

Programowanie obiektowe — projekt

Prowadzący – mgr inż. Jakub Zgraja

Adres e-mail – jakub.zgraja@pwr.edu.pl

Konsultacje – D-20, p. 912, środa 9-10 – ze względu na jedną godzinę konsultacji zalecam wcześniejszą konsultację e-mailową

Materiały – <http://eportal.pwr.edu.pl>

1 Kontakt z prowadzącym

1. mail:
 - (a) używając skrzynki studenckiej (@student.pwr.edu.pl),
 - (b) temat powinien zawierać informacje o zajęciach oraz grupie (np. [P0] [PN19]; godzina grupy – zaokrąglenie do najbliższej pełnej godziny),
2. poza e-mailem, należy sprawdzać wiadomości na Edukacji.CL/JSOS (jednakże proszę tam nie wysyłać wiadomości),
3. podczas zajęć,
4. podczas konsultacji.

2 Dostęp do e-Portalu

1. <http://eportal.pwr.edu.pl>
2. Kursy → Kursy wydziałowe → Wydział Elektroniki → Programowanie → Programowanie obiektowe - projekt do wykładu JK
3. Hasło: P0_InicjałyProwadzącego_DzieńTygodniaGodzinaZajęć – zakładając, że zajęcia prowadzi Jan Kowalski, z programowania obiektowego, w czwartki o 9:15: P0_JK_cz9

3 Środowisko

1. C++ (obowiązujący standard to co najmniej C++98, zachęcam do używania C++11 lub C++14):
 - (a) Visual Studio¹,
 - (b) C-Lion²,
 - (c) Makefile/CMake (na plus!),
2. C#:

¹Microsoft Imagine

²<https://www.jetbrains.com/student/>

(a) Visual Studio¹,

3. Java:

(a) NetBeans,

(b) IntelliJ²,

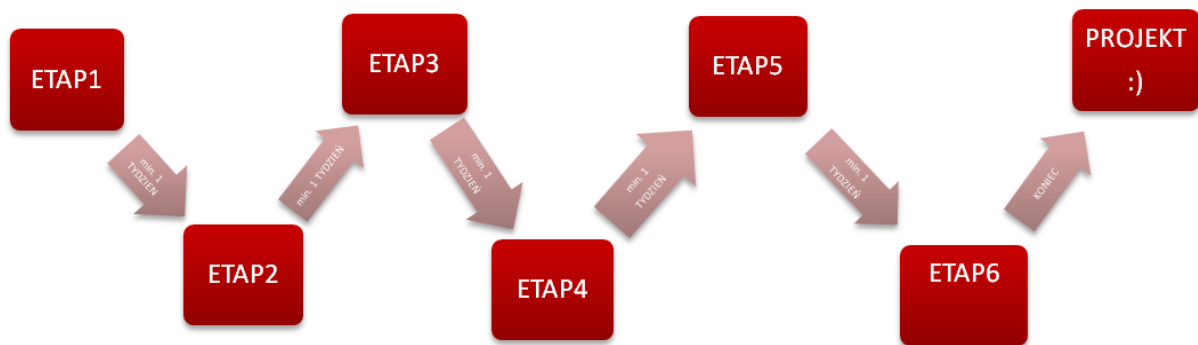
4 Realizacja projektu

Projekt może być realizowany w grupach (maksymalnie dwuosobowych). Poszczególne etapy powinny zostać zamieszczone na ePortalu, oraz zaprezentowane podczas zajęć, zgodnie z harmonogramem.

1. Zgłoszenie projektu (do **drugich** zajęć),
2. Zaprojektowanie szczegółowych założeń aplikacji – diagram przypadków użycia (UML) (do **trzecich** zajęć),
3. Zaprojektowanie aplikacji w UML – diagram klas oraz innych potrzebnych diagramów (do **piątych** zajęć),
4. Implementacja aplikacji w C++ (do **ósmych** zajęć),
5. Implementacja aplikacji w Javie/C# (do **dziesiątych** zajęć)
6. Wykonanie dokumentacji projektowej oraz zaprezentowanie wykonanego projektu (do **czternastych** zajęć).

	LUTY		MARZEC				KWIECIEŃ					MAJ				CZERWIEC				LIPIEC
PN	19	26	5	12	19	26 ptN	2	9	16	23	30	7	14	21	28	4 P	11 N	18	25	2
WT	20	27	6	13	20	27	3	10	17	24	1	8	15	22	29	5 P	12 N	19	26	3
ŚR	21	28	7	14	21	28	4	11	18	25	2	9	16	23	30 czN	6 P	13 N	20	27	4
CZ	22	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7 P	14 N	21	28	5
PT	23	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8 P	15 N	22	29	6
SO	24	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	7
N	25	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8
P - PARZYSTY N - NIEPARZYSTY	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N

Między etapami dozwolona jest karencja jednego tygodnia zajęć. Każdy tydzień opóźnienia, za dowolny etap powoduje obniżenie oceny końcowej o 0.5 stopnia! Na każdych zajęciach istnieje możliwość rozliczenia bieżącego, lub zaległych etapów projektu.



5 Wymagania projektowe

1. definiowanie klas $\rightarrow^3, 4$
2. hermetyzacja danych i metod $\rightarrow^3, 4$
3. dziedziczenie $\rightarrow^3, 4$
4. polimorfizm $\rightarrow^3, 4$
5. przeciążanie operatorów \rightarrow^3
6. GUI $\rightarrow^3, 4$

Oceniana również będzie czytelność kodu (proszę także pamiętać o komentowaniu kodu), oraz bezawaryjne działanie kodu.

Osobom ambitnym polecam używanie wzorców projektowych, stosowania zasad SOLID oraz stworzenia testów do kodu.

6 Warunki zaliczenia

1. obecność na zajęciach (terminowe rozliczenie etapów projektowych),
2. uzyskanie oceny pozytywnej z *realizacji* projektu (ocena \mathbf{F}_2) – maks. 100 pkt,
 - (a) F_{21} – ocena: trudności i złożoności (w tym funkcjonalności) realizowanego tematu, diagramów UML (w szczególności diagramu klas – ich hermetyzacji i dziedziczenia).
 - (b) F_{22} – ocena aplikacji w C++.
 - (c) F_{23} – ocena aplikacji w C# albo Java.
3. uzyskanie oceny pozytywnej z *prezentacji* projektu (ocena \mathbf{F}_3) – maks. 100 pkt.

³C++

⁴C#/Java

7 Ocena końcowa

Ocena z projektu: $0.75 * F_2 + 0.25 * F_3$

Punkty	Ocena końcowa
<0 ; 50)	2.0
<50 ; 60)	3.0
<60 ; 70)	3.5
<70 ; 80)	4.0
<80 ; 90)	4.5
<90 ; 97)	5.0
<97 ; 100>	5.5

8 Efekty kształcenia

Z zakresu wiedzy:

PEK_W01 Zna filozofię podejścia obiektowego

PEK_W02 Zna podejście obiektowe jako sposób pojmowania otaczającej rzeczywistości

PEK_W03 Zna podstawy zunifikowanego języka modelowania (UML)

PEK_W04 Zna podstawy inżynierii i metodologii programowania obiektowego

PEK_W05 Zna podstawowe narzędzia obiektowo zorientowanego języka programowania na przykładzie języka C++

Z zakresu umiejętności:

PEK_U01 Potrafi uzasadnić i stosować techniki obiektowe w programach.

PEK_U02 Potrafi konstruować kod modelujący zadany problem z wykorzystaniem hierarchii klas

PEK_U03 Potrafi konstruować i wykorzystywać związki pomiędzy obiektami w oparciu o polimorfizm

PEK_U04 Potrafi wykonać dokumentację kodu źródłowego

9 Literatura

1. Grębosz J., Symfonia C++ standard. Programowanie w języku C++ orientowane obiektowo, T1 i T2, Kraków, Oficyna Kallimach, 2015
2. Stroustrup B., Język C++, Kompendium wiedzy, Gliwice, Helion, 2014
3. Eckel B., Thinking in Java, Gliwice, Helion, 2009
4. Hejlsberg A. i inni, Język C#. Programowanie. Wyd. III, Microsoft .NET Development Series, Gliwice, Helion, 2010

5. Kisilewicz J., Język C++. Programowanie obiektowe, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 2005
6. Martin F., UML w kropelce, Warszawa, Oficyna Wydawnicza LTP, 2005
7. Martin J., Odell J.J., Podstawy metod obiektowych, WNT, 1997
8. Graham I., O'Callaghan A., Metody obiektowe w teorii i praktyce, WN-T, 2004
9. Miles R., Hamilton K., UML 2.0 wprowadzenie, Gliwice, Helion, 2007