next(); wortholen=wortholen.replaceAll("...",""); import java.util.date; datei=new File("..."); Date d=new Date(); d.setTime(datei.lastModified()); d.toString .isDirectory .exists .length .getAbsolutePath .canRead

Kopieren

Kopieren: Path q=Paths.get(String); Path z=Paths.get(String); Files.copy(q,z); File datei=new File(verz); datei.mkdir;

fx-Fenster

import javafx.application.Application; geometry.Insets; stage.Stage; scene.Scene; scene.control.Button; scene.layout.GridPane; scene.text.Text;] public class View extends Application{Text x=new Text(); public void start(Stage primaryStage){try{GridPane root=new GridPane(); root.setHgap(10); root.setVgap(10); root.setPadding(new Insets(25,25,25,25)); Button b=new Button(String); root.add(b,0,0); x.setText(String); root.add(x,1,0); b.setOnAction(e->methode()); Scene scene=new Scene(root,200,150); primaryStage.setScene(scene); primaryStage.show();}catch(Exception e){e.printStackTrace();}} public static void main(String[]args){launch(args);}}

Label

javafx.scene.control.[Label,TextField] javafx.scene.image.[Image,ImageView] public Image; public ImageView; image=new Image(getClass().getResource("/.../...jpg").toString()); imageview=new ImageView(image);

rechteck

public void start(Stage primaryStage){try{Group root=new Group(); for(MShape s:shapes){if(s instanceof MRectangle){MRectangle srec=(MRectangle)s; Rectangle rec=new Rectangle(srec.getXOrigin(),srec.getYOrigin(),2*srec.getXDelta(),2*srec.getYDelta()); rec.setFill(Color.rgb(255,0,0,0.15)); root.getChildren().addAll(rec,text); Scene scene=new Scene(root,400,400);

rechteck

public class MRectangle extends MShape{private double xDelta;...yDelta; public MRectangle(double xOrigin,d'yOrigin d xDelta,d'yDelta){super(xOrigin,yOrigin); this.xDelta=xDelta; this.yDelta=yDelta;

rekursion

public class Adder{public static void main(String[]args){int eingabe=10; System.out.println("rekursiv"+rek(eingabe)); System.out.println("iterativ"+it(eingabe));} public static int rek(int eingabe){if(eingabe==1){return 1;}else{return eingabe+rek(eingabe-1);}} public static int it(int eingabe){int i=0; int e=0; while(i<=eingabe){e=e+i; i++;}return e;}}

Typen

int result=Integer.parseInt(String) DoubleparseDouble. Double.toString/Integer.toString

Sax

javax.xml.parsers.[ParserConfigurationException,SAXParser,SAXParserFactory]; org.xml.sax.[Attributes,SAXException,helpers.DefaultHandler] public class SAXAusgabe{private static String XMLDateiName="...xml"; public static void main(String[]args){File eineDatei=new File(XMLDateiName); MyDefaultHandler handler=new MyDefaultHandler(); try{SAXParserFactory factory=SAXParserFactory.newInstance(); SAXParser parser=factory.newSAXParser(); parser.parse(eineDatei,handler);}catch(SAXException e,IOException e,ParserConfigurationException e){e.printStackTrace();}} private static class MyDefaultHandler extends DefaultHandler{public void startDocument(){System.out.println("Start");} public void endDocument(){System.out.println("Ende");} public void startElement(String namespaceURI,String localName,String qualifiedName,Attributes attr){String elementName=localName.equals("")?qualifiedName:localName; if(elementName.equals("Vorname")||elementName.equals("Nachname")||elementName.equals("Strasse")){printData=true;}