

repo_report5					
repo_report					
s_line	code	Check leaky	file	leak resource	repository
81	<pre> public static void initDB() { JDBCConnection conn = JDBCFactory.takeJDBC(); conn.submit(new Transaction() { @Override public boolean trying() { try { for(String sql : ddl) { conn.update(sql); } } catch (SQLException e) { e.printStackTrace(); return false; } return true; } }); } </pre>	yes	dogfightZ/src/dogfight_Z/doglog/dao/DBInit.java	JDBCConnection	https://github.com/PointRider/open-java
95	<pre> @Override public void receive(TransportClient client, ByteBuffer message, RpcResponseCallback callback) { try { semaphore.acquire(); callback.onSuccess(ByteBuffer.allocate(responseSize)); } catch (InterruptedException e) { // do nothing } } </pre>	no	network-common/src/test/java/com/bytedance/css/network/Req	semaphore	http://github.com/bytedance/CloudShuffleService
7	<pre> public static void main(String[] args) throws Exception{ URLClassLoader urlClassLoader = new URLClassLoader(new URL[]{}); Class calc = urlClassLoader.loadClass("Calc"); calc.newInstance(); } </pre>	yes	JavaSecurity/Reappearance/Serializable/src/ClassLoader/URLClassLoader	URLClassLoader	https://github.com/Drun1baby/JavaSecurityLearning
6	<pre> public static void main(String[] args) { // TODO Auto-generated method stub Scanner scn = new Scanner(System.in); int n = scn.nextInt(); int fn = fib(n); // // int[] strg = new int[n + 1]; // int fn = fibm(n, strg); // int fn = fib(n); // System.out.println(fn); // } </pre>	yes	Aug23/Codes/dp/Fib.java	Scanner	https://github.com/sumet-malik/level2and3
255	<pre> public void doGet(HttpServletRequest request, HttpServletResponse response) throws response.setContentType("text/html"); // Hello PrintWriter out = response.getWriter(); out.println("<html><body>"); out.println("<h1>" + message + "</h1>"); out.println("</body></html>"); } </pre>	yes	JavaSecurity/EL/BasicELVu/src/main/java/drun1baby/basicel/vuln/	PrintWriter	https://github.com/Drun1baby/JavaSecurityLearning
37	<pre> public static void main(String args[]) { Scanner in = new Scanner(System.in); // base_x,base_y: The corner of the map representing your base int baseX = in.nextInt(); int baseY = in.nextInt(); // heroesPerPlayer: Always 3 int heroesPerPlayer = in.nextInt(); // game loop while (true) { int myHealth = in.nextInt(); // Your base health int myMana = in.nextInt(); // Ignore in the first league: Spend ten mana to cast int oppHealth = in.nextInt(); int oppMana = in.nextInt(); int entityCount = in.nextInt(); // Amount of heroes and monsters you can see List<Entity> myHeroes = new ArrayList<>(entityCount); List<Entity> oppHeroes = new ArrayList<>(entityCount); List<Entity> monsters = new ArrayList<>(entityCount); for (int i = 0; i < entityCount; i++) { int id = in.nextInt(); // Unique identifier int type = in.nextInt(); // 0=monster, 1=your hero, 2=opponent hero int x = in.nextInt(); // Position of this entity int y = in.nextInt(); int shieldLife = in.nextInt(); // Ignore for this league: Count down until shield is controlled = in.nextInt(); // Ignore for this league: Equals 1 when the int health = in.nextInt(); // Remaining health of this monster int vx = in.nextInt(); // Trajectory of this monster int vy = in.nextInt(); int nearBase = in.nextInt(); // 0=monster with no target yet, 1=monster int threatFor = in.nextInt(); // Given this monster's trajectory, is it a threat Entity entity = new Entity(id, type, x, y, shieldLife, isControlled, health, vx, vy, nearBase, threatFor); switch (type) { case TYPE_MONSTER: // monster break; } } } } </pre>	yes	starterAIs/Starter.java	Scanner	https://github.com/CodinGame/SpringChallenge2022
48	<pre> @RequestMapping("/jdbc/vuln") public String jdbc_sql_vuln(@RequestParam("username") String username) { StringBuilder result = new StringBuilder(); try { Class.forName(driver); Connection con = DriverManager.getConnection(url, user, password); if (!con.isClosed()) System.out.println("Connect to database successfully."); // sql vuln code Statement statement = con.createStatement(); String sql = "select * from users where username = '" + username + "'"; logger.info(sql); ResultSet rs = statement.executeQuery(sql); while (rs.next()) { String res_name = rs.getString("username"); String res_pwd = rs.getString("password"); String info = String.format("%s: %s\n", res_name, res_pwd); result.append(info); logger.info(info); } rs.close(); con.close(); } catch (ClassNotFoundException e) { logger.error("Sorry,can't find the Driver!"); } catch (SQLException e) { logger.error(e.toString()); } return result.toString(); } </pre>	yes	JavaSecurity/CodeReview/JavaSec-Code/MybatisSql/src/main/java/	Statement	https://github.com/Drun1baby/JavaSecurityLearning
92	<pre> @RequestMapping("/jdbc/sec") public String jdbc_sql_sec(@RequestParam("username") String username) { StringBuilder result = new StringBuilder(); try { Class.forName(driver); Connection con = DriverManager.getConnection(url, user, password); if (!con.isClosed()) System.out.println("Connecting to Database successfully."); // fix code String sql = "select * from users where username = ?"; PreparedStatement st = con.prepareStatement(sql); st.setString(1, username); logger.info(st.toString()); // sql after prepare statement ResultSet rs = st.executeQuery(); while (rs.next()) { String res_name = rs.getString("username"); String res_pwd = rs.getString("password"); String info = String.format("%s: %s\n", res_name, res_pwd); result.append(info); logger.info(info); } rs.close(); con.close(); } catch (ClassNotFoundException e) { logger.error("Sorry, can't find the Driver!"); e.printStackTrace(); } catch (SQLException e) { logger.error(e.toString()); } return result.toString(); } </pre>	yes	JavaSecurity/CodeReview/JavaSec-Code/MybatisSql/src/main/java/	PreparedStatement	https://github.com/Drun1baby/JavaSecurityLearning

40	<pre>public static void main(String args[]) { Scanner in = new Scanner(System.in); int baseX = in.nextInt(); int baseY = in.nextInt(); int heroesPerPlayer = in.nextInt(); int[] posX = baseX == 0 ? new int[] { 3000, 7000, 5500 } : new int[] { baseX - 3 int[] posY = baseY == 0 ? new int[] { 6500, 1500, 4000 } : new int[] { baseY - 6 // game loop while (true) { int myHealth = 0; int mana = 0; myHealth = in.nextInt(); mana = in.nextInt(); in.nextInt(); in.nextInt(); int entityCount = in.nextInt(); Unit[] units = new Unit[entityCount]; for (int i = 0; i < entityCount; i++) { int id = in.nextInt(); int type = in.nextInt(); int x = in.nextInt(); int y = in.nextInt(); int shieldLife = in.nextInt(); int isControlled = in.nextInt(); int health = in.nextInt(); int vx = in.nextInt(); int vy = in.nextInt(); int nearBase = in.nextInt(); int threatFor = in.nextInt(); units[i] = new Unit(id, type, x, y, shieldLife, isControlled, health, vx, vy, ne } Unit[] myHeroes = Arrays.stream(units).filter(unit -> unit.type == 1).toArray int nbHeroesRoaming = 2; Unit closestEnemyToBase = null; double minDistToBase = Double.POSITIVE_INFINITY; for (Unit unit : units) { if (unit.type != 0) continue; double curDist = computeDist(baseX, baseY, unit.x, unit.y); if (curDist < minDistToBase) { minDistToBase = curDist; closestEnemyToBase = unit; } } for (int i = 0; i < nbHeroesRoaming; i++) { Unit hero = myHeroes[i]; Unit target = null; double minDist = Double.POSITIVE_INFINITY; for (Unit unit : units) { if (unit.type != 0 unit == closestEnemyToBase) continue; double curDist = computeDist(hero.x, hero.y, unit.x, unit.y); if (curDist < minDist) { minDist = curDist; target = unit; } } if (target == null) { System.out.println("MOVE " + posX[i] + " " + posY[i]); } else { if ((canWind(mana) && computeDist(hero.x, hero.y, target.x, target.y) < 1 System.out.println("SPELL WIND " + (1/330-baseX) + " " + (9000-ba) else { System.out.println("MOVE " + target.x + " " + target.y); } } } for (int i = nbHeroesRoaming; i < heroesPerPlayer; i++) { Unit hero = myHeroes[i]; Unit target = closestEnemyToBase; if (target == null minDistToBase > 5000) { System.out.println("WAIT"); } else { if ((canWind(mana) && computeDist(hero.x, hero.y, target.x, target.y) < 1 System.out.println("SPELL WIND " + posX[i] + " " + posY[i]);) else { System.out.println("MOVE " + target.x + " " + target.y); } } } } }</pre>	yes	config/level2/Boss.java	Scanner in	https://github.com/CodinGame/SpringChallenge2022
53	<pre>public static String httpClient(String url) { StringBuilder result = new StringBuilder(); try { CloseableHttpClient client = HttpClients.createDefault(); HttpGet httpGet = new HttpGet(url); // set redirect enable false // httpGet.setConfig(RequestConfig.custom().setRedirectsEna HttpResponse httpResponse = client.execute(httpGet); // send BufferedReader rd = new BufferedReader(new InputStreamReader String line; while ((line = rd.readLine()) != null) { result.append(line); } return result.toString(); } catch (Exception e) { return e.getMessage(); } }</pre>	yes	rc/main/java/com/drunkbaby/util/HttpUtils.java	CloseableHttpClient , BufferedReader	https://github.com/Drun1baby/JavaSecurityLearning
211	<pre>public static void main (String[] args) { String url = "http://127.0.0.1:7777/#JndiCalc"; int port = 1234; try { InMemoryDirectoryServerConfig config = new InMemoryDirectoryServerCon config.setListenerConfigs(new InMemoryListenerConfig[] "listen", InetAddress.getByAddress("0.0.0.0"), port, ServerSocketFactory.getDefault(), SocketFactory.getDefault(), (SSLContextFactory) SSLContextFactory.getDefault()); config.addInMemoryOperationInterceptor(new OperationInterceptor(new UI InMemoryDirectoryServer ds = new InMemoryDirectoryServer(config); System.out.println("Listening on 0.0.0.0:" + port); ds.startListening(); } catch (Exception e) { e.printStackTrace(); } finally { ds.shutdown(true); } }</pre>	yes	JavaSecurity/ndi/UndiRMIServer/src/main/java/JNDIGe	InMemoryDirectoryServer	https://github.com/Drun1baby/JavaSecurityLearning
7	<pre>public static boolean telnet(String host, int port) { Socket socket = new Socket(); boolean isConnected = false; long str = System.currentTimeMillis(); try { socket.connect(new InetSocketAddress(host, port), 1000); isConnected = socket.isConnected(); System.out.println(isConnected); } catch (IOException e) { log.error("can not telnet {}:{}", host, port); } finally { try { try { socket.close(); } catch (IOException e) { log.error("can not telnet {}:{}", host, port); } } } System.out.println(System.currentTimeMillis() - str); return isConnected; }</pre>	yes	manager/dm-server/src/main/java/org/apache/doris/stack/util/Tel	Socket	https://github.com/apache/doris-manager

153	<pre>public static void main(String[] args) { Object mutex = new Object(); Thread t1 = new Thread(new Runnable() { @Override public void run() { JDBCConnection[] arr = new JDBCConnection[16]; synchronized (mutex) { for(int i = 0; i < 16; ++i) { //Thread.sleep(1000); arr[i] = JDBCFactory.takeJDBC(); System.out.println("t1 taken " + i); } } for(int i = 0; i < 14; ++i) { JDBCFactory.returnBack(arr[i]); System.out.println("t1 returned " + i); } try { Thread.sleep(2000); } catch (InterruptedException e) { e.printStackTrace(); } for(int i = 14; i < 16; ++i) { JDBCFactory.returnBack(arr[i]); } System.out.println("t1 returned all"); } }); Thread t2 = new Thread(new Runnable() { @Override public void run() { JDBCConnection[] arr = new JDBCConnection[16]; synchronized (mutex) { for(int i = 0; i < 16; ++i) { arr[i] = JDBCFactory.takeJDBC(); System.out.println("t2 taken " + i); } } for(int i = 0; i < 14; ++i) { JDBCFactory.returnBack(arr[i]); System.out.println("t2 returned " + i); } try { Thread.sleep(10000); } catch (InterruptedException e) { e.printStackTrace(); } for(int i = 14; i < 16; ++i) { JDBCFactory.returnBack(arr[i]); } System.out.println("t2 returned all"); } }); t1.start(); t2.start(); try { t1.join(); t2.join(); } catch (InterruptedException e) { e.printStackTrace(); } }</pre>	no	open-java/dogfightZ/src/dogfight_Z/dogLog/utlis	arr[i]	https://github.com/PointRider/open-java
38	<pre>public static ArrayList<Trajectory> readOriginalTrajectoriesFromGeolife() { ArrayList<Trajectory> trajectoryList = new ArrayList<Trajectory>(); //Initialize a dynamic array of type Trajectory try { // open files from folder File diretory = new File(INPUT_FILE + ORIGINAL_TRA_FOLDER); File files[] = openDirectoryFiles(diretory); // read files for (int field = 0; field < files.length; field++) { File currentFile = files[field]; // read file BufferedReader buffer = new BufferedReader(new FileReader(currentFile)); // fields to be read from the file double coordinate[] = new double[DIMENSION]; // new trajectory for this file, set features Trajectory trajectory = new Trajectory(); // read file lines while (buffer.ready()) { String line = buffer.readLine(); String[] tokens = line.split(" "); // if new trajectory if (tokens.length != 7) { // Add trajectories with more than 20 points only continue; } else { // Parse the inputs String timeString = tokens[5] + " " + tokens[6]; coordinate[1] = Double.parseDouble(tokens[0]); coordinate[0] = Double.parseDouble(tokens[1]); SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss"); Date date = null; try { date = format.parse(timeString); } catch (ParseException e) { e.printStackTrace(); } // Convert time to timestamp long timeStamp = date.getTime(); // create a new point from the line input, set features Point point = new Point(coordinate, timeStamp); trajectory.addPoint(point); } } // Add trajectories with more than 20 points only if (trajectory.numberOfPoints() >= 20 && trajectory.numberOfPoints() <= 1000) trajectoryList.add(trajectory); if (trajectoryList.size() == 5000) break; } // close file buffer.close(); } catch (IOException e) { System.out.println("Error opening input files."); e.printStackTrace(); } return trajectoryList; }</pre>	yes	JDBCFactory.java	BufferedReader	https://github.com/leoshuncheng/vldb-trajectory-distance-measure
34	<pre>@Override protected Result translate(String query, String t) throws Exception { AtomicReference<Exception> exception = new AtomicReference<>(); AtomicReference<String> detectedLanguage = new AtomicReference<>(); final CountdownLatch waitDetect = new CountdownLatch(1); if (LanguageDetector.hasSupport()) { LanguageDetector.detectLanguage(query, lng -> { if (!Objects.equals(lng, "und")) { detectedLanguage.set(lng); } waitDetect.countDown(); }); exception.set(e); waitDetect.countDown(); }; waitDetect.await(); if (exception.get() != null) { throw exception.get(); } } ArrayList<String> blocks = getStringBlocks(query, 2500); StringBuilder resultString = new StringBuilder(); for (String block : blocks) { final CountdownLatch waitTranslate = new CountdownLatch(1); AtomicReference<String> translated = new AtomicReference<>(); TLRPC.TL_messages_translateText req = new TLRPC.TL_messages_translateText(); req.flags = 2; req.to_lang = t; req.text = block.replace("\n", "
"); ConnectionsManager.getInstance().userConfig.selectedAccount().sendRequest(req); if (res instanceof TLRPC.TL_messages_translateResultText) { TLRPC.TL_messages_translateResultText result = (TLRPC.TL_messages_translateResultText) res; translated.set(result.text); } else if (err != null) { exception.set(new Exception(err.text)); } else { exception.set(new Exception("Unknown error")); } waitTranslate.countDown(); }; waitTranslate.await(); if (exception.get() != null) { throw exception.get(); } resultString.append(buildTranslatedString(block, translated.get()).replace("
", "\n")); } return new Result(resultString.toString(), detectedLanguage.get()); }</pre>	no	TMessagesProj/src/main/java/uz/unnams/cherrygram/translator/Translator.java	dict_keys['LanguageDetector.de']	https://github.com/arsLan4k1390/Cherrygram
22	<pre>public static void main (String[] args) { String url = "http://127.0.0.1:7777/#JndiCalc"; int port = 1234; try { InMemoryDirectoryServerConfig config = new InMemoryDirectoryServerConfig("listen", InetAddress.getByAddress("0.0.0.0"), port, ServerSocketFactory.getDefault(), SocketFactory.getDefault(), (SSLContext.getDefault().getSocketFactory())); config.addInMemoryOperationInterceptor(new OperationInterceptor(new UI InMemoryDirectoryServer ds = new InMemoryDirectoryServer(config); System.out.println("Listening on 0.0.0.0:" + port); ds.startListening(); } catch (Exception e) { e.printStackTrace(); } }</pre>	No	JavaSecurity/JndiCode/UndiRMServer/src/main/java/UNDIGeometry/Server.java	some	https://github.com/Drun1baby/JavaSecurityLearning

pull request	
https://github.com/PointRider/open-java/pull/2/commits/60112e8a007b1fd575137423686fab82bec838	
https://github.com/Drun1baby/JavaSecurityLearning/pull/3/commits/fdce70c7848a8919dcdb893b55f5fb9a28e24741	
https://github.com/suneet-malik/level2and3/pull/3/commits/ec1c5b365f70116b2562853285bb3d5876eb4fe4	
https://github.com/Drun1baby/JavaSecurityLearning/pull/3/commits/7726285850e5e7d2670738fd1b0ec07a4a92b19c	
https://github.com/CodinGame/SpringChallenge2022/pull/29/commits/3a4fddfdccb7d15464b81620a72c64bf4b5b5908a0	
https://github.com/Drun1baby/JavaSecurityLearning/pull/4/commits/866739eed18c74c7d0d4daac83f6650c0fddeda8d	
https://github.com/Drun1baby/JavaSecurityLearning/pull/4/commits/a86f062d9ce68069c3e21f9ec043352ef671167e	

https://github.com/CodinGame/SpringChallenge2022/commit/3e9ed49e82737574d94451673c5f932f67f41faa	
https://github.com/Drun1baby/JavaSecurityLearning/pull/4/commits/69f231313247bb61fe544b8863b6da24ea95b23e	
https://github.com/Drun1baby/JavaSecurityLearning/pull/3/commits/04d35eea5d683f302af9819f66d9fa5f87eb23	
https://github.com/apache/doris-manager/pull/69/commits/90692505ab5775b9fb1c37c299e58d534523b781	

https://github.com/leoshuncheng/vldbj-trajectory-distance-measures/pull/1/commits/6d3339a4e3c58e0299a0d77d97ee198c2ea7b3cb9	