



Politechnika Wrocławska

Metryki oprogramowania

Marian Jureczko

Plan wykładu

- Metryki wyliczane z kodu źródłowego
 - CK Metrics (1994)
 - Złożoność cyklomatyczna McCabe'a (1976)
- Metryki wyliczane z diagramów (2002)
- Narzędzia do wyliczania wartości metryk
- Przykład zastosowania metryk



Metryki wyliczane z kodu źródłowego

CK Metrics



CK Metrics - Weight Method per Class (WMC)

```
class ClassA{  
    public void method1 {...}  
    protected void method2 {...}  
    private void method3 {...}  
}
```

WMC = 3



CK Metrics - Response For a Class (RFC)

```
class ClassA{  
    private ClassB _obj;  
    public void method1 {...}  
    public void method2 {_obj.method();}  
    private void method3 {...}  
}
```

RFC = 3



CK Metrics - Response For a Class (RFC)

- Typowe wartości: 20-100
- Wysoka wartość metryki wskazuje na:
 - Dużą złożoność wewnętrzną klasy
 - Dużą odpowiedzialność przypisaną do klasy
 - Prawdopodobnie duże koszty testowania i pielęgnacji klasy



CK Metrics - Coupling Between Objects (CBO)

```
class ClassA extends BaseClass{  
    private ClassB _obj;  
    protected void method1 {  
        ClassC obj = new ClassC();  
        StringBuilder sb = new StringBuilder("txt");  
    }  
}
```

CBO = 2



CK Metrics - Coupling Between Objects (CBO)

- Typowe wynosi do 5 (im mniejsza wartość metryki tym lepiej)
- Klasy o niskiej wartości metryki:
 - Są łatwe w modyfikacji
- Wysokie wartości metryki wskazują na:
 - Niepoprawne rozmieszczenie w klasach metod i atrybutów
 - Prawdopodobnie duże koszty testowania i pielęgnacji klasy



CK Metrics - Depth of Inheritance Tree (DIT)

```
class ClassA{...}
```

```
class ClassB extends ClassA {...}
```

```
class ClassC extends ClassB {...}
```

$\text{DIT}(\text{ClassB}) = 1$

$\text{DIT}(\text{ClassC}) = 2$



CK Metrics - Depth of Inheritance Tree (DIT)

- Zalecane wartości metryki: do 5
- Wysoka wartość metryki wskazuje na:
 - Specjalizację klasy
 - Duży stopień powtórnego wykorzystania kodu poprzez dziedziczenie



CK Metrics - Number of Children (NOC)

```
class ClassA{...}
```

```
class ClassB extends ClassA {...}
```

```
class ClassC extends ClassA {...}
```

```
class ClassD extends ClassB {...}
```

$\text{NOC}(\text{ClassA}) = 2$

$\text{NOC}(\text{ClassB}) = 1$



CK Metrics - Number of Children (NOC)

- Wysoka wartość metryki świadczy o:
 - Obejmowaniu przez klasę zbyt dużego obszaru odpowiedzialności
 - Trudnościach z pielęgnacją klasy
 - Konieczności dokładnego przetestowania klasy
- Niska wartość średnia metryki dla wszystkich klas świadczy o:
 - Znikomym stosowaniu dziedziczenia



CK Metrics - Lack of Cohesion Of Methods (LCOM)

```
class ClassA{  
    private int a,b,c,d,e;  
    void method1() {a=0;}  
    void method2() {b=0;}  
    void method3() {c=0;}  
    void method4() {d=0;}  
    void method5() {  
        e=0;  
        method4();  
    }  
}
```

if $|P| > |Q|$

$LCOM = |P| - |Q|$

else

$LCOM = 0$

I_j – zbiór atrybutów używanych
przez metodę j

$P = \{(I_i, I_j) \mid I_i \cap I_j = \emptyset\}$

$Q = \{(I_i, I_j) \mid I_i \cap I_j \neq \emptyset\}$



CK Metrics - Lack of Cohesion Of Methods (LCOM)

```
class ClassA{  
    private int a,b,c,d,e;  
    void method1() {a=0;}  
    void method2() {b=0;}  
    void method3() {c=0;}  
    void method4() {d=0;}  
    void method5() {  
        e=0;  
        method4();}  
}
```

$Q = \{ (4,5) \}$

$P = \{ (1,2), (1,3), (1,4), (1,5),$
 $(2,3), (2,4), (2,5), (3,4),$
 $(3,5) \}$

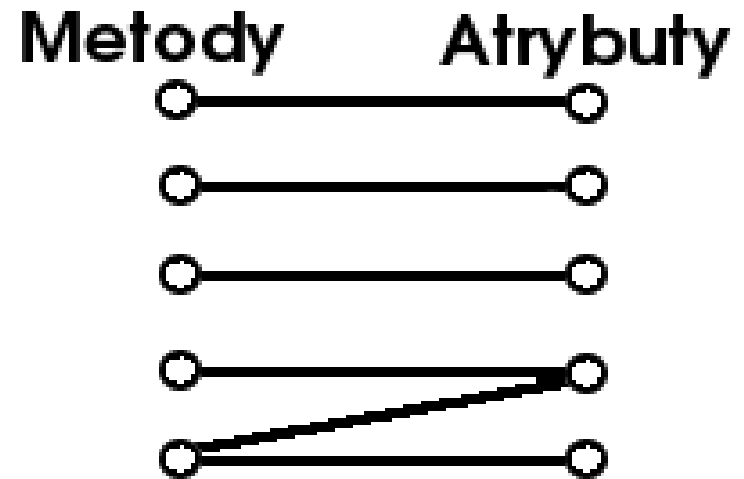
$|Q| = 1$

$|P| = 9$

$LCOM = |P| - |Q| = 8$

CK Metrics - Lack of Cohesion Of Methods (LCOM3)

```
class ClassA{  
    private int a,b,c,d,e;  
    void method1() {a=0;}  
    void method2() {b=0;}  
    void method3() {c=0;}  
    void method4() {d=0;}  
    void method5() {  
        e=0;  
        method4();}  
}
```



$$LCOM = \frac{(k/a - m)}{(1 - m)} = 0,95$$

CK Metrics - Lack of Cohesion Of Methods (LCOM)

- Przyjmuje wartości: 0-2
- Anomalią są wartości większe od 1
- Zbyt wysokie wartości metryki wskazują na:
 - Błędne rozmieszczenie metod i atrybutów w klasach

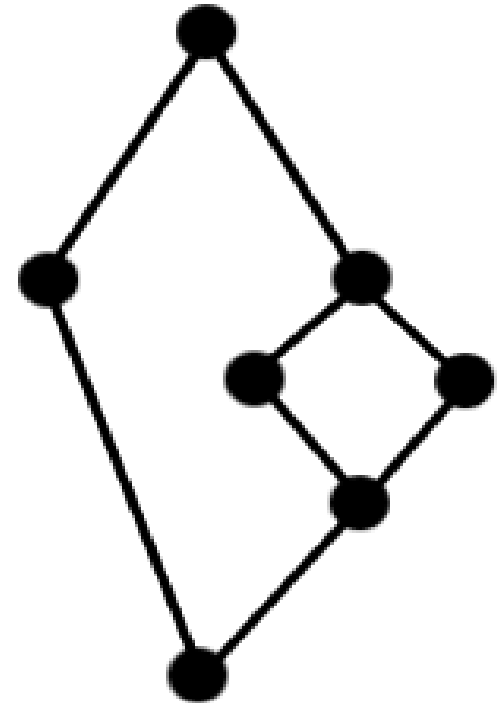


Metryki wyliczane z kodu źródłowego

Złożoność cyklomatyczna McCabe'ay

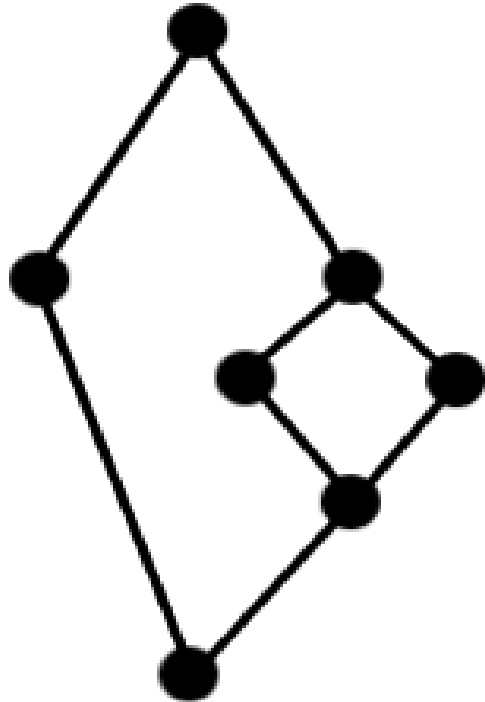
Złożoność cyklomatyczna McCabe'ay

```
boolean method(float a){  
    boolean b;  
    if( a<0 )  
        b=false  
    else  
        if(a<=1)  
            b=true;  
        else  
            b=false;  
    return b;}
```



Złożoność cykломatyczna McCabe'ay

$$V = e - n + 2p$$



V – liczba cykломatyczna grafu

e – liczba krawędzi grafu

n – liczba wierzchołków

p – liczba niepołączonych części grafu

Złożoność cyklomatyczna McCabe'ay

- Wartość metryki nie powinna przekraczać 10
- Wysokie wartości metryki powodują:
 - Trudności z testowaniem kodu
 - Problemy ze zrozumieniem i pielęgnacją kodu



Metryki wyliczane z diagramów

Kim H., Boldyref C., *Developing Software Metrics Applicable to UML*

Metryki modelu

- Number of the packages in a model (NPM)
- Number of the classes in a model (NCM)
- Number of actors in a model (NAM)
- Number of the use cases in a model (NUM)
- Number of the objects in a model (NOM)



Metryki modelu

- Number of the associations in a model (NASM)
- Number of the aggregations in a model (NAGM)
- Number of the inheritance relations in a model (NIM)



Metryki klasy

- Number of the attributes in a class – unweighted (NATC1)
- Number of the attributes in a class - weighted (NATC2)
- Number of the operations in a class - unweighted (NOPC1)
- Number of the operations in a class - weighted (NOPC2)

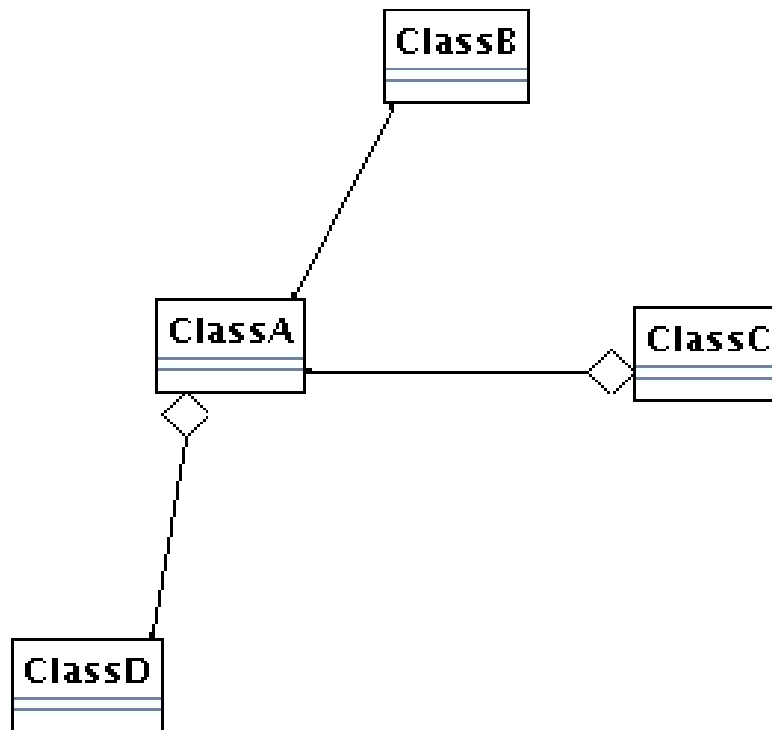
Wagi: 1 – public

0,5 – protected

0 - private

Metryki klasy

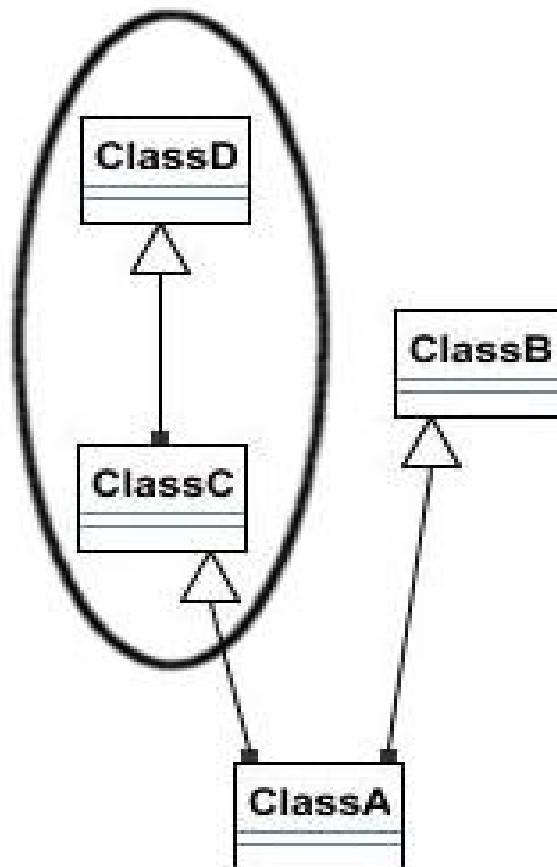
- Number of the associations linked to a class (NASC)



$$\text{NASC}(\text{ClassA}) = 3$$

Metryki klasy

- Depth of inheritance tree (DIT)

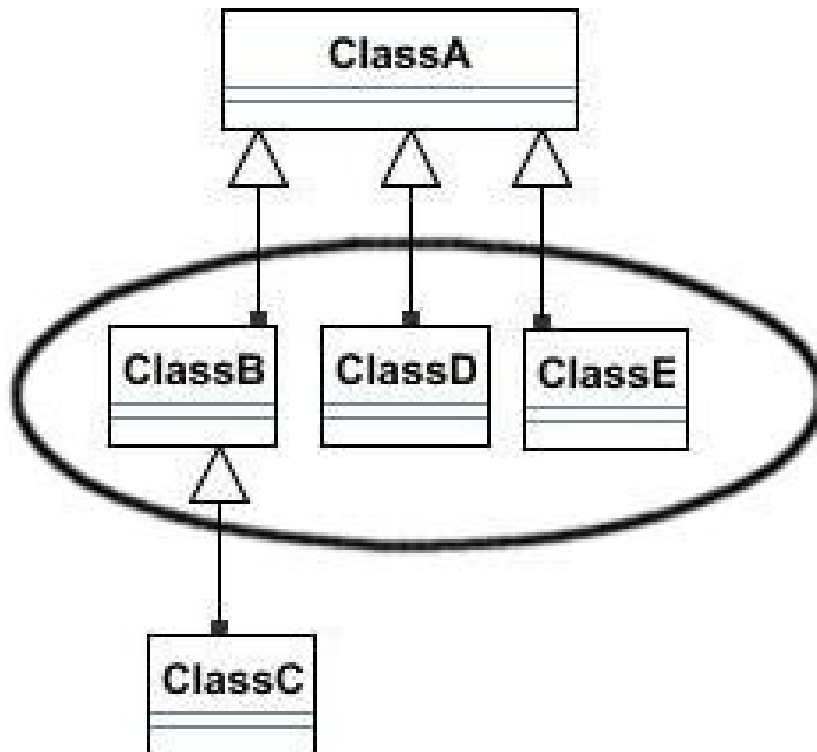


$\text{DIT}(\text{ClassA}) = 2$

$\text{DIT}(\text{ClassC}) = 1$

Metryki klasy

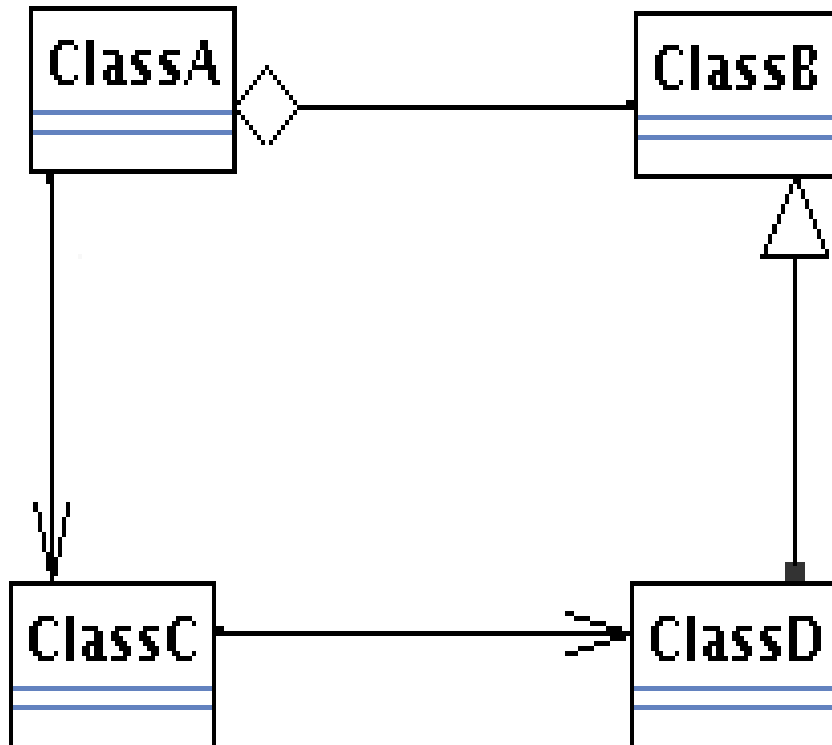
- Number of the subclasses of a class (NSUBC)



$$\text{NSUBC}(\text{ClassA}) = 3$$

Metryki klasy

- Coupling between classes (CBC)



$CBC(ClassA) = 2$

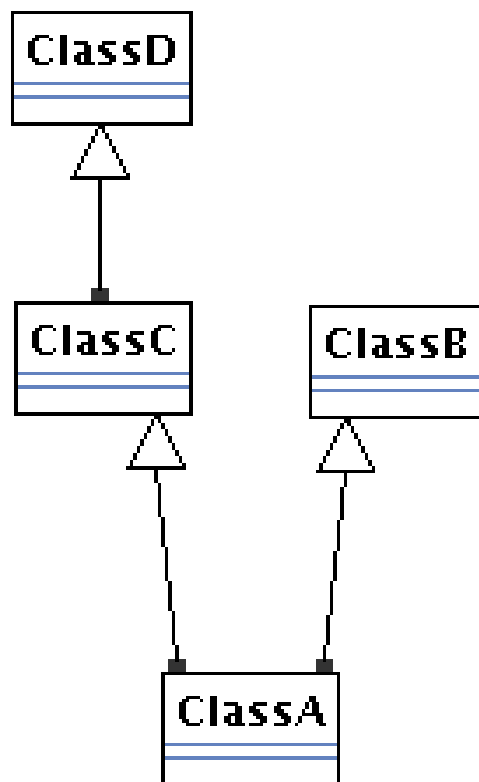
$CBC(ClassB) = 0$

$CBC(ClassC) = 1$

$CBC(ClassD) = 0$

Metryki klasy

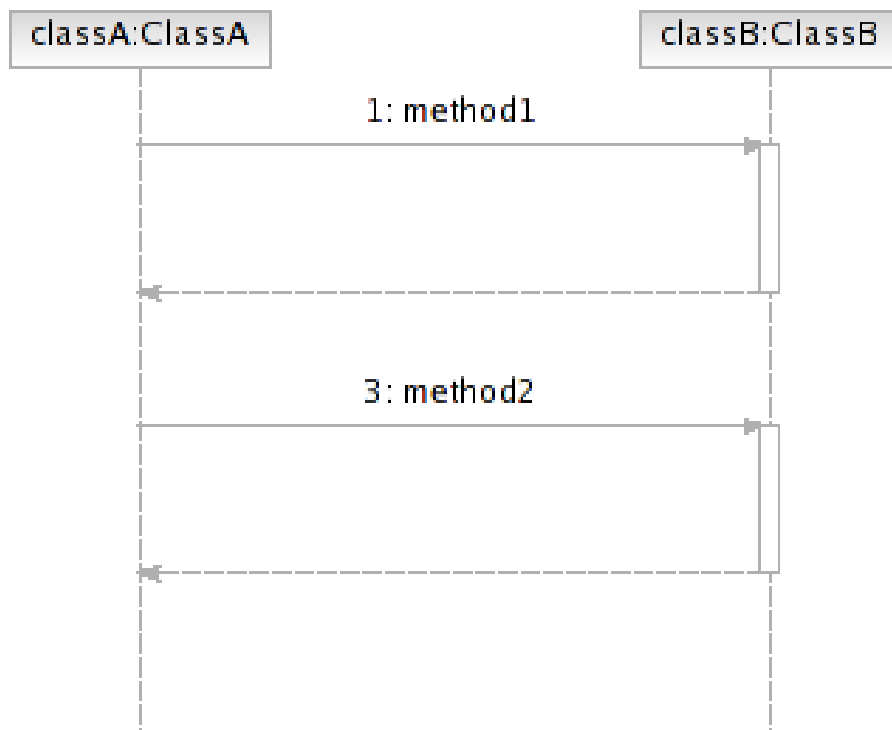
- Number of the superclasses of a class (NSUPC)



$$\text{NSUPC}(\text{ClassA}) = 2$$

Metryki klasy

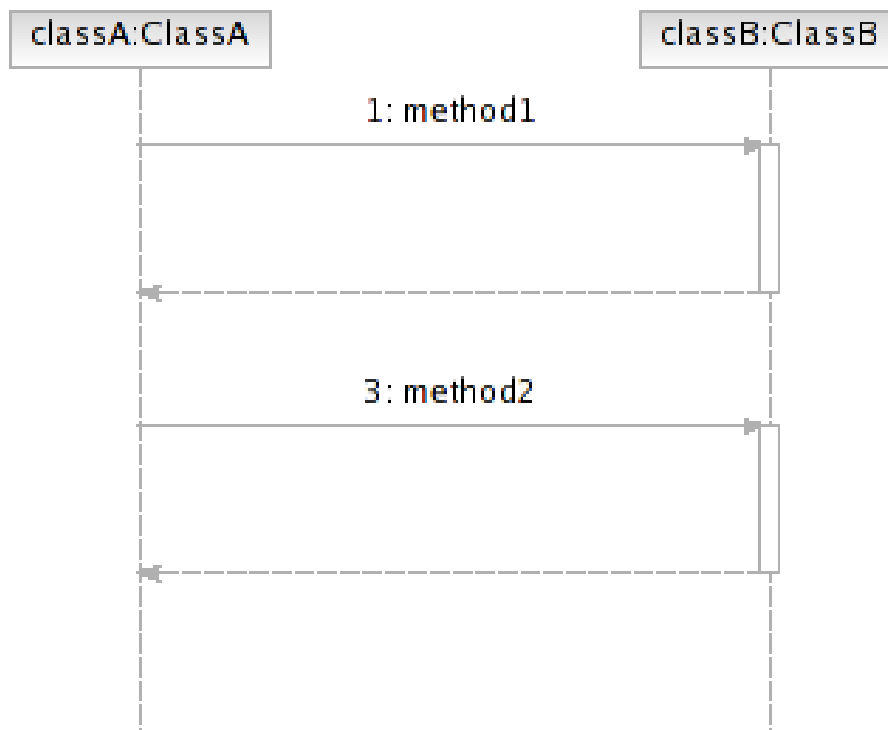
- Number of messages sent by the instantiated objects of a class (NMSC)



$$\text{NMSC}(\text{ClassA}) = 2$$

Metryki klasy

- Number of messages received by the instantiated objects of a class (NMRC)



$$\text{NMRC}(\text{ClassB}) = 2$$

Metryki przypadku użycia

- Number of actors associated with a use case (NAU)
- Number of message associated with a use case (NMU)
- Number of system classes associated with a use case (NSCU)



Narzędzia do wyliczania wartości metryk



NDepend

The screenshot displays the NDepend application interface. The top status bar shows 'Octopus 1.0.2533.27602 (Fri 08 Dec 15:29 most recent)' and various project statistics: 'Assemblies:10', 'Namespaces:11', 'Types:194', 'Methods:2 379', and 'Fields:781'. The main window is divided into several panes:

- Code Tree View:** Lists methods and their 'DepthOfIsL' values. The 'GetInstance(String)' method is highlighted.
- Query Result:** A table showing statistics for the selected query.
- Dependency Graph:** A visual representation of code dependencies, showing nodes like 'Octopus.CoreDomain', 'Octopus.Manag', 'Octopus.Se', 'CreditContract', 'EventPro', 'Person', 'Package', 'Octopus.Shared', and 'System.Globa'.
- Query Editor:** Contains a SQL-like query: `SELECT METHODS OUT OF TYPES "System.Data.SqlClient.SqlConnection" WHERE IsUsing "System.Data.SqlClient.SqlConnection" ORDER BY DepthOfIsUsing`.

At the bottom, a status bar indicates 'Query compilation succeeded, no warning'.

methods	DepthOfIsL
get_SqlConnection()	1
get_SqlTransaction()	1
.ctor()	1
CloseConnection()	1
.ctor(String)	1
GetInstance()	2
.ctor()	2
GetInstance(String)	2
.ctor(String)	2
.ctor(String)	2
AddGroup(ClientGroup)	2
AddPerson(Person)	2
UpdatePerson(Person)	2
UpdateGroup(ClientGro...	2
.ctor()	2
UpdateProvisioningTabl...	2

# Items	234
# Items tested	4222
% matched items	5.5424%
Sum	1087
Average	4.6453
Min	1
Max	8
Standard deviation	1.8874
Variance	3.5622

- Wsparcie dla platformy .NET
- Wylicza ponad 60 metryk kodu, w tym:
 - Wszystkie metryki z pośród CK Metrics
 - Złożoność cykliczną



CKJM Chidamber and Kemerer Java Metrics - SeaMonkey

Designed for use with [CKJM](#) and [Ant](#).

Top 10: wmc [\[wmc\]](#) [\[dit\]](#) [\[noc\]](#) [\[cbo\]](#) [\[rfc\]](#) [\[lcom\]](#) [\[ca\]](#) [\[npm\]](#) [\[explanations\]](#)

name	wmc	dit	noc	cbo	rfc	lcom	ca	npm
org.hsqldb.jdbc.jdbcDatabaseMetaData	172	1	0	6	197	14692	1	165
org.hsqldb.jdbc.jdbcResultSet	147	1	0	18	198	10333	5	139
org.hsqldb.Table	142	2	2	41	330	8117	46	24
org.hsqldb.Expression	90	1	0	22	190	581	18	2
org.hsqldb.jdbc.jdbcCallableStatement	83	3	0	7	121	3397	1	81
org.hsqldb.Server	72	1	1	23	187	1896	5	45
org.hsqldb.DatabaseCommandInterpreter	72	1	0	44	301	0	2	1
org.hsqldb.Library	71	1	0	6	139	2473	5	64
org.hsqldb.Result	60	1	0	12	104	1054	36	27
org.hsqldb.Parser	60	1	0	27	191	842	4	0

Top 10: dit [\[wmc\]](#) [\[dit\]](#) [\[noc\]](#) [\[cbo\]](#) [\[rfc\]](#) [\[lcom\]](#) [\[ca\]](#) [\[npm\]](#) [\[explanations\]](#)

name	wmc	dit	noc	cbo	rfc	lcom	ca	npm
org.hsqldb.rowio.RowInputTextQuoted	3	4	0	3	17	1	1	2
org.hsqldb.rowio.RowOutputTextQuoted	3	4	0	2	10	3	1	1
org.hsqldb.rowio.RowInputTextLog	23	3	0	10	52	223	1	8
org.hsqldb.Expression\$Collector	3	3	0	4	7	3	4	0
org.hsqldb.View	9	3	0	17	36	10	6	0
org.hsqldb.TextTable	13	3	0	15	39	30	1	2
org.hsqldb.scriptio.ScriptWriterZipped	4	3	0	6	14	4	1	1
org.hsqldb.rowio.RowInputBinary	23	3	0	9	50	243	9	9
org.hsqldb.MemoryNode	4	3	0	3	6	0	1	1
org.hsqldb.scriptio.ScriptReaderZipped	2	3	0	4	12	1	1	0

- Wylicza metryki ze źródeł języka java
- Wylicza wszystkie metryki z pośród CK Metrics

<http://www.spinellis.gr/sw/ckjm/>



Eclipse Metrics plugin

The screenshot shows the Eclipse Metrics plugin interface. The title bar reads 'Metrics - /net.sourceforge.metrics'. The table displays various metrics for the project, including the number of packages, methods, lines of code, and interfaces. The table is organized into columns: Metric, Total, Mean, Std. Dev., Maximum, Resource causing Maximum, and Method. The metrics are grouped by expandable/collapsible icons. The 'Lines of Code (avg/max per method)' section is expanded, showing details for the 'MetricsTable' class.

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
Number of Packages	16					
Number of Methods (avg/max per type)	1310	6.65	8.553	76	/net.sourceforge.metrics/tgsrc/com/touchgrap...	
tgsrc	489	7.191	11.544	76	/net.sourceforge.metrics/tgsrc/com/touchgrap...	
src	761	6.238	6.553	45	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.core.sources	108	15.429	12.129	45	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.ui	77	9.625	10.111	33	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.core	198	6.6	7.093	27	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.ui.preferences	52	6.5	7.467	26	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.ui.dependencies	95	5.588	3.727	15	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.internal.persistence	18	4.5	4.33	12	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.internal.prevayler.implementa...	54	5.4	2.871	10	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.internal.xml	41	4.1	2.022	9	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.calculators	79	4.158	2.254	8	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.propagators	31	5.167	1.067	7	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.internal.tests	8	2.667	1.886	4	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.internal.prevayler	0	0	0			
classycle	60	8.571	2.556	13	/net.sourceforge.metrics/classycle/classycle/g...	
Lines of Code (avg/max per type)	6593	33.467	49.02	339	/net.sourceforge.metrics/tgsrc/com/touchgrap...	
Number of Interfaces (avg/max per packageFragment)	16	1	1.414	4	/net.sourceforge.metrics/src/net/sourceforge/...	
Lines of Code (avg/max per method)	6593	4.812	7.355	69	/net.sourceforge.metrics/classycle/classycle/g...	calculateAttributes
classycle	324	5.4	9.94	69	/net.sourceforge.metrics/classycle/classycle/g...	calculateAttributes
tgsrc	2321	4.661	8.278	59	/net.sourceforge.metrics/tgsrc/com/touchgrap...	scrollSelectPanel
src	3948	4.862	6.473	52	/net.sourceforge.metrics/src/net/sourceforge/...	setMetrics
net.sourceforge.metrics.ui	544	6.8	8.707	52	/net.sourceforge.metrics/src/net/sourceforge/...	setMetrics
MetricsTable.java	194	10.778	13.831	52	/net.sourceforge.metrics/src/net/sourceforge/...	setMetrics
MetricsTable	194	10.778	13.831	52	/net.sourceforge.metrics/src/net/sourceforge/...	setMetrics
setMetrics	52					

- Wylicza metryki ze źródeł języka java
- Wylicza: CK Metrics, złożoność cyklomatyczna i inne

<http://sourceforge.net/projects/metrics>



NetBeans Metrics Module

NetBeans IDE Development Version (Build 200204171818) - Project Default

File Edit View Project Build Debug Versioning Tools Window Help

Editing GUI Editing Running Debugging

Explorer [Filesystems]

Filesystems

- /home/tball/tmp/ffj_user/s
- /home/tball/tmp
- CVS /home/tball/src/netbea
 - org
 - netbeans
 - modules
 - metrics

Metrics

Class Name	WMC	CBO	RFC	MPC	DIT	NOC	NT
org.netbeans.modules.metrics.ClassMetrics	97	45	134	202	2	0	0
org.netbeans.modules.metrics.MethodMetrics	73	29	101	139	1	0	0
org.netbeans.modules.metrics.MetricsNode	19	26	37	41	4	0	0
org.netbeans.modules.metrics.ApproveMetricAction	30	25	57	82	5	0	2
org.netbeans.modules.metrics.MetricsFactory	21	20	45	55	2	0	2
org.netbeans.modules.metrics.ApprovalsFile	15	18	50	85	1	0	0
org.netbeans.modules.metrics.options.MetricSettings	0	18	0	0	1	0	0
org.netbeans.modules.metrics.AbstractMetric	14	17	17	19	1	7	0
org.netbeans.modules.metrics.ClassApprovals	27	17	43	70	1	0	0
org.netbeans.modules.metrics.TableSorter	43	17	46	61	3	0	0
org.netbeans.modules.metrics.ReferenceScanner	36	16	47	76	2	0	0
org.netbeans.modules.metrics.ShowMetricsAction	29	16	38	64	5	0	1
org.netbeans.modules.metrics.TrafficNode	15	16	25	23	4	0	1
org.netbeans.modules.metrics.MetricDetailsInvoker	12	15	35	30	2	0	0
org.netbeans.modules.metrics.ApprovalsFile.XMLHandler	17	14	32	72	2	0	0
org.netbeans.modules.metrics.MetricValue	16	14	23	43	1	0	0
org.netbeans.modules.metrics.MethodApprovals	18	13	31	35	1	0	0
org.netbeans.modules.metrics.Metric	0	13	0	0	1	0	0
org.netbeans.modules.metrics.MetricsPane	2	12	28	31	6	0	0
org.netbeans.modules.metrics.NTMetric	18	11	33	39	2	0	0
org.netbeans.modules.metrics.NodeHandler	0	11	0	0	1	0	0
org.netbeans.modules.metrics.MetricsTableModel	17	10	15	9	2	0	0
org.netbeans.modules.metrics.TrafficNode.Light	4	10	9	11	1	0	0
org.netbeans.modules.metrics.RFCMetric	17	9	30	41	2	0	0
org.netbeans.modules.metrics.CBOMetric	35	8	29	61	2	0	1
org.netbeans.modules.metrics.MPCMetric	14	8	28	22	2	0	0
org.netbeans.modules.metrics.MetricsModule	6	8	15	23	3	0	0

- Wylicza metryki ze źródeł języka java
- Wylicza wszystkie metryki z pośród CK Metrics

<http://metrics.netbeans.org/>



CCCC

Metric	Tag	Overall	Per Function
Lines of Code	LOC	35	*****
McCabe's Cyclomatic Number	MVG	5	*****
Lines of Comment	COM	1	*****
LOC/COM	L_C	35.000	
MVG/COM	M_C	5.000	
Weighted Methods per Class (weighting = unity)	WMC1	2	
Weighted Methods per Class (weighting = visible)	WMCv	0	
Depth of Inheritance Tree	DIT	0	
Number of Children	NOC	0	
Coupling between objects	CBO	2	
Information Flow measure (inclusive)	IF4	0	*****
Information Flow measure (visible)	IF4v	0	*****
Information Flow measure (concrete)	IF4c	0	*****

Definitions and Declarations

- Wylicza metryki ze źródeł języków C++ i java
- Wylicza CK Metrics, złożoność cyklomatyczną i inne

<http://sourceforge.net/projects/cccc>



Understand for ...

Non-Alpha	A	B	<u>C</u>	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
-----------	---	---	----------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Class	LCOM (Percent Lack of Cohesion)	DIT (Max Inheritance Tree)	IFANIN (Count of Base Classes)	CBO (Count of Coupled Classes)	NOC (Count of Derived Classes)	R (Count of Associations Made)
CChemia	0	1	3	2	0	
CChemik	0	1	3	0	0	
CCOMChemikModule	0	1	1	0	0	

Non-Alpha	A	B	<u>C</u>	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
-----------	---	---	----------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- Wylicza metryki ze źródeł języków Ada, C++, Delphi, Fortran, java i JOVIAL
- Wylicza wszystkie metryki z pośród CK Metrics, oraz wiele innych

<http://www.scitools.com/>



Przykład zastosowania metryk

Badanie zależności pomiędzy wartościami metryk, a czasem potrzebnym na poprawienie defektów

Wartości metryk kodu źródłowego

	Max	Min	Średnia	Mediana	δ
WMC	47	0	9,73	7,5	8,11
DIT	8	1	3,34	3	2,41
NOC	22	0	0,33	0	1,98
RFC	87	1	25,54	21	17,79
CBO	66	0	15,56	9	16,11
LCOM	2	0	0,67	0,88	0,45



Uzyskane wartości metryk wyliczanych z diagramów

	Max	Min	Średnia	Mediana	d
NATC1	79	0	6,36	3	9,64
NATC2	12	0	1,12	0	1,88
NOPC1	50	0	8,32	6	8,19
NOPC2	43,5	0	6,52	4	6,83
NASC	8	0	0,86	0	1,24
DIT	3	0	0,36	0	0,73
NSUBC	10	0	0,24	0	1,12



Korelacja pomiędzy wartościami metryk a niską jakością klasy

Projekt	A	B	C	D	Wszystkie
WMC	0,83	0,64	0,24	0,43	0,52
DIT	0,82	-0,3	0,32	0,26	0,03
NOC	0,13	0,03	-0,1	-0,1	0,13
RFC	0,07	0,07	0	0,38	0,04
CBO	0,65	0,59	0,36	0,81	0,3
LCOM	-0,2	0,11	0,14	0,33	-0,1



Korelacja pomiędzy wartościami metryk a niską jakością klasy

Projekt	A	B	C	D	Wszystkie
NATC1	0,28	0,09	-0,12	0,14	0,2
NATC2	0,13	0,2	-0,21	0,18	0,05
NOPC1	0,23	0,54	0,34	0,34	0,28
NOPC2	0,13	0,5	-0,21	0,25	0,2
NASC	0,13	0,27	0,58	0,38	0,02
DIT (diag)	-0,24	0,55	-0,1	-0,11	-0,08
NSUBC	-0,1	0	-0,1	-0,11	-0,07

Model predykcji jakości na podstawie wartości metryk

- Wejście modelu (x): wartości metryk.
- Wyjście modelu (y): szacowany czas potrzebny na poprawianie klasy.
- Badane modele postaci:
 - $y = a_0 + a_1x_1 + \dots + a_nx_n$
 - $y = a_0 + a_1x_1 + \dots + a_nx_n + \dots + a_mx_{n-1}x_n$
- Zastosowanie regresji liniowej

Uzyskane rezultaty – metryki wyliczane z kodu źródłowego

$$\begin{aligned} y = & -0,07 + 0,05*WMC - 0,11*DIT - 0,06*NOC \\ & - 0,01*RFC + 0,04*CBO - 0,12*LCOM \\ & - 0,01*WMC*DIT - 0,01*WMC*NOC \\ & + 0,03*WMC*LCOM + 0,06*DIT*NOC \\ & + 0,01*DIT*RFC - 0,01*DIT*CBO \\ & + 0,09*DIT*LCOM + 0,03*NOC*CBO \\ & - 0,04*NOC*LCOM - 0,03*CBO*LCOM \end{aligned}$$

$$r = 0,906$$



Uzyskane rezultaty – metryki wyliczane z diagramów

$$\begin{aligned} y = & 0,15 + 0,0001 * NATC1 \\ & - 0,0411 * NATC2 + 0,0481 * NOPC1 \\ & - 0,0171 * NOPC2 - 0,0079 * NASC \\ & - 0,0421 * DIT - 0,0904 * NSUBC \end{aligned}$$

$$r = 0,3272$$



Uzyskane rezultaty – metryki wyliczane z kodu źródłowego

	Modele bez członów interakcyjnych	Modele z członami interakcyjnymi
Projekt A	WMC-0,16; r=0,85	DIT*NOC-0,22; WMC*LCOM-0,1; r=0,97
Projekt B	WMC-0,03; CBO-0,03; r=0,76	LCOM-0,1; CBO*LCOM-0,1; r=0,82
Projekt C	WMC-0,01; DIT-0,04; r=0,54	LCOM-2,5; DIT*LCOM-0,23; r=0,86
Projekt D	WMC-0,03; DIT-0,19; r=0,88	DIT-0,3; WMC-0,2; r=0,99
Wszystkie projekty	WMC-0,06; r=0,58	WMC-0,05; CBO-0,04; DIT*LCOM-0,09; DIT*NOC-0,06; r=0,906
Klasy impement.	WMC-0,05; NOC-0,06; r=0,58	WMC-0,05; NOC*CBO-0,04; r=0,84
Klasy gui	WMC-0,08; r=0,72	WMC-0,2; DIT*CBO-0,02; NOC*CBO-0,04; r=0,84

r – współczynnik korelacji



Uzyskane rezultaty – metryki wyliczane z diagramów

	Modele bez członów interakcyjnych	Modele z członami interakcyjnymi
Projekt A	NOPC2-0,13; $r=0,4$	NATC2*NOPC1-0,58; NOPC1*NSUBC-0,85; NASC*NSUBC-1,19; $r=0,59$
Projekt B	NASC-0,08; NOPC1-0,06; $r=0,63$	NATC2-0,4; NATC1*NASC-0,04; NOPC2*NASC-0,06; $r=0,71$
Projekt C	NOPC1-0,06; $r=0,88$	NOPC2-0,1; NOPC2*NASC-0,5; NATC2*NOPC1-0,36; $r=0,97$
Projekt D	NATC2-0,19; NOPC1-0,03; $r=0,88$	NATC1-0,2; NATC2-0,3; NOPC2-0,2; NATC2*NASC-0,07; $r=0,99$
Wszystkie projekty	NOPC1-0,048; $r=0,33$	NATC2-0,15; NOPC1*DIT-0,19; NASC*NSUBC-0,24; $r=0,46$

r – współczynnik korelacji

Literatura

- HENDERSON-SELLERS B., *Object-Oriented Metrics, measures of Complexity*. Prentice Hall, 1996.
- KAN S.H., *Metrics and Models in Software Quality Engineering*. Addison-Wesley Professional, 2002 (*Metryki i modele w inżynierii jakości oprogramowania*. PWN, 2006.)



Politechnika Wrocławska

Dziękuję za uwagę

Marian Jureczko