

SZYMCZYK Tomasz

DARMOWE OPROGRAMOWANIE WSPOMAGAJĄCE NAUCZANIE PROJEKTOWANIA OBIEKTOWEGO -ANALIZA PORÓWNAWCZA

Streszczenie

W artykule poruszono problem nauczania projektowania obiektowego. Komercyjne oprogramowanie umożliwiające projektowanie systemów informatycznych z wykorzystaniem UML jest drogie. Alternatywą jest więc darmowe oprogramowanie typu: open source czy freeware. W artykule dokonano przeglądu aktualnych programów, zaproponowano kryteria porównawcze i dokonano ich analizy porównawczej.

WSTĘP

W ciągu ostatnich kilku lat architektura zorientowana na modele MDA (ang. Model Driven Architecture) zyskała coraz większe znaczenie w rozwoju oprogramowania. Najważniejszą ideą MDA jest modelowanie niezależne od platformy w skrócie PIM (ang. Platform Independent Model). Takie podejście stało się źródłem wszystkich innych działań projektowych. Jedną z zalet projektowania PIM jest możliwość wymiany modeli między platformami MDI (ang. Metadata Interchange), generowanie kodu, szybka walidacja i weryfikacja projektu. W takim schemacie model staje się podstawą procesu tworzenia oprogramowania. Model – obiekt jest podstawowym elementem, jaki można wykorzystać w procesie projektowania obiektowego w UML (ang. Unified Modeling Language).

1. UML JAKO WIODĄCA METODA PROJEKTOWANIA OBIEKTOWEGO

Do sukcesu podejścia MDA przyczyniła się rosnąca popularność języka UML [7,8]. Według przeprowadzonych badań preferencyjnych zunifikowany język modelowania jest najczęściej wykorzystywanym językiem w procesie projektowania systemów informatycznych [2]. Kilkanaście typów diagramów oraz wiele elementów wspomagających modelowanie sprawiają, że UML zapewnia środki niezbędne do modelowania wszystkich rodzajów systemów oprogramowania, niezależnie od ich wielkości czy zakresu tematycznego [1].

UML został stworzony przez Grady'ego Boocha, Jamesa Rumbaugh'a oraz Ivana Jacobsona w 1997 roku, kiedy to powstała wersja UML 1.0 wydana przez OMG (ang. Object Management Group) [5]. W roku 2005 została wypuszczona wersja 2.0, a następnie 2.1.1 oraz 2.1.2. Najnowsza wersja to UML 2.4.

Głównym celem UML jest modelowanie systemów informatycznych. Istnieje także możliwość wykorzystania tego języka nie tylko dla potrzeb oprogramowania.

UML jest idealnym narzędziem do prezentacji szeregu pojęć z języków obiektowych. Jest także notacją pośrednią łączącą ludzkie rozumienie struktury i działania programów, a obiektowym kodem programów. Diagramy UML ustanawiają bezpośrednie powiązanie elementów modelu pojęciowego z wykonywalnymi programami [6].

Stosowanie UML pozwala na ujednoczenie notacji, wczesne wykrywanie luk i niespójności w specyfikacji wymagań, zapewnienie zgodności użytych nazw obiektów i klas oraz ich atrybutów i metod.

Model UML projektowanego systemu ma postać diagramów graficznych. Każdy rodzaj diagramu obiera jeden punkt widzenia i ukazuje system pod jego kątem, pomijając lub upraszczając przy tym pozostałe elementy. Wysoki poziom abstrakcji pozwala na zwięzły opis nawet bardzo dużych systemów [3].

2. OPROGRAMOWANIE WSPOMAGAJĄCE NAUKĘ UML

Projektowanie systemów informatycznych w podejściu obiektowym realizowane jest z wykorzystaniem UML. Istnieje szereg zaawansowanych programów wspomagających ten proces. Należą do nich min:

- Visual Paradigm
- IBM Rational Architect
- Altova Umodel 2012
- Astah (dawniej JUDE)
- BoUML
- Enterprise Architect
- MagicDraw UML

Niektóre z wyżej wymienionych, tak jak Visual Paradigm, czy poprzednik IBM Rational Architect – Rational Rose w okrojonej wersji są lub były darmowe dla państwowych uczelni wyższych. Pozostałe firmy będące właścicielami wymienionych wyżej programów obecnie nie prowadzą takiej polityki. Ceny komercyjne na jedno stanowisko zaprezentowano w tabeli 1 poniżej.

Tab. 1. Ceny wybranego komercyjnego oprogramowania na jedno stanowisko komputerowe.

lp	Nazwa oprogramowania	Cena za 1 stanowisko
1	Altova Umodel 2012	Enterprise: 299 €
2	Astah (dawniej JUDE)	Professional 280 \$
3	BoUML	15 0€
4	Enterprise Architect	Corporate: 239 \$
5	IBM Rational Architect(dawniej Rational Rose)	3968 €
6	MagicDraw UML	Enterprise: 2399 \$
7	Visual Paradigm for UML	Enterprise: 5525 PLN

Źródło: [opracowanie własne]

Zaprezentowane ceny oprogramowania w tabeli 1 jednoznacznie pokazują iż oprogramowanie wspomagające projektowanie obiektowe jest drogie. Warto jest więc poszukać alternatywnego tańszego lub wprost darmowego oprogramowania.

2.1. Przegląd darmowego oprogramowania

Programy darmowe (ang. freeware) to określenie dotyczące oprogramowania, które może być uruchamiane, kopiowane, rozpowszechniane, analizowane oraz zmieniane i poprawiane przez użytkowników.

Do takich narzędzi UML należą:

- AmaterasUML,
- ArgoUML,

- Dia,
- Fujaba,
- StarUML,
- UMLet,
- UML pad,
- UML Sculptor,
- Violet.

AmaterasUML

AmaterasUML jest rozszerzeniem- wtyczką (ang. plug-in) platformy Eclipse. Oferuje możliwość tworzenia diagramów klas, przypadków użycia, sekwencji i aktywności. Warto zauważyć, że są to raczej podstawowe diagramy UML. Niewątpliwą zaletą programu jest możliwość generowania kodu Java.

ArgoUML

ArgoUML jest narzędziem wspierającym UML 1.4, daje możliwość stworzenia diagramu klas, przypadków użycia, sekwencji, komunikacji, aktywności, rozlokowania oraz maszyny stanowej. Nie umożliwia tworzenia diagramów dodanych w standardzie UML 2.0 i nowszych. Posiada wbudowaną opcję generowania kodu źródłowego ze schematów do następujących języków programowania obiektowego: Java, C++, C#, PHP4 oraz PHP5.

Dia

Dia nie jest typowym programem stworzonym do wspomaganie UML . Posiada bibliotekę umożliwiającą tworzenie podstawowych diagramów UML. Jest to program graficzny, nie posiada więc funkcji wykrywania błędów. Niewątpliwą zaletą Dia jest możliwość tworzenia estetycznych diagramów z możliwością eksportowania ich do wielu formatów graficznych. Interfejs użytkownika jest tak zaprojektowany, że Dia jest programem bardzo intuicyjnym w obsłudze. Ponadto potrafi generować wydruki dla bardzo dużych projektów obejmujących kilka bądź kilkanaście stron.

Fujaba

Projekt Fujaba jest rozwijany od 1997 roku przez kilka uniwersytetów w Niemczech, Belgii, Holandii, Kanadzie, Estonii, Finlandii i Szwecji. Fujaba jest narzędziem wciąż rozwijanym przez użytkowników.

Umożliwia tworzenie diagramów klas, sterowania interakcją, przypadków użycia, pakietów, maszyny stanowej oraz aktywności. Posiada wsparcie dla generowania kodu Java. W przypadku zmian w modelu następuje generowanie kodu, edycja pliku XML oraz kompilacja wygenerowanego kodu. Cały mechanizm działa bardzo szybko i skutkuje wykrywaniem błędów Java już w momencie tworzenia diagramów.

StarUML

Program napisany jest w języku Delphi, istnieje jednak możliwość używania innych języków w celu rozszerzenia lub edycji możliwości programu. StarUML oferuje wsparcie dla UML 2.0. Posiada możliwość generowania kodu źródłowego dla języków programowania obiektowego takich jak: Java, C++ oraz C#. Niewątpliwym pozytywnym zaskoczenia jest fakt iż ten darmowy program umożliwia odczytu plików utworzonych w komercyjnym oprogramowaniu Rational Rose. Pliki tworzone w StarUML są zapisywane w formacie XML z rozszerzeniem .uml. Największą wadą tego programu jest brak wsparcia jakichkolwiek baz danych.

UMLet

UMLet powstał w 2002 roku, jest narzędziem umożliwiającym tworzenie diagramu przypadków użycia, obiektów, pakietów, klas, komunikacji, struktur połączonych, sekwencji, maszyny stanowej, wdrożenia i czynności. Jest to bardzo prosty program, pozwalający jednak użytkownikowi na tworzenie nowych elementów UML. Projektowanie polega na określeniu właściwości elementów, które są w formie tekstowej. Użytkownik powinien więc znać składnię programu, aby móc w pełni wykorzystać jego możliwości.

UML pad

UML pad umożliwia projektowanie tylko pięciu rodzajów diagramów: klas, przypadków użycia, sekwencji, maszyny stanowej oraz czynności. Ma możliwość eksportu dokumentów do formatu HTML. Ze względu na bardzo ograniczone możliwości jest raczej mało popularny.

UML Sculptor

UML Sculptor oferuje jedynie tworzenie diagramów klas. Nie posiada funkcji cofnij, ponów, a jego szata graficzna jest bardzo uboga. Ze względu na ograniczenie możliwości jest mało praktyczny.

Violet

Violet jest programem oferującym sześć diagramów UML: przypadków użycia, klas, sekwencji, obiektów, aktywności i maszyny stanowej. Nie posiada możliwości generowania kodu, ani wykrywania błędów. Jest prostą w obsłudze aplikacją skierowaną bardziej do osób nie potrzebujących rozbudowanych komercyjnych programów niż do studentów informatyki.

Software Ideas Modeler

Program oferuje szeroki wachlarz narzędzi do edycji elementów, co sprawia, że tworzone diagramy są bardzo estetyczne i czytelne. Stworzone schematy można wyeksportować do kilku formatów. Program umożliwia także wygenerowanie kodu źródłowego dla języków: Action Script, C++, C#, Java, JavaScript, PHP, Python, Ruby, SQL DDL, VB 6, VB.NET oraz XSD. W programie można pracować nad kilkoma projektami równocześnie, ponadto istnieje możliwość kopiowania elementów już stworzonych diagramów. Aplikacja posiada także opcję filtrowania diagramów i wyszukiwania słów kluczowych, a także rozbudowany generator dokumentacji HTML. Tworzenie diagramów w Software Ideas Modeler jest bardzo intuicyjne i ergonomiczne. Program jest darmowy dla uczelni w wersji Standard.

W tabeli 2 zestawiono zaprezentowane wcześniej programy. Zaprezentowano ich możliwości w zakresie rodzajów tworzonych diagramów UML, obsługi błędów popełnianych przez projektantów czy też studentów, możliwości generowania kodu źródłowego czy też możliwości eksportu projektów do innych użytecznych formatów. Tabela 2 zawiera także uwagi jakie nasuwają się podczas korzystania z tego oprogramowania.

Tab. 2. Zestawienie darmowego oprogramowania wspierającego UML

	Liczba diagramów możliwych do utworzenia	Wykrywanie błędów	Liczba rodzajów eksportów do innych formatów	Generowanie kodu źródłowego	Uwagi dotyczące działania i ergonomii
AmaterasUML	Klas, przypadków użycia, sekwencji, aktywności (4)	Nie pozwala na łączenie elementów, które do siebie nie pasują	ucd, sqd, cld, acd (4)	Java (1)	Brak pętli w diagramie sekwencji, nie ma możliwości podpisania bloków decyzyjnych. Na diagramie sekwencji wszystkie elementy są takie same (nie ma typów obiektów). Program jest trudny w obsłudze.
ArgoUML	Klas, przypadków użycia, sekwencji, komunikacji, aktywności, wdrożenia, maszyny stanowej 7	Istnieje pewne wykrywanie błędów, program sam dokonuje poprawek.. Program potrafi sam skasować część diagramu, jeśli uzna, że jest błędna.	Zargo, uml, zip (3)	C++, C#, Java, PHP4, PHP5, SQL (6)	Program mało intuicyjny, niestety nie ma żadnego podziału elementów. Nieintuicyjne przypisanie funkcji klawiszom ENTER i ESC. Diagramy są mało estetyczne i mało czytelne. W diagramie wdrożenia nie można wyróżnić poszczególnych elementów (wszystkie są takie same).
Dia	Przypadków użycia, klas, sekwencji, komunikacji, maszyny stanowej, czynności, komponentów, wdrożenia, pakietów, (8)	Nie ma wykrywania błędów	emf, png, pdf, ps, svg, wmf, cgm, dia, shape, eps, emf, code, fig, plt, hpigl, tex, mp, bmp, gif, hpeg, jpeg, jpg, tiff, tif, dxf, wdx, wpg (27)	Brak (0)	Wszystkie elementy (obiekty) diagramów UML-owych są wyświetlane razem w przyborniku. Dodatkowo program nie realizuje wykrywania błędów. Pomimo powyższych program należy zaliczyć do łatwych w użytkowaniu.
Fujaba	Klas, sterowania interakcją, przypadków użycia, pakietów, maszyny stanowej, aktywności (6)	Wykrywanie błędów Javy. Program nie pozwala na łączenie niepasujących ze sobą elementów	ctr (1)	Java (1)	Bardzo utrudnione, nieergonomiczne korzystanie z diagramu aktywności. Enigmatyczny przybornik obiektów rysowanych na diagramach.

	Liczba diagramów możliwych do utworzenia	Wykrywanie błędów	Liczba rodzajów eksportów w do innych formatów	Generowanie kodu źródłowego	Uwagi dotyczące działania i ergonomii
StarUML	Przypadków użycia, komponentów, klas, wdrożenia, kolaboracji, maszyny stanowej, czynności, struktur połączonych (8)	Blokuje pewne działania, ale nie ma objaśnień, ani czytelnych komunikatów. Posiada pewien zestaw narzędzi dla danego diagramu i nie można nic pomieszać.	jpg, jpeg, bmp, emf, wmf, xml (6)	C++, C#, Java. (3)	W diagramie klas program nie podpowiada typów klas – istnieje całkowita dowolność. W diagramie sekwencji jest bardzo trudno edytować powtórzenia i nie ma możliwości opisu ich warunków działania. W diagramie wdrożenia nie ma komponentów, są tylko pakiety, co jest mylące dla użytkownika. W diagramie struktur połączonych nie ma podstawowego elementu obrazującego struktury złożone. Program należy sklasyfikować jako skomplikowany i mało intuicyjny.
UMLet	Przypadków użycia, obiektów, pakietów, klas, komunikacji, struktur połączonych, sekwencji, maszyny stanowej, wdrożenia, czynności (9)	Nie ma żadnego wykrywania błędów.	Uxf, bmp, eps, gif, jpg, pdf, png, svg (8)	Brak (0)	Dane elementów są określane w sposób tekstowy, więc użytkownik musi znać składnię. Można zaprogramować własny typ elementów (Java). Tworzone diagramy są przejrzyste. Program bardzo prosty w obsłudze, intuicyjny.
UML pad	Klas, przypadków użycia, sekwencji, czynności (4)	Nie pozwala na błędne połączenie elementów, ale nie ma żadnych komunikatów	ucd, uud, usd, uss, uad (5)	Brak (0)	W diagramie sekwencji nie ma możliwości dodania żadnych pętli. Strzałki bardzo trudno jest przemieścić. Dość łatwy w obsłudze. Mało diagramów, nie można zapisać ich w jednym pliku.
UML Sculptor	Klas (1)	Jasne komunikaty, co się źle robi.	Uml (1)	Brak (0)	Program, który umożliwia tylko stworzenie diagramu klas. Nie ma funkcji cofnij, ponów. Diagramy tworzy się w łatwy sposób.

	Liczba diagramów możliwych do utworzenia	Wykrywanie błędów	Liczba rodzajów eksportów w do innych formatów	Generowanie kodu źródłowego	Uwagi dotyczące działania i ergonomii
Violet	Przypadków użycia, klas, obiektów, maszyny stanowej, czynności, sekwencji, (6)	Brak	violet, png, (2)	Brak (0)	W diagramie klas użytkownik może stworzyć kilka klas o tej samej nazwie, w podawaniu metod i atrybutów jest pełna dowolność, nie ma żadnych ograniczeń, ani sugestii. Nie można przesuwac strzałek (przejsć między stanami, czynnościami). W diagramie sekwencji wszystkie elementy są takie same, Nie ma żadnych pętli. Bardzo łatwy w użytkowaniu.
Software Ideas Modeler	Przypadków użycia, klas, pakietów, obiektów, komponentów, struktur złożonych, sekwencji, komunikacji, aktywności, maszyny stanowej, sterowania interakcją, wdrożenia, harmonogramowania, profili. (14)	Pełna kontrola programu nad poprawnością połączeń na diagramach.	png, jpg, gif, bmp, tiff, svg, wmf, emf, pdf, xmi, simp (11)	Action Script, C++, C#, Java, Java Script, PHP, Python, Ruby, SQL DDL, VB 6, XSD, VB.NET (12)	Program rozbudowany i intuicyjny w użytkowaniu. Diagramy bardzo estetyczne i czytelne. Mnogość diagramów, ale bez podziału na kategorie.

Zródło: [4]

3. KRYTERIA PORÓWNYWANIA PROGRAMÓW

Ocenę oprogramowania oparto na kryteriach punktowych: Punkty przydzielono programom w pięciu kategoriach.

Pierwsza oceniana kategoria to ergonomia pracy. Projektowanie wymaga od projektanta nieustannych zmian i poprawek w tworzonych przez niego diagramach. Istotne jest więc aby możliwość ich wprowadzania była łatwa i intuicyjna. Jest to jeden z subiektywnych punktów oceny. Został przeprowadzony przez dwóch ochotników na co dzień zajmujących się projektowaniem w UML.

Ocenę punktową dotyczącą ergonomii zdefiniowano następująco:

- 0 pkt. – mała intuicyjność i ergonomia,
- 1 pkt. – użytkowanie intuicyjne i ergonomiczne z pewnymi brakami,
- 2 pkt. – użytkowanie w pełni intuicyjne i ergonomiczne.

W przypadku nauczania i kształcenia istotnym elementem jest możliwość tworzenia wszystkich 14 typów diagramów UML dostępnych w specyfikacji 2.4. Ustalono więc następujące kryteria punktowe:

- 0 pkt. – 1 diagram,
- 1 pkt. – od 2 do 5 diagramów,
- 2 pkt. – od 6 do 10 diagramów,
- 3 pkt. – ponad 10 diagramów.

Kolejną przydatną funkcją dla dydaktyki jest śledzenie i poprawa błędów na projektowanych przez studentów diagramach. Jest to korekcja formalna a nie merytoryczna. Często informacje o błędach są enigmatyczne i niezrozumiałe dla projektanta. Zaproponowano następującą skalę punktową:

- 0 pkt. – brak wykrywania błędów,
- 1 pkt. – ograniczone bądź nieczytelne dla użytkownika wykrywanie,
- 2 pkt. – rozbudowane i czytelne dla użytkownika wykrywanie błędów.

Przydatną funkcjonalnością, pozytywnie podnoszącą wartość darmowego oprogramowania jest możliwość generowania kodu źródłowego współczesnych języków programowania. Zaproponowano tutaj następującą ocenę punktową:

- 0 pkt. – brak generowania kodu,
- 1 pkt. – do 3 języków programowania,
- 2 pkt. – ponad 3 języki programowania.

Warto jest także rozważyć ocenę możliwości eksportu danych do innych formatów:

- 0 pkt. – brak możliwości eksportu lub zapisu do innego formatu zapisu,
- 1 pkt. – do 5 formatów plików,
- 2 pkt. – ponad 5 formatów plików.

Analizie poddano także intuicyjność obsługi danego programu. Sposób przydzielania punktów zaproponowano następująco:

- 0 pkt. – mała intuicyjność obsługi, konieczność „szukania” podstawowych funkcji programu,
- 1 pkt. – użytkowanie intuicyjne ale z pewnymi brakami,
- 2 pkt. – użytkowanie w pełni intuicyjne.

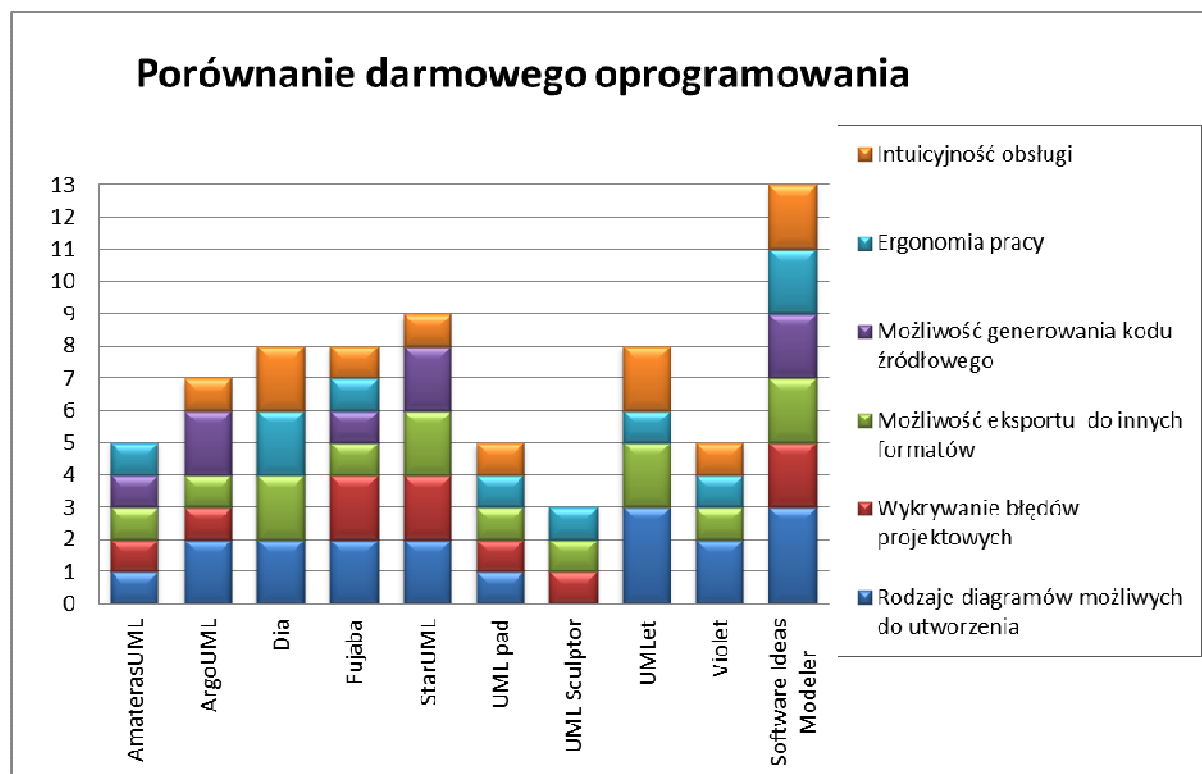
4. ANALIZA PORÓWNAWCZA OPROGRAMOWANIE WSPOMAGAJĄCEGO UML

W poniższej tabeli 3 dokonano przedstawienia wyników składowych analizy porównawczej.

Tab. 3. Analiza porównawcza darmowego oprogramowania wspierającego UML

Nazwa oprogramowania	Rodzaje diagramów możliwych do utworzenia	Wykrywanie błędów projektowych	Możliwość eksportu do innych formatów	Możliwość generowania kodu źródłowego	Ergonomia pracy	Intuicyjność obsługi	Suma punktów
AmaterasUML	1	1	1	1	1	0	5
ArgoUML	2	1	1	2	0	1	7
Dia	2	0	2	0	2	2	8
Fujaba	2	2	1	1	1	1	8
StarUML	2	2	2	2	0	1	9
UML pad	1	1	1	0	1	1	5
UML Sculptor	0	1	1	0	1	0	3
UMLet	3	0	2	0	1	2	8
Violet	2	0	1	0	1	1	5
Software Ideas Modeler	3	2	2	2	2	2	13

Źródło: [opracowanie własne]



Rys. 1. Porównanie darmowego oprogramowania wspomagającego projektowanie systemów informatycznych w UML.

Źródło: [opracowanie własne].

PODSUMOWANIE

Istnieje możliwość nauczania projektowania systemów informatycznych bazując jedynie na darmowym oprogramowaniu. Porównano dziesięć różnych darmowych dla uczelni programów. Analizując ich możliwości dokonano klasyfikacji. Najlepszym z testowanych bezpłatnych programów okazał się Software Ideas Modeler (wersja Standard). Charakteryzuje się on dużą liczbą oferowanych diagramów a także możliwością generowania kodu

źródłowego w kilku językach popularnych językach programowania. Program w bardzo wysokim stopniu kontroluje błędy użytkownika. Tworzone diagramy są bardzo estetyczne i czytelne. Program jest bardzo intuicyjny w użytkowaniu. W rankingu przyznano mu 13 punktów na 13 możliwych. Takim wynikiem mogą pochwalić się jedynie komercyjne programy takie jak Visual Paradigm czy IBM Rational Architect. Kolejną wysoką notę – 9 punktów przyznano programowi Star UML. Minimalnie niższe noty otrzymały programy: DIA, Fujaba i UMLet. Te programy wydają się posiadać większość funkcjonalności i możliwości przydatnych do nauczania UML.

BIBLIOGRAFIA

1. Berkenkötter K.: *Reliable UML Models and Profiles*, Electronic Notes in Theoretical Computer Science 217, 2008
2. Filipowski P., Szymczyk T., *Sposoby projektowania systemów informatycznych we współczesnych firmach*, praca magisterska, Politechnika Lubelska, 2011
3. Mrozek Z.: *Wprowadzenie do inżynierii oprogramowania i języka UML*, Abaton, Kraków, 2011
4. Rzeszutko A., Szymczyk T., *Oprogramowanie wspomagające projektowanie z wykorzystaniem UML – analiza porównawcza*, praca magisterska, Politechnika Lubelska, 2012
5. Schmuller J.: *UML dla każdego*, Helion, Gliwice, 2003
6. Subieta K.: *Język UML*, V Konferencja PLOUG, Zakopane, 1999

ANALYSIS OF FREE SOFTWARE SUPPORTING TEACHING OF OBJECT ORIENTED

Abstract

The article refers to the problem of teaching object-oriented design. Commercial software to design systems using UML is expensive. The alternative is so free software types: open source or freeware. This article reviews current programs, proposed benchmarks and made a comparative analysis.

Autor:

dr inż. Tomasz Szymczyk – Politechnika Lubelska, Instytut Informatyki,
e-mail t.szymczyk@pollub.pl