

PureCoverage

Jest to narzędzie pakietu Rational PurifyPlus pozwalające ocenić poziom pokrycia kodu testowanej aplikacji przez opracowane testy. Każdorazowo PureCoverage weryfikuje, czy linia bądź funkcja została wykonana i ile razy to nastąpiło i zbiera wynik. Wykonując szereg uruchomień dla wielu testów PureCoverage nie tylko pozwala ocenić łączne pokrycie, ale także umożliwia dokonanie oceny nowego testu pod względem wzrostu pokrycia kodu. Dzięki tym możliwościom pozwala na zwiększenie niezawodności programu.

1. Możliwości narzędzia

PureCoverage umożliwia badanie pokrycia kodu aplikacji napisanych w następujących językach programowania: Visual C++, Visual Basic, Java, Microsoft Visual Studio.NET. Narzędzie to pozwala na:

- Dokonanie ogólnej oceny testów obserwując pokrycie kodu
- Zlokalizowanie metod, które nie zostały wywołane (zatem nie były przetestowane)
- Dla metod, które były uruchomione, PureCoverage pozwala określić liczbę nieprzetestowanych linii
- Pokazanie w kodzie źródłowym tych linii, które nie zostały wykonane, natomiast dla wykonanych - liczbę odwołań do nich
- Połączenie wyników działania wielu testów prezentując sumaryczne pokrycie kodu i metod
- Ułatwienie testowanie pokrycia poprzez integrację z narzędziami: Microsoft Visual C++, Visual Basic
- Dokonanie filtracji wywołań standardowych bibliotek (np. MFC)
- Zapisanie rezultatu pokrycia kodu

2. Korzyści z używania narzędzia

Systematyczne wykonywanie testów na pokrycie kodu w trakcie prac nad tworzoną aplikacją, pozwala na ocenę postępu w pracach nad testami, w stosunku do rozwoju kodu źródłowego programu. Ponieważ kod programu systematycznie podlega zmianom i jego zawartości jest powiększana o nowe funkcje lub metody, częste poddawanie nowo powstałego kodu testowaniu okazuje się niezbędne. Dzięki temu możliwe jest we wczesnej fazie tworzenia aplikacji zweryfikowanie, czy początkowo zaprojektowanie testy spełniają warunek całkowitego pokrycia kodu. Jeśli nie, to należy rozszerzyć zestaw testów.

3. Badanie pokrycia kodu

Wynik działania narzędzia PureCoverage prezentowany jest w trzech oknach:

- Coverage Browser
- Function List
- Annotated Source (tylko dla opcji z pokryciem linii)

Coverage Browser

Okno to ma możliwość pokazania pokrycia kodu względem metod oraz linii w sposób hierarchiczny - zakładka „Module View”:

- dla całej aplikacji
- kolejnych modułów
- metod modułu (zaznaczenie wykonania oraz pokrycie względem linii)

Coverage Item	Calls	Functions Missed	Functions Hit	% Functions Hit	Lines Missed	Lines Hit	% Lines Hit
Auto Merge @ 10:15:50.0758	2315	43	32	42.67	1316	707	34.91
D:\Pw\ALK\PEFILE.DLL	69	11	3	21.43	129	55	29.88
D:\Pw\ALK\PROCS.DLL	29	1	4	80.00	6	24	75.00
D:\Pw\ALK\pwalk.exe	2227	31	25	44.64	1181	628	34.72

Dostępna jest także prezentacja wyników względem plików (zamiast modułów) także w sposób hierarchiczny – zakładka „File View”.

Function List

Prezentuje listę wszystkich funkcji z liczbą wywołań oraz wykonanymi liniami i procentową skutecznością pokrycia kodu tej metody dla wybranego testu (dla opcji badania linii).

Annotated Source

Służy do zidentyfikowania linii w metodach lub funkcjach, które nie zostały wykonane. Ponieważ okno to prezentuje wynik pracy narzędzia PureCoverage w kodzie źródłowym, umożliwia w prosty i czytelny sposób wskazanie linii nie wywołanych w tym przebiegu testowym. Dla pozostałych linii prezentowana jest liczba odwołań.

- Kolorem czerwonym przedstawione są linie, które nie zostały wykonane
- Kolor niebieski oznacza, że dana linia została wywołana
- Kolorem różowym oznaczane są te fragmenty kodu, które powodują wywołanie metody, dla której oddzielnie prezentuje się wynik pokrycia

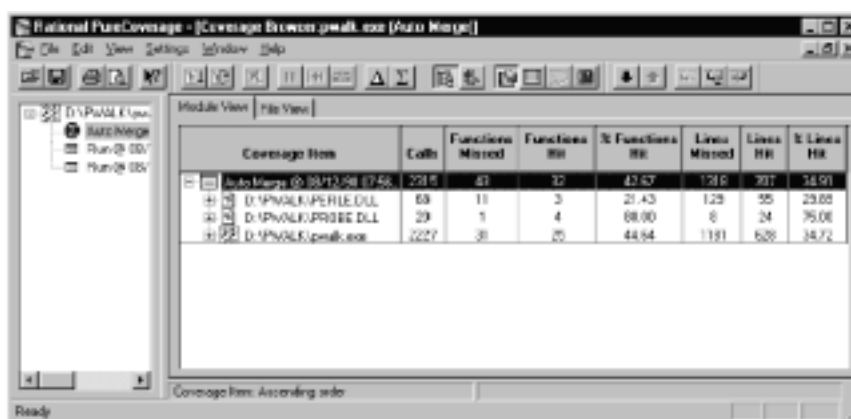
```

209 bool Btree::insertRecord(Record *rec)
210 {
211     Node *ptr,*res;
212     int found=-1;
213
214     ptr=findEntry(rec->key(),found);
215     if(found>=0) return false;    //rekord o danym kluczu juz istnieje
216     if(ptr==NULL)
217     {
218         root=new Node(treeOrder);
219         root->addRecord(rec);
220     } else
221     {
222         res=ptr->addRecord(rec);
223         if(res!=NULL) root=res;    //powstal nowy root
224     }
225     return true;
226 }

```

3.1. Ponowne uruchomienie

Po dokonaniu poprawek w kodzie lub stworzeniu nowych testów (zmodyfikowaniu) należy uruchomić testowanie ponownie. PureCoverage w oknie nawigacyjnym prezentuje wywołania dla poszczególnych testów. Mając włączoną opcję „Auto merge” pojawia się w tym oknie dodatkowe pole będące połączonym wynikiem pod względem pokrycia. Dzięki temu możliwa jest ocena najlepszego testu i ich zbioru pod względem maksymalnego pokrycia.



Coverage Item	Colts	Functions Missed	Functions Hit	% Functions Hit	Lines Missed	Lines Hit	% Lines Hit
Auto Merge (D:\PVAL\PERIOD.DLL)	2285	40	32	44.67	1369	397	28.91
D:\PVAL\PERIOD.DLL	69	11	3	21.43	123	55	29.68
D:\PVAL\PROBE.DLL	29	1	4	68.00	6	24	75.00
D:\PVAL\pmval.exe	2227	38	25	44.64	1191	628	34.72

4. Dodatkowe możliwości

PureCoverage potrafi współpracować z narzędziem Rational Robot – uruchamiane skrypty w programie Robot mogą posłużyć jako testy lub ich zestawy, którymi poddawana jest dana aplikacja w celu zbadania pokrycia kodu.

W opcjach ustawień uruchamiania programu testowanego możliwe jest wyspecyfikowanie, które moduły (także dołączane biblioteki) mają być traktowane jako:

- Nie poddawane badaniu pokrycia
- Analizowane pod względem pokrycia linii
- Analizowane pod względem pokrycia funkcji / metod