

KARTA PRZEDMIOTU

Nazwa w języku polskim: **Projekt programistyczny**

Nazwa w języku angielskim: **Software development project**

Karta przedmiotu ważna od roku akademickiego: **2022/2023**

Kierunek studiów: **Informatyka**

Poziom studiów: **Studia I stopnia**

Forma studiów: **Niestacjonarne**

Profil: **Praktyczny**

Specjalność: **Cyberbezpieczeństwo, Systemy i sieci komputerowe, Informatyka przemysłowa**

Język wykładowy: **Polski**

Jednostka prowadząca: **Wydział Nauk Społecznych i Technicznych**

Prowadzący: **mgr inż. Arkadiusz Dębski**

OBCIĄŻENIE STUDENTA

	Wykład	Konwersatorium	Laboratorium	Projekt	Seminarium
Liczba godzin zajęć dydaktycznych organizowanych przez Uczelnię				18	
Liczba godzin całkowitego nakładu pracy studenta				75	
Forma zaliczenia				Zaliczenie z oceną	
Liczba punktów ECTS				3	

WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

Umiejętność dobierania algorytmów do rozwiązywania zadanego problemu.

Praktyczna znajomość jednego z wysokopoziomowych języków programowania (Java, Python, C++, C#...).

CELE PRZEDMIOTU

C1	Nabycie umiejętności stosowania metod programowania algorytmów „dziel i zwyciężaj”. Nabycie wiedzy związanej z wyszukiwaniem i zastosowaniem istniejących rozwiązań danego problemu.
C2	Wykształcenie umiejętności korzystania ze zintegrowanego środowiska deweloperskiego (IDE) oraz dostępnych w nich narzędzi.
C3	Nabycie umiejętności pracy nad projektem programistycznym w grupie.

PRZEDMIOTOWE EFEKTY UCZENIA – PEU	
Z zakresu wiedzy:	
PEU_W01	Student posiada specjalistyczną wiedzę dotyczącą rozwijania oprogramowania.
Z zakresu umiejętności:	
PEU_U01	Student potrafi wykorzystać narzędzia ułatwiające rozwój oprogramowania takie, jak środowisko deweloperskie i kontrola wersji.
PEU_U02	Student potrafi samodzielnie sformułować i zrealizować cel, który chce osiągnąć i dobrać narzędzia potrzebne do jego realizacji.
Z zakresu kompetencji społecznych:	
PEU_K01	Jest gotów współpracować w zespole, wykazuje się świadomością swojej roli w projekcie oraz dbałością o terminową realizację powierzonych zadań.

TREŚCI PROGRAMOWE		
Forma zajęć – projekt		Liczba godzin
P1	Wprowadzenie, zajęcia organizacyjne, omówienie dotychczasowych doświadczeń studentów.	1
P2	Prezentacja dostępnych rozwiązań w postaci środowisk deweloperskich, zasady wykorzystywania oprogramowania na licencjach.	1
P3	Propozycja zadań projektowych, omówienie przykładowych projektów.	1
P4	Zapoznanie studentów z narzędziami programistycznymi (IDE, debugger, kontrola wersji).	1
P5	Dyskusja, wybór zadań, prezentacja założeń.	2
P6	Realizacja projektu. Prezentacja poszczególnych etapów. Rozwiązywanie zaistniałych problemów.	10
P7	Prezentacja wyników, analiza stanu aplikacji.	1
P8	Podsumowanie. Zaliczenie przedmiotu.	1
Razem		18

STOSOWANE NARZĘDZIA DYDAKTYCZNE	
1.	Prezentacja treści z wykorzystaniem multimediów.
2.	Literatura dotycząca rozwoju oprogramowania.
3.	Dokumentacja techniczna języków programowania.
4.	Dokumentacja techniczna narzędzi programistycznych.

METODY I FORMY OCENY
OSIĄGNIĘCIA PRZEDMIOTOWYCH EFEKTÓW UCZENIA

Formy oceny (F lub P)*	Numer efektu uczenia (przedmiotowego)	Metody oceny osiągnięcia efektu uczenia
F p	PEU_W01, PEU_K01, PEU_U02, PEU_U01	Praca w grupach, dobieranie odpowiednich metod rozwiązywania problemu.
P p	PEU_W01, PEU_K01, PEU_U02, PEU_U01	Ocena założeń problemu, ocena stopnia realizacji danych założeń.

*F – ocena formująca (w trakcie semestru), P – ocena podsumowująca (na koniec semestru)

KRYTERIA OCENY
OSIĄGNIĘCIA PRZEDMIOTOWYCH EFEKTÓW UCZENIA

Nr PEU	Ocena dostateczna	Ocena dobra	Ocena bardzo dobra
PEU_W01	Student posiada podstawową wiedzę na temat rozwoju oprogramowania, potrafi wymienić paradygmaty programowania i różnicę pomiędzy nimi.	Student potrafi posiada pogłębioną wiedzę na temat rozwoju oprogramowania, potrafi dobrać odpowiednie narzędzia i rozumie ich zastosowanie.	Student posiada szczegółową wiedzę na temat rozwoju oprogramowania. Rozumie sposób działania rozwijanego oprogramowania i potrafi wytłumaczyć zasady działania wykorzystanych rozwiązań.
PEU_U01	Student potrafi korzystać z podstawowych funkcji środowiska deweloperskiego. Student zna podstawowe operacje związane z kontrolą wersji projektu.	Student potrafi korzystać z dodatkowych funkcji środowiska deweloperskiego. Student potrafi rozwiązywać problemy związane z kontrolą wersji.	Student potrafi korzystać z zaawansowanych funkcji środowiska deweloperskiego. Student wymienia zaawansowane operacje związane z kontrolą wersji.
PEU_U02	Student potrafi zaprogramować prostą aplikację korzystając z gotowych rozwiązań w postaci bibliotek.	Student potrafi zaprogramować złożoną aplikację korzystając z gotowych rozwiązań w postaci bibliotek.	Student potrafi zaprogramować zaawansowaną aplikację korzystając z istniejących bibliotek, a także potrafi samodzielnie rozwinąć takie biblioteki o dodatkowe funkcje.
PEU_K01	Student potrafi pracować w grupie. Rozumie swoją rolę w zadaniu.	Student potrafi pracować w grupie, wyrażać własne opinie i pomysły dotyczące realizacji projektu.	Student potrafi pracować w grupie, nie ma problemu z integracją swoich rozwiązań z resztą grupy, student umiejętnie rozdziela nakład pracy pomiędzy członków grupy.

LITERATURA PODSTAWOWA

Martin R.C., Czysty kod: Podręcznik dobrego programisty, Pearson, 2008.
Cormen T.H, Rivest R.L., Wprowadzenie do Algorytmów, MIT, 2009.

LITERATURA DODATKOWA

Dokumentacja techniczna wybranego języka programowania - dostępna na stronie języka.

MACIERZ POWIĄZANIA EFEKTÓW UCZENIA DLA PRZEDMIOTU PROJEKT PROGRAMISTYCZNY Z EFEKTAMI UCZENIA NA KIERUNKU INFORMATYKA

Przedmiotowy efekt uczenia	Odniesienie przedmiotowego efektu do efektów uczenia zdefiniowanych dla kierunku studiów i specjalności	Cele przedmiotu	Treści programowe	Numer narzędzia dydaktycznego
PEU_W01	K_W08	C1	P2	2, 3
PEU_U01	K_U07	C1	P2, P4, P6	1, 4
PEU_U02	K_U08	C2	P6	1, 2, 3, 4
PEU_K01	K_K03	C3	P6	4