

PROGRAM PRAKTYKI ZAWODOWEJ – Projektowanie, programowanie i testowanie aplikacji - 140 godzin (4 tygodnie)

Cele ogólne praktyki zawodowej:

1. Poznanie przepisów dotyczących bezpieczeństwa i higieny pracy;
2. Tworzenie programów z zastosowaniem obiektowości;
3. Tworzenie programu z zastosowaniem dziedziczenia;
4. Tworzenie aplikacji webowych z zastosowaniem MySQL, HTML, PHP, JavaScript;
5. Wykonywanie zaawansowanych zapytań SQL-owych w bazach danych;
6. Tworzenie aplikacji mobilnych;
7. Tworzenie aplikacji desktopowych;

Cele operacyjne

Uczeń potrafi:

1. stosować zasady bezpieczeństwa i higieny pracy, ochrony przeciwpożarowej oraz ochrony środowiska w miejscu pracy
2. organizować własne stanowisko pracy zgodnie z zasadami ergonomii
3. zaprezentować strukturę organizacyjną zakładu pracy oraz jego działalność informatyczną
4. dobrać konfigurację sprzętu i oprogramowania komputerowego odpowiednio do realizowanych zadań
5. posługiwać się językiem angielskim w zakresie dokumentacji technicznej
6. korzystać z gotowych pakietów oprogramowania użytkowego i narzędziowego
7. wykorzystywać lokalne sieci komputerowe oraz zasoby internetu w pracy programisty
8. projektować aplikacje zgodnie z wymaganiami użytkownika
9. programować aplikacje webowe, desktopowe i mobilne
10. obsługiwać zdarzenia i wyjątki w aplikacjach
11. integrować aplikacje z relacyjnymi bazami danych
12. wykorzystywać języki programowania wysokiego poziomu
13. stosować struktury danych i algorytmy w programowaniu
14. testować aplikacje pod kątem poprawności działania
15. wykonywać testy jednostkowe, integracyjne i funkcjonalne
16. analizować wyniki testów i usuwać błędy w aplikacjach
17. optymalizować kod źródłowy aplikacji
18. tworzyć interfejsy użytkownika zgodnie z zasadami ergonomii i dostępności
19. dokumentować proces projektowania, programowania i testowania aplikacji
20. korzystać z systemów kontroli wersji (np. Git) i repozytoriów kodu
21. współpracować w zespole projektowym
22. przestrzegać zasad kultury osobistej i norm zachowania w miejscu pracy
23. wykazywać odpowiedzialność za realizowane działania

- 24. stosować techniki twórczego myślenia w rozwiązywaniu problemów
- 25. doskonalić jakość wykonywanych działań
- 26. analizować i oceniać podejmowane działania
- 27. przestrzegać przepisów prawa pracy i zasad etyki zawodowej

MATERIAŁ NAUCZANIA Praktyka zawodowa – Projektowanie, programowanie i testowanie aplikacji.

Dział programowy	Wymagania programowe	
	Podstawowe Uczeń potrafi:	Ponadpodstawowe Uczeń potrafi:
Organizacja pracy w ramach praktyki zawodowej	– stosować przepisy bezpieczeństwa i higieny pracy oraz ochrony przeciwpożarowej – stosować regulamin praktyki – wymienić zasady oceniania w ramach praktyki zawodowej – stosować zasady organizacji stanowiska teleinformatycznego zgodnie z zasadami ergonomii. – stosować zasady współpracy w zespole	– określać warunki i organizację pracy zapewniające wymagany poziom ochrony zdrowia i życia przed zagrożeniami występującymi w środowisku pracy – stosować zasady współpracy w zespole

I. Tworzenie programów strukturalnych.	– posługiwać się pętlami, – zagnieżdżać pętle, – tworzyć proste typy danych oraz umie z nich korzystać, – korzystać z instrukcji warunkowych, – określić i zastosować instrukcję OR, AND, XOR, negacja, – wykonać operacje na bitach, – korzystać z funkcji wypisujących na ekran oraz wczytujących z klawiatury, – korzystać z tablic, – utworzyć wskaźnik i połączyć go ze zmienną, – poruszać się wskaźnikiem po tablicy, – uzyskać adres zmiennej, – wskaźnikiem przesyłać dane między funkcjami, – wymienić podstawowe funkcje które są w bibliotece matematycznej, – wymienić podstawowe funkcje które są w bibliotece String, – dołączyć bibliotekę do projektu, – pierwiastkować i potęgować, – łączyć ciągi napisów, – posługiwać się funkcjami szukającymi w ciągu, – posługiwać się funkcjami zmieniającymi wartości ciągów, – zapisać dane ze zmiennej do pliku tekstowego, – odczytać dane z pliku tekstowego do zmiennej – – utworzyć plik, – dopisywać dane do już istniejącego pliku,	– podzielić program na funkcje celem lepszej czytelności programu, – narysować schemat blokowy programu, – określić wielkość zmiennych, – określić co to są priorytety operatorów oraz umie podać ich zastosowanie, widzi zastosowanie dla wskaźników, – przesyłać tablice przez wskaźnik do funkcji, – zwracać wskaźnik z funkcji, – przekazywać do funkcji referencję, – podać różnicę między wskaźnikiem a referencją, – posługiwać się dokumentacją do biblioteki, – zapisać strukturę do pliku, – zabezpieczyć program przed błędami związanymi z otwieraniem i zapisywaniem do pliku,
II. Tworzenie programów z zastosowaniem obiektowości.	– podzielić zagadnienie na klasy, – definiować pola klas, – definiować metody klas, – określać zakres widoczności pól i metod klasy, – definiować kwalifikatory dostępu, – definiować konstruktory, w tym konstruktor kopiujący i destruktor klasy, – implementować funkcjonalność klasy, – deklarować obiekty, – budować hierarchię dziedziczenia klas w programie, – wydzielać metody i pola do odpowiednich klas w hierarchii dziedziczenia, – ukryć metody i pola, aby nie były widoczne w innych obiektach dziedziczących, – definiować klasy bazowe i pochodne, – stosować metody wirtualne, definiować klasy abstrakcyjne, – stosować szkielet obsługi wyjątków z instrukcjami try i catch, – stosować instrukcję throw, – opracować listę możliwych błędów wykonania aplikacji,	– planować aplikację z zastosowaniem hermetyzacji, dziedziczenia, polimorfizmu, – opisać obiekt na przykładzie z życia wziętym, – tworzyć obiekty statyczne i dynamiczne, – napisać przeciążenie konstruktora, – napisać przeciążoną metodę – zaprezentować dziedziczenie na przykładzie z życia wziętym – stworzyć szablon, – definiować obsługę dla błędów wykonania aplikacji, – wykorzystać wyjątki przy obsłudze plików,

III. Programowanie aplikacji okienkowych i mobilnych.	– zainstalować i skonfigurować środowisko programistyczne, – utworzyć nowy projekt programu, – stosować formatki typowe dla aplikacji desktopowych, – używać elementów interfejsu użytkownika np. okno, kontrolki, dialog modalny i niemodalny, – programować okna aplikacji, – programować system menu aplikacji, – programować okno dialogowe aplikacji, – programować obsługę zdarzeń myszy i klawiatury, – programować aplikacje w jednym z systemów mobilnych, – programować przechowywanie danych i preferencji użytkownika w aplikacjach mobilnych, – używać elementów UI takich jak: przyciski, okienka dialogowe, nawigacja, paski narzędziowe, grafika, dźwięk, – programować aplikacje, – uruchamiać aplikacje mobilne, – zastosować kontrolkę łączącą aplikację z bazą danych, – zapisywać i odczytywać dane z bazy danych – testować aplikacje, – testować zdarzenia, – testować kontrolki,	– dobrać środowisko programistyczne do określonych zadań, – doinstalować odpowiednie dodatki w środowisku programistycznym, – charakteryzować pojęcie framework, – korzystać z zaawansowanych zdarzeń, – skorzystać z zaawansowanych kontrolek, – stosować frameworki typowe dla aplikacji desktopowych np. WPF, – pobierać i wysyłać dane z lub do Internetu w aplikacjach mobilnych, – programować aplikację mobilną korzystającą z bazy danych, – wykonywać bardziej złożone aplikacje powiązane z bazą danych, – zabezpieczyć połączenie z bazą danych, aby było ono stabilne i odporne na błędy, – wykorzystać widoki z bazy danych, – wykonać dokumentację testowania, – testować połączenie z bazą danych, – przygotować aplikacje mobilne – opublikować aplikację w sklepie
IV Programowanie aplikacji webowych.	– tworzyć tabele, – stosować polecenia języka SQL, – wyszukiwać informacje w bazie danych przy użyciu języka SQL, – zmieniać rekordy w bazie danych przy użyciu języka SQL, – usuwać rekordy w bazie danych przy użyciu języka SQL, – zakładać użytkowników i nadawać im uprawnienia, – połączyć zapytania SQL wraz z innym językiem np. PHP, – zainstalować środowisko pracy, – skonfigurować środowisko pracy, – poruszać się po systemie, – napisać program w wybranym języku skryptowym, – uruchamiać napisany program, – znaleźć błędy w skryptach – implementować poznane algorytmy – stosować w programowaniu obsługę zdarzeń myszy i klawiatury – wykorzystać wybrane biblioteki, – stosować składnię oraz podstawowe polecenia, instrukcje i funkcje wybranych języków skryptowych, – korzystać z funkcji do obsługi ciasteczek,	– definiować tabele w bazie danych na podstawie projektu, – tworzyć skrypty w strukturalnym języku zapytań, – kontrolować spójność bazy danych, – tworzyć kopię zapasową bazy danych, – przywracać dane z kopii zapasowej, – importować i eksportować tabele bazy danych, – diagnozować i naprawiać bazę danych, – analizować problemy programistyczne, – czytać dokumentację do zaawansowanych funkcji – stosować kaskadowość stylów, – wykonywać responsywne strony internetowe, – stosować zasady dostępności WCAG – posługiwać się narzędziami „Dla twórców witryn” które są wbudowane w przeglądarkę internetową, – podać różnicę między programem kompilowanym a skryptem, – przepisać program z wersji kompilowanej na skrypt – oszacować wynik skryptu, – testować skrypt, – przekazać dane wynikowe z wyjścia jednego programu do wejścia drugiego,

	– korzystać z funkcji do obsługi plików, – osadzić skrypt napisany w PHP na stronie internetowej, – odczytać dane z bazy danych, – korzystać ze standardów i dokumentów, – stosować znaczniki języka HTML – definiować strukturę dokumentu hipertekstowego korzystając ze znaczników sekcji, – definiować hierarchię treści stosując znaczniki nagłówków i paragrafów, – definiować elementy strony internetowej: listy, tabele, obrazy, odnośniki, kontrolki, – wykonywać formularze na stronie internetowej, – projektować wygląd strony internetowej przy wykorzystaniu języka CSS, – osadzić skrypt napisany w JavaScript na stronie internetowej, – rozróżniać podstawowe pojęcia dotyczące grafiki, – dobierać oprogramowanie do obróbki grafiki komputerowej, – identyfikować różne formaty plików graficznych, – stosować różne modele barw, – korzystać z edytora grafiki wektorowej, – korzystać z edytora grafiki rastrowej, – wykonywać edycję plików graficznych na potrzeby stron internetowych, – projektować elementy graficzne dla strony internetowej, – osadzać tekst na grafice, dobierać jego krój i styl, – osadzać na stronie internetowej swoje grafiki, – dobierać oprogramowanie do edycji obrazu ruchomego i dźwięku, – osadzać elementy multimedialne na stronie internetowej, – dobierać oprogramowania do edycji, – stosować podstawy rysunku 2D i 3D – stosować podstawowe zasady rysowania, – stosować podstawy wymiarowania – stosować podstawowe funkcje, – stosować zasady oświetlania tekstury, – stosować zasady nakładania tekstury, – zainstalować CMS, – korzystać z panelu administratora, – administrować systemem zarządzania treścią, – wykorzystywać gotowe szablony, – uzupełniać portal treścią np. dodawać artykuły,	– znaleźć różnicę między grafiką rastrową a wektorową, – korzystać z zaawansowanych funkcji graficznych, – działać na warstwach, – określać zasady komputerowego przetwarzania wideo i dźwięku przygotowanego na potrzeby strony internetowej, – importować materiały multimedialne do systemów CMS, – wykonać animacji na potrzeby strony internetowej, – wykonać materiały wideo na potrzeby strony internetowej, – edytować wideo i dźwięk na potrzeby strony internetowej – używać zaawansowanych funkcji do edycji wideo oraz dźwięku, – zastosować zaawansowane funkcje, – zobrazować przestrzennie bryłę, – aktualizować system CMS, – projektować strony internetowe przy wykorzystaniu systemów CMS,
--	--	---

V Podstawowe algorytmy	– podać podstawy teoretyczne działania algorytmów sortowania, – charakteryzować algorytmy sortowania i ich złożoność obliczeniową, – stosować różne algorytmy sortowania, np. bąbelkowe, zachłanne, przez wstawienie, szybkie, metodą dzieli i zwyciężaj, – charakteryzować podstawy teoretyczne działania algorytmów wyszukiwania, – stosować algorytmy wyszukiwania dla tablic, list, kolejek, stosów, – zaimplementować program z rekurencją, – wskazać algorytmy, w których można zaimplementować rekurencję,	– określić złożoność obliczeniową algorytmów, – przepisać program i zaimplementować w nim rekurencję
------------------------	---	--

Kompetencje personalne i społeczne:

- stosuje zasady kultury osobistej i ogólnie przyjęte normy zachowania w środowisku pracy
- przyjmuje odpowiedzialność za powierzone informacje zawodowe
- przestrzega tajemnicy związanej z wykonywanym zawodem i miejscem pracy
- wyjaśnia, na czym polega zachowanie etyczne w zawodzie
- wskazuje przykłady zachowań etycznych w zawodzie
- omawia czynności realizowane w ramach czasu pracy
- określa czas realizacji zadań
- realizuje działania w wyznaczonym czasie
- monitoruje realizację zaplanowanych działań
- określa zakres umiejętności i kompetencji niezbędnych w wykonywaniu zadań zawodowych
- analizuje własne kompetencje
- wyznacza własne cele rozwoju zawodowego
- identyfikuje sygnały werbalne i niewerbalne
- stosuje aktywne metody słuchania
- prowadzi dyskusje
- udziela informacji zwrotnej
- dokonuje modyfikacji zaplanowanych działań
- dokonuje samooceny wykonanej pracy
- przewiduje skutki podejmowanych działań, w tym skutki prawne
- wykazuje świadomość odpowiedzialności za wykonywaną pracę
- ocenia podejmowane działania
- przewiduje konsekwencje niewłaściwego wykonywania czynności zawodowych na stanowisku pracy, w tym niewłaściwej eksploatacji maszyn i urządzeń na stanowisku pracy
- rozróżnia techniki rozwiązywania konfliktów związanych z wykonywaniem zadań zawodowych
- opisuje sposób przeciwdziałania problemom w zespole realizującym zadania
- pracuje w zespole, ponosząc odpowiedzialność za wspólnie realizowane zadania
- przestrzega podziału ról, zadań i odpowiedzialności w zespole
- angażuje się w realizację wspólnych działań zespołu
- modyfikuje sposób zachowania, uwzględniając stanowisko wypracowane wspólnie z innymi członkami zespołu

Program nauczania należy traktować w sposób elastyczny i dopasować do specyfiki przedsiębiorstwa mając na uwadze realizację założonych celów.