



**Fundusze
Europejskie**
Wiedza Edukacja Rozwój



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz Społeczny



PROGRAM NAUCZANIA

KURSU UMIEJĘTNOŚCI ZAWODOWYCH

INF.04.3 Projektowanie oprogramowania

w zakresie kwalifikacji

INF.04 Projektowanie, programowanie i testowanie aplikacji

wyodrębnionej w zawodzie

technik programista 351406

Branża teleinformatyczna (INF)

Autorzy:

mgr inż. Piotr Golonko

mgr Robert Fleischer

Recenzenci:

Recenzent 1 – Recenzja dydaktyczna (nauczyciel uczący w zawodzie, w którym wyodrębniono daną kwalifikację) **mgr inż. Marek Józwiak**

Recenzent 2 – Recenzja merytoryczna (przedstawiciel pracodawców właściwy dla danego zawodu) **mgr Adam Mazgajczyk**

Ekspert:

mgr inż. Grażyna Mrozińska-Hotłoś

Program opracowany we współpracy podmiotów z otoczenia społeczno-gospodarczego wskazanego we wniosku o powierzenie grantu na opracowanie modelowego kwalifikacyjnego kursu zawodowego (KKZ):

DGA S.A. (Partner Wiodący) z Gminą Miastem Toruń (Partner) reprezentowaną przez Toruński Ośrodek Doradztwa Metodycznego i Doskonalenia Osoba prowadząca kursy z Torunia przy współpracy z Edukacja i Kształcenie Zawodowe. EKZ. podmiotami otoczenia społeczno-gospodarczego szkół lub placówek systemu oświaty prowadzących kształcenie zawodowe.

Program Operacyjny Wiedza Edukacja Rozwój

Oś priorytetowa II

Efektywne polityki publiczne dla rynku pracy, gospodarki i edukacji

Działanie 2.14. Rozwój narzędzi dla uczenia się przez całe życie

Konkurs nr POWR.02.14.00-IP.02-00-003/19

Opracowanie modelowych programów kwalifikacyjnych kursów zawodowych (kkz)

Spis treści

PROGRAM NAUCZANIA kursu umiejętności zawodowych INF.04.3 Projektowanie oprogramowania.....	4
1. Wprowadzenie	4
2. Plan zajęć kursu umiejętności zawodowych	9
2.1. Pogrupowanie efektów kształcenia	9
2.2. Określenie liczby godzin na kształcenie zawodowe	17
2.3. Plan kursu umiejętności zawodowych	20
3. Cele kształcenia kursu umiejętności zawodowych	21
4. Programy poszczególnych zajęć	21
4.1. Program nauczania dla przedmiotu: Tworzenie aplikacji.....	21
4.1.1 Cele ogólne przedmiotu to:	21
4.1.2 Cele szczegółowe przedmiotu (słuchacz/uczestnik potrafi):	21
4.1.3 Materiał nauczania z uwzględnieniem opisu efektów kształcenia.....	22
4.1.4 Procedury osiągania celów kształcenia	25
4.1.5 Proponowane metody sprawdzania osiągnięć edukacyjnych słuchacza/uczestnika	26
5. Ewaluacja programu kursu dodatkowych umiejętności zawodowych.....	27
6. Wykaz literatury oraz niezbędnych środków i materiałów dydaktycznych.....	28
6.1. Wykaz literatury	28
6.2. Wykaz niezbędnych środków i materiałów dydaktycznych.....	30
7. Sposób i forma zaliczenia kursu	31
8. Sprawdzenie kompletności i poprawności opracowanego programu zajęć.....	32

PROGRAM NAUCZANIA kursu umiejętności zawodowych INF.04.3 Projektowanie oprogramowania

1. Wprowadzenie

Charakterystyka kursu umiejętności zawodowych

Kurs może być prowadzony przez:

- publiczne i niepubliczne szkoły prowadzące kształcenie zawodowe, z wyjątkiem szkół artystycznych – w zakresie zawodów, w których kształcą, oraz w zakresie innych zawodów przypisanych do branż, do których należą zawody, w których kształci szkoła,
- publiczne i niepubliczne placówki kształcenia ustawicznego i centra kształcenia zawodowego,
- instytucje rynku pracy, o których mowa w art. 6 ustawy z dnia 20 kwietnia 2004 r. o promocji zatrudnienia i instytucjach rynku pracy, prowadzące działalność edukacyjno-szkoleniową,
- podmioty prowadzące działalność oświatową, o której mowa w art. 170 ust. 2, posiadające akredytację, o której mowa w art. 118. ustawy z dnia 14 grudnia 2016 r. Prawo oświatowe (Dz. U. z 2019 r. poz. 1148, z późn. zm.).

Kurs może być realizowany w formie:

- stacjonarnej – w czasie około 6 miesięcy (1 semestr), łącznie 75 godzin – zajęcia odbywają się 3 lub 4 dni w tygodniu po min. 8 godzin dziennie,
- zaocznej – w czasie około 6 miesięcy (65% z 75 godzin = 50 godzin) – zajęcia odbywają się co 2 tygodnie przez 2 dni po 10 godzin dziennie, a w uzasadnionych przypadkach – co tydzień przez 2 dni po 10 godzin dziennie.
- efekty kształcenia wskazane do realizacji w kształceniu teoretycznym mogą być (po spełnieniu wymagań określonych w aktualnych przepisach oświatowych) realizowane w formie kształcenia na odległość, przy czym zaliczenie tych zajęć nie może odbywać się w formie zdalnej. Przedmioty o kształceniu praktycznym ze względu na efekty uczenia wymagające fizycznej interakcji muszą być prowadzone w formie stacjonarnej.

Informacje dodatkowe:

- Kurs jest prowadzony na poziomie IV Polskiej Ramy Kwalifikacji
- Kurs nie jest związany z szczególnymi uwarunkowaniami związanymi z kształceniem w kwalifikacji.
- Ośrodek prowadzący kurs ma obowiązek zgłoszenia okręgowej komisji egzaminacyjnej informacji o jego rozpoczęciu (w przeciągu 14 dni od rozpoczęcia realizacji KKZ).

Kurs jest przeznaczony dla osób chcących:

- zdobyć nowy zawód
- uzupełnić swoje wykształcenie
- udoskonalić swoje umiejętności
- wspomóc rozwój swojej kariery zawodowej
- zwiększyć szanse na znalezienie pracy
- dokonać zmiany pracy
- uzyskać awans zawodowy

- utrzymać zatrudnienie.

Wymagania wstępne:

- pozytywne przejście badań lekarskich (brak przeciwwskazań lekarskich do odbycia kursu)

Sposób i forma zaliczenia kursu:

- Kwalifikacyjny kurs zawodowy kończy się zaliczeniem w formie walidacji osiągnięć uczestnika kursu, polegającej na ocenie wykonywanych w trakcie nauki projektów i ćwiczeń oraz na podstawie uzyskanych w trakcie kursu ocen z poszczególnych przedmiotów.
- Do oceny osiągnięć edukacyjnych słuchaczy proponuje się stosowanie testów wielokrotnego wyboru, zadań z luką, ocenę aktywności słuchacza podczas wykonywania zadań w grupie, ocenę jakości wykonania zadań przez słuchacza. Proponuje się, aby osiągnięcia słuchaczy oceniać w zakresie zaplanowanych, uszczegółowionych celów kształcenia na podstawie:
 - obserwacji wykonanych ćwiczeń,
 - testu pisemnego.
- Umiejętności praktyczne proponuje się sprawdzać na podstawie obserwacji czynności wykonywanych przez słuchacza w trakcie realizacji ćwiczeń. Podczas obserwacji należy zwrócić uwagę na:
 - wyszukiwanie i przetwarzanie rzetelnych informacji pozyskanych z różnych źródeł,
 - poprawność merytoryczną wykonanych ćwiczeń praktycznych,
 - umiejętność pracy w zespole.
- Ważne kryteria oceny efektów kształcenia to: zaplanowanie wykonania zadania, dobór elementów oraz sporządzona dokumentacja techniczna. Możliwe są również inne sposoby i formy zaliczenia, takie jak: testy praktyczne, wykonanie projektów, próby pracy, aktywność uczącego się na zajęciach, prezentacje na forum grupy z przeprowadzonych prac.
- Osoba, która uzyskała zaliczenie, otrzymuje zaświadczenie o ukończeniu kursu dodatkowych umiejętności zawodowych.
- Kurs musi się zakończyć co najmniej 6 tygodni przed planowanym egzaminem z kwalifikacji zawodowej.

Założenia programowe

Głównym celem kształcenia w technik programista jest przygotowanie szeroko wykwalifikowanej kadry specjalistów. Przygotowanych do:

- profesjonalnego i rzetelnego wykonywania czynności zawodowych,
- pracy w ciągle zmieniającej się rzeczywistości zawodowej,
- samodzielnego podnoszenie swoich kwalifikacji,
- podejmowania własnej działalności gospodarczej zgodnej z zawodem,
- pracy w zespole,
- szybkiej aktualizacji wiedzy z niezwykle dynamicznej dziedziny, jaką jest branża teleinformatyczna,
- kontynuowania edukacji w szkołach wyższych na kierunkach: elektronika, teleinformatyka, telekomunikacja, informatyka lub zbliżonych.

Struktura programu

Program nauczania jest o strukturze przedmiotowej i spiralnej w układzie treści, z układem materiału nauczania zaczynającym się od zagadnień najprostszych po trudniejsze. Taki układ umożliwia powrót do treści zrealizowanych na początku edukacji, aby je powtórzyć i poszerzyć w kolejnych latach nauki. Utrwala to zarówno wiedzę jak i nabywane umiejętności celem przygotowania do realizacji zadań zawodowych. Dodatkowo taki układ i cykl nauczania w znaczącym stopniu niweluje braki edukacyjne, oraz pozwala na analizę materiału nauczania przez słuchaczy na różnych poziomach umiejętności.

Cele kierunkowe

Absolwent kursu powinien być przygotowany do wykonywania następujących zadań zawodowych w zakresie:

- znajomości i stosowanie różnych typów danych,
- znajomości i stosowanie różnych algorytmów,
- umiejętności tworzenia projektu aplikacji na podstawie wymagań klienta,
- planowania prac projektowych,
- licencjonowania.

Rozkład treści nauczania uwzględnia wzajemną korelację pomiędzy przedmiotami, a kolejność zdobywania wiedzy i umiejętności pozwala na nabycie wiedzy teoretycznej, by w krótkim czasie wykorzystać ją praktycznie. Zajęcia są realizowane na przedmiotach kształcenia teoretycznego oraz praktycznego. Dzięki czemu po zakończeniu kursu absolwent potrafi:

- stosować różne typy danych,
- stosować algorytmy,
- stosować narzędzia wspomagające projektowanie aplikacji,
- współpracować z klientem,
- opracowywać funkcjonalność aplikacji,
- projektować aplikację,
- tworzyć harmonogramy pracy,
- określać zasady licencjonowania,
- planować system zabezpieczeń licencyjnych w aplikacjach.

Opis branży

Branża teleinformatyczna będąc jedną z wielu branż przyszłości składa się z szeregu zawodów ukierunkowanych na świadczenie usług związanych z budową i eksploatacją systemów teleinformatycznych. Branża ta obejmuje szeroki zakres umiejętności i zawodów. Od umiejętności monterskich związanych z mechanicznym i elektrycznym montażem kabli i urządzeń poprzez ich konfigurację i eksploatację aż po tworzenie programów, zarządzania infrastrukturą informatyczną czy też bazami danych. W części tych zawodów bardzo ważne są kompetencje społeczne zwłaszcza w pracy z osobami niepełnosprawnymi. Branża cechuje się dużą dynamiką rozwoju i ciągłym zapotrzebowaniem na pracowników spowodowanym ciągłym i dynamicznym rozwojem usług i technologii przesyła i przetwarzania informacji.

Odniesienie do rynku pracy

Zawód technik programista 351406, który powstał w oparciu o rekomendacje pracodawców zgłaszających zapotrzebowanie na wykwalifikowanych kandydatów do pracy. Jest nowoczesnym i wymagającym zawodem przyszłości, stawiającym ciągle nowe wyzwania i dającym możliwości samorealizacji i dużej satysfakcji z wykonywanej pracy. Branż teleinformatyczna dynamicznie się rozwija dając możliwość samorozwoju i zapewniając liczne miejsca pracy. W związku z tym istnieje zapotrzebowanie na osoby wykwalifikowane w tym zawodzie. Pracodawcy oczekują absolwenta wyposażonego w wiele kluczowych umiejętności i potrafiącego szybko reagować na zmieniającą się rzeczywistość oraz pogłębiać swoją wiedzę i umiejętności w zakresie nowych rozwiązań konstrukcyjnych i technologii.

Głównym celem kształcenia w zawodzie technik programista jest przygotowanie wykwalifikowanej kadry specjalistów do pracy w dynamicznie zmieniającym się sektorze gospodarki jakim jest branża teleinformatyczna.

Osoba posiadająca wykształcenie w tym zawodzie może pracować:

- przedsiębiorstwa zajmujące się tworzeniem programów desktopowych i aplikacji internetowych;
- przedsiębiorstwa zajmujące się tworzeniem aplikacji mobilnych;
- przedsiębiorstwa zajmujące się projektowaniem UI;
- przedsiębiorstwa zajmujące się modelowaniem, projektowaniem i drukiem 3D.

Poza swoją branżą technik programista 351406 może pracować na stanowiskach związanych z montażem elektrycznym, elektronicznym i mechanicznym sieci i urządzeń teletransmisyjnych związanych z szerokopasmową transmisją danych oraz wykonywaniem instalacji urządzeń teletransmisyjnych i ich uruchamianiem. Do podjęcia pracy w tym zawodzie niezbędna jest wysoka sprawność manualna i dobra koordynacja wzrokowo - ruchowa.

Pracodawcy powinni aktywnie uczestniczyć w cyklu kształcenia poprzez:

- konferencje branżowe,
- seminaria,
- lekcje problemowe,
- lekcje pokazowe,
- praktyki zawodowe,

organizowane wspólnie z szkołą.

Powiązanie z zawodami

Branża teleinformatyczna pokrywa znaczny zakres usług, technik programista 351406 świadczy usługi z pogranicza telekomunikacji i informatyki, które są ukierunkowane na montaż, uruchamianie i utrzymanie sieci teleinformatycznych oraz instalowanie i administrowanie sieciowymi systemami komputerami. Umiejętności praktyczne są wzbogacone wiedzą ogólną z zakresu techniki analogowej i cyfrowej oraz wiedzą szczegółową z zakresu systemów komputerowych, sieciowych systemów operacyjnych, lokalnych sieci komputerowych, urządzeń sieciowych, systemów transmisji danych i systemów komutacyjnych.

Zawód technik programista wiąże się z szeregiem zawodów o zbliżonej tematyce, zakresie wiedzy i umiejętności. Umożliwia to zarówno pogłębianie wiedzy jak i łatwe przebranżowienie się na jeden z zawodów pokrewnych:

- technik teleinformatyk,
- technik informatyk,

- technik szerokopasmowej komunikacji elektronicznej,
- technik telekomunikacji,
- monter sieci i urządzeń telekomunikacyjnych,
- technik tyfłoinformatyk.

Dodatkowo technik programista może też się sprawdzić w zawodach z branży związanej z elektroniką takich jak:

- elektronik,
- technik elektronik.

Informacje o pozostałych Kursach Umiejętności Zawodowych

Istnieje możliwość uzupełnienia wiedzy i umiejętności zawodowych poprzez udział w dodatkowych Kursach Umiejętności Za dedykowanych dla zawodu technik programista 351406 (opisanych w odrębnych plikach):

- Bezpieczeństwo sieci komputerowych;
- Bezpieczeństwo systemów komputerowych;
- Budowa i konfiguracja sieci komputerowych;
- Eksploatacja baz danych;
- Grafika 3D i wydruk 3D;
- Programowanie mikrokontrolerów i prostych układów scalonych;
- Programowanie w języku Python;
- Serwis urządzeń techniki komputerowej;
- Tworzenie i testowanie aplikacji.

Po ukończeniu KUZ absolwent może realizować program kwalifikacyjnego kursu zawodowego INF.04 Projektowanie, programowanie i testowanie aplikacji opartego o podstawę programową kształcenia branżowego w zawodzie technik programista 351406, w której to wyodrębniono dla kwalifikacji INF.04 Projektowanie, programowanie i testowanie aplikacji następujące jednostki efektów kształcenia:

- INF.04.1. Bezpieczeństwo i higiena pracy,
- INF.04.2. Podstawy informatyki,
- INF.04.3. Projektowanie oprogramowania,
- INF.04.4. Programowanie obiektowe,
- INF.04.5. Programowanie aplikacji desktopowych,
- INF.04.6. Programowanie aplikacji mobilnych,
- INF.04.7. Programowanie aplikacji zaawansowanych webowych,
- INF.04.8. Testowanie i dokumentowanie aplikacji,
- INF.04.9. Język obcy zawodowy.

oraz efekty kształcenia realizowane na wszystkich obowiązkowych zajęć edukacyjnych z zakresu kształcenia zawodowego związane z nabywaniem kompetencji personalnych i społecznych i organizacji pracy małych zespołów, zgrupowane w jednostkach efektów kształcenia:

- INF.04.10. Kompetencje personalne i społeczne,
- INF.04.11. Organizacja pracy małych zespołów

Po ukończeniu KUZ absolwent może realizować programy kursów umiejętności zawodowych w obrębie kwalifikacji INF.04 Projektowanie, programowanie i testowanie aplikacji opartego o podstawę programową kształcenia branżowego w zawodzie technik programista 351406, w której to wyodrębniono następujące jednostki efektów kształcenia (opisanych w odrębnych plikach):

- INF.04.1. Bezpieczeństwo i higiena pracy,
- INF.04.2. Podstawy informatyki,
- INF.04.4. Programowanie obiektowe,
- INF.04.5. Programowanie aplikacji desktopowych,
- INF.04.6. Programowanie aplikacji mobilnych,
- INF.04.7. Programowanie aplikacji zaawansowanych webowych,
- INF.04.8. Testowanie i dokumentowanie aplikacji,
- INF.04.9. Język obcy zawodowy.

2. Plan zajęć kursu umiejętności zawodowych

2.1. Pogrupowanie efektów kształcenia

Tabela 1. Przyporządkowanie efektów kształcenia wraz z kryteriami weryfikacji do poszczególnych przedmiotów.

Efekty kształcenia Stopniowane efektów kształcenia efekt kluczowy ek,, efekt ważny ew, efekt pomocniczy ep	Liczba godzin na efekt kształcenia	Kryteria weryfikacji	Tworzenie aplikacji
A	B	C	D
1) posługuje się prostymi typami danych (ek)	4	1) rozróżnia typy numeryczne stałoprzecinkowe i zmiennoprzecinkowe	x
		2) rozpoznaje typ logiczny	x



Efekty kształcenia Stopniowane efektów kształcenia efekt kluczowy ek,, efekt ważny ew, efekt pomocniczy ep	Liczba godzin na efekt kształcenia	Kryteria weryfikacji	Tworzenie aplikacji
A	B	C	D
2) posługuje się złożonymi typami danych (ek)	6	3) rozróżnia typy znakowe i łańcuchowe	x
		4) posługuje się typami liczbowymi stałoprzecinkowe i zmiennoprzecinkowymi, typem logicznym, typem znakowym i łańcuchowym	x
		1) rozróżnia rodzaje złożonych typów danych	x
		2) posługuje się tablicami jednowymiarowymi i dwuwymiarowymi	x
		3) posługuje się tablicami dynamicznymi, asocjacyjnymi	x
		4) posługuje się typem rekordowym, np. struktura, unia	x
		5) posługuje się typem plikowym	x
		6) posługuje się typem wskaźnikowym	x
		7) charakteryzuje cechy kolekcji, w tym znaczenie iteratora	x
		8) posługuje się kolekcjami, np. listami, kolejkami, stosami, wektorami	x
		9) projektuje zestawy danych dla problemu programistycznego	x
3) stosuje metody rozwiązywania problemów za pomocą algorytmów (ek)	15	1) projektuje algorytmy za pomocą różnych metod: schematów blokowych, listy kroków, drzew decyzyjnych, pseudokodu	x
		2) charakteryzuje algorytmy iteracyjne, tekstowe i szyfrowania, tablicowe	x
		3) charakteryzuje algorytmy rekurencyjne	x
		4) charakteryzuje problemy i metody ich rozwiązywania, np. algorytmy heurystyczne, problem komiwojażera	x
		5) określa złożoność obliczeniową algorytmów	x
		1) charakteryzuje typy sortowania i ich złożoność obliczeniową	x
		2) stosuje różne typy sortowania, np. bąbelkowe, zachłanne, przez wstawianie, szybkie, metodą dziel i zwyciężaj	x



Efekty kształcenia Stopniowane efektów kształcenia efekt kluczowy ek,, efekt ważny ew, efekt pomocniczy ep	Liczba godzin na efekt kształcenia	Kryteria weryfikacji	Tworzenie aplikacji
A	B	C	D
		3) stosuje algorytmy wyszukiwania dla tablic, list, kolejek, stosów	x
5) dobiera narzędzia i metodologie do planowania i zarządzania projektem (ew)	7	1) określa funkcje narzędzi do zarządzania projektem	x
		2) stosuje diagramy do zarządzania etapami projektu, zadaniami i czasem, np. diagram Gantta	x
		3) korzysta z programów wspierających zarządzanie projektami, np. Jira, Trello	x
		4) korzysta z systemu kontroli wersji, np. Git	x
6) projektuje aplikację (ek)	15	1) analizuje wymagania klienta i tworzy zgodny z nimi projekt	x
		2) tłumaczy wymagania klienta na specyfikację techniczną dla zespołu programistów	x
		3) identyfikuje elementy interfejsu użytkownika, np. okna, dialogi, kontrolki, formularze, paski narzędziowe, widgety	x
		4) projektuje interfejs użytkownika i wygląd aplikacji	x
		5) dostosowuje interfejs do różnych platform	x
		6) projektuje aplikacje w różnych paradygmatach programowania: strukturalnym, obiektowym	x
		7) projektuje aplikację opartą na architekturze klient-serwer	x
		8) projektuje struktury danych dla aplikacji	x
		9) projektuje funkcjonalność aplikacji	x
		10) planuje system zabezpieczeń aplikacji	x
7) planuje przedsięwzięcie programistyczne (ew)	20	1) określa cel projektu	x
		2) określa fazy realizacji projektu	x
		3) charakteryzuje cykl życia projektu informatycznego i jego poszczególne etapy	x
		4) określa zasoby ludzkie oraz ramy czasowe wykonania projektu	x



Efekty kształcenia Stopniowane efektów kształcenia efekt kluczowy ek,, efekt ważny ew, efekt pomocniczy ep	Liczba godzin na efekt kształcenia	Kryteria weryfikacji	Tworzenie aplikacji
A	B	C	D
		5) planuje etapy tworzenia aplikacji	x
		6) korzysta z metodologii zarządzania projektem: model kaskadowy (waterfall), model przyrostowy, model prototypowy, metodyki zwinne (Agile oraz przynajmniej jedną z Scrum, Lean, Kanban)	x
		7) dobiera optymalną metodologię zarządzania projektem	x
		8) organizuje prace projektowe	x
		9) stosuje harmonogram czynności w celu efektywnego osiągnięcia celów	x
8) stosuje wzorce projektowe (ew)	4	1) dobiera wzorzec projektowy do zadania programistycznego	x
		2) stosuje wzorce projektowe w programowaniu obiektowym, np. Metoda szablonowa (Template method), Fasada (Facade), Kompozyt (Composite)	x
9) stosuje zagadnienia prawa autorskiego w dziedzinie programowania (ep)	4	1) rozróżnia autorskie prawa osobiste i majątkowe	x
		2) określa czas trwania praw autorskich	x
		3) określa konsekwencje naruszenia prawa autorskiego	x
		4) charakteryzuje elementy własności intelektualnej (dobra niematerialne, własności przemysłowe)	x
		5) rozróżnia typy licencji oprogramowania	x
Razem liczba godzin w jednostce efektów kształcenia	75		

Osoba prowadząca zajęcia wszystkich obowiązkowych zajęć edukacyjnych powinni stwarzać warunki do nabywania kompetencji personalnych i społecznych w ramach godzin przeznaczonych na kształcenie zawodowe.

Tabela 2. Grupowanie efektów kształcenia i nadawanie nazw tym zajęciom.

Nazwa jednostki efektów kształcenia	Efekt kształcenia wraz z kodowaniem (ek, ew, ep)	Kryteria weryfikacji kryteria podstawowe kryteria ponadpodstawowe	Grupowanie efektów kształcenia w przedmioty	Liczba godzin	Okres realizacji w cyklu nauczania
A	B	C	D	E	F
INF.04.3. Projektowanie oprogramowania (75h)	1) posługuje się prostymi typami danych (ek)	1) rozróżnia typy numeryczne stałoprzecinkowe i zmiennoprzecinkowe	Tworzenie aplikacji (75h)	4	W czasie kursu
		2) rozpoznaje typ logiczny			
		3) rozróżnia typy znakowe i łańcuchowe			
		4) posługuje się typami liczbowymi stałoprzecinkowe i zmiennoprzecinkowymi, typem logicznym, typem znakowym i łańcuchowym			
	2) posługuje się złożonymi typami danych (ek)	1) rozróżnia rodzaje złożonych typów danych		6	
		2) posługuje się tablicami jednowymiarowymi i dwuwymiarowymi			
		3) posługuje się tablicami dynamicznymi, asocjacyjnymi			
		4) posługuje się typem rekordowym, np. struktura, unia			
		5) posługuje się typem plikowym			
		6) posługuje się typem wskaźnikowym			
		7) charakteryzuje cechy kolekcji, w tym znaczenie iteratora			
		8) posługuje się kolekcjami, np. listami, kolejkami, stosami, wektorami			

Nazwa jednostki efektów kształcenia	Efekt kształcenia wraz z kodowaniem (ek, ew, ep)	Kryteria weryfikacji kryteria podstawowe kryteria ponadpodstawowe	Grupowanie efektów kształcenia w przedmioty	Liczba godzin	Okres realizacji w cyklu nauczania
A	B	C	D	E	F
	3) stosuje metody rozwiązywania problemów za pomocą algorytmów (ek)	9) projektuje zestawy danych dla problemu programistycznego		15	
		1) projektuje algorytmy za pomocą różnych metod: schematów blokowych, listy kroków, drzew decyzyjnych, pseudokodu			
		2) charakteryzuje algorytmy iteracyjne, tekstowe i szyfrowania, tablicowe			
		3) charakteryzuje algorytmy rekurencyjne			
		4) charakteryzuje problemy i metody ich rozwiązywania, np. algorytmy heurystyczne, problem komiwojażera			
		5) określa złożoność obliczeniową algorytmów			
		1) charakteryzuje typy sortowania i ich złożoność obliczeniową			
		2) stosuje różne typy sortowania, np. bąbelkowe, zachłanne, przez wstawianie, szybkie, metodą dziel i zwyciężaj			
		3) stosuje algorytmy wyszukiwania dla tablic, list, kolejek, stosów			
		1) określa funkcje narzędzi do zarządzania projektem		7	

Nazwa jednostki efektów kształcenia	Efekt kształcenia wraz z kodowaniem (ek, ew, ep)	Kryteria weryfikacji kryteria podstawowe kryteria ponadpodstawowe	Grupowanie efektów kształcenia w przedmioty	Liczba godzin	Okres realizacji w cyklu nauczania
A	B	C	D	E	F
	5) dobiera narzędzia i metodologie do planowania i zarządzania projektem (ew)	2) stosuje diagramy do zarządzania etapami projektu, zadaniami i czasem, np. diagram Gantta			
		3) korzysta z programów wspierających zarządzanie projektami, np. Jira, Trello			
		4) korzysta z systemu kontroli wersji, np. Git			
	6) projektuje aplikację (ek)	1) analizuje wymagania klienta i tworzy zgodny z nimi projekt		15	
		2) tłumaczy wymagania klienta na specyfikację techniczną dla zespołu programistów			
		3) identyfikuje elementy interfejsu użytkownika, np. okna, dialogi, kontrolki, formularze, paski narzędziowe, widgety			
		4) projektuje interfejs użytkownika i wygląd aplikacji			
		5) dostosowuje interfejs do różnych platform			
		6) projektuje aplikacje w różnych paradygmatach programowania: strukturalnym, obiektowym			
		7) projektuje aplikację opartą na architekturze klient-serwer			
		8) projektuje struktury danych dla aplikacji			

Nazwa jednostki efektów kształcenia	Efekt kształcenia wraz z kodowaniem (ek, ew, ep)	Kryteria weryfikacji kryteria podstawowe kryteria ponadpodstawowe	Grupowanie efektów kształcenia w przedmioty	Liczba godzin	Okres realizacji w cyklu nauczania
A	B	C	D	E	F
		9) projektuje funkcjonalność aplikacji			
		10) planuje system zabezpieczeń aplikacji			
	7) planuje przedsięwzięcie programistyczne (ew)	1) określa cel projektu		20	
		2) określa fazy realizacji projektu			
		3) charakteryzuje cykl życia projektu informatycznego i jego poszczególne etapy			
		4) określa zasoby ludzkie oraz ramy czasowe wykonania projektu			
		5) planuje etapy tworzenia aplikacji			
		6) korzysta z metodologii zarządzania projektem: model kaskadowy (waterfall), model przyrostowy, model prototypowy, metodyki zwinne (Agile oraz przynajmniej jedną z Scrum, Lean, Kanban)			
		7) dobiera optymalną metodologię zarządzania projektem			
		8) organizuje prace projektowe			
		9) stosuje harmonogram czynności w celu efektywnego osiągnięcia celów			
	8) stosuje wzorce projektowe (ew)	1) dobiera wzorzec projektowy do zadania programistycznego		4	

Nazwa jednostki efektów kształcenia	Efekt kształcenia wraz z kodowaniem (ek, ew, ep)	Kryteria weryfikacji kryteria podstawowe kryteria ponadpodstawowe	Grupowanie efektów kształcenia w przedmioty	Liczba godzin	Okres realizacji w cyklu nauczania
A	B	C	D	E	F
	9) stosuje zagadnienia prawa autorskiego w dziedzinie programowania (ep)	2) stosuje wzorce projektowe w programowaniu obiektowym, np. Metoda szablonowa (Template method), Fasada (Facade), Kompozyt (Composite)		4	
		1) rozróżnia autorskie prawa osobiste i majątkowe			
		2) określa czas trwania praw autorskich			
		3) określa konsekwencje naruszenia prawa autorskiego			
		4) charakteryzuje elementy własności intelektualnej (dobra niematerialne, własności przemysłowe)			
		5) rozróżnia typy licencji oprogramowania			

2.2. Określenie liczby godzin na kształcenie zawodowe

Tabela 3. Określenie liczby godzin poszczególnych zajęć z podziałem na zajęcia teoretyczne i praktyczne

Przedmiot	Liczba godzin zajęć teoretycznych	Liczba godzin zajęć praktycznych	Efekty kształcenia	Kryteria weryfikacji
A	B	C	D	E
Tworzenie aplikacji (75h)		4	1) posługuje się prostymi typami danych (ek)	1) rozróżnia typy numeryczne stałoprzecinkowe i zmiennoprzecinkowe
				2) rozpoznaje typ logiczny
				3) rozróżnia typy znakowe i łańcuchowe

Przedmiot	Liczba godzin zajęć teoretycznych	Liczba godzin zajęć praktycznych	Efekty kształcenia	Kryteria weryfikacji
A	B	C	D	E
				4) posługuje się typami liczbowymi stałoprzecinkowe i zmiennoprzecinkowymi, typem logicznym, typem znakowym i łańcuchowym
				1) rozróżnia rodzaje złożonych typów danych
		6	2) posługuje się złożonymi typami danych (ek)	2) posługuje się tablicami jednowymiarowymi i dwuwymiarowymi
				3) posługuje się tablicami dynamicznymi, asocjacyjnymi
				4) posługuje się typem rekordowym, np. struktura, unia
				5) posługuje się typem plikowym
				6) posługuje się typem wskaźnikowym
				7) charakteryzuje cechy kolekcji, w tym znaczenie iteratora
				8) posługuje się kolekcjami, np. listami, kolejkami, stosami, wektorami
				9) projektuje zestawy danych dla problemu programistycznego
		15	3) stosuje metody rozwiązywania problemów za pomocą algorytmów (ek)	1) projektuje algorytmy za pomocą różnych metod: schematów blokowych, listy kroków, drzew decyzyjnych, pseudokodu
				2) charakteryzuje algorytmy iteracyjne, tekstowe i szyfrowania, tablicowe
				3) charakteryzuje algorytmy rekurencyjne
				4) charakteryzuje problemy i metody ich rozwiązywania, np. algorytmy heurystyczne, problem komiwojażera
				5) określa złożoność obliczeniową algorytmów
				1) charakteryzuje typy sortowania i ich złożoność obliczeniową
				2) stosuje różne typy sortowania, np. bąbelkowe, zachłanne, przez wstawianie, szybkie, metodą dziel i zwyciężaj

Przedmiot	Liczba godzin zajęć teoretycznych	Liczba godzin zajęć praktycznych	Efekty kształcenia	Kryteria weryfikacji
A	B	C	D	E
		7	5) dobiera narzędzia i metodologie do planowania i zarządzania projektem (ew)	3) stosuje algorytmy wyszukiwania dla tablic, list, kolejek, stosów
				1) określa funkcje narzędzi do zarządzania projektem
				2) stosuje diagramy do zarządzania etapami projektu, zadaniami i czasem, np. diagram Gantta
				3) korzysta z programów wspierających zarządzanie projektami, np. Jira, Trello
		15	6) projektuje aplikację (ek)	4) korzysta z systemu kontroli wersji, np. Git
				1) analizuje wymagania klienta i tworzy zgodny z nimi projekt
				2) tłumaczy wymagania klienta na specyfikację techniczną dla zespołu programistów
				3) identyfikuje elementy interfejsu użytkownika, np. okna, dialogi, kontrolki, formularze, paski narzędziowe, widgety
				4) projektuje interfejs użytkownika i wygląd aplikacji
				5) dostosowuje interfejs do różnych platform
				6) projektuje aplikacje w różnych paradygmatach programowania: strukturalnym, obiektowym
				7) projektuje aplikację opartą na architekturze klient-serwer
				8) projektuje struktury danych dla aplikacji
				9) projektuje funkcjonalność aplikacji
				10) planuje system zabezpieczeń aplikacji
		2	7) planuje przedsięwzięcie programistyczne (ew)	1) określa cel projektu
				2) określa fazy realizacji projektu
				3) charakteryzuje cykl życia projektu informatycznego i jego poszczególne etapy
				4) określa zasoby ludzkie oraz ramy czasowe wykonania projektu
				5) planuje etapy tworzenia aplikacji

Przedmiot	Liczba godzin zajęć teoretycznych	Liczba godzin zajęć praktycznych	Efekty kształcenia	Kryteria weryfikacji
A	B	C	D	E
				6) korzysta z metodologii zarządzania projektem: model kaskadowy (waterfall), model przyrostowy, model prototypowy, metodyki zwinne (Agile oraz przynajmniej jedną z Scrum, Lean, Kanban)
				7) dobiera optymalną metodologię zarządzania projektem
				8) organizuje prace projektowe
				9) stosuje harmonogram czynności w celu efektywnego osiągnięcia celów
		4	8) stosuje wzorce projektowe (ew)	1) dobiera wzorzec projektowy do zadania programistycznego
				2) stosuje wzorce projektowe w programowaniu obiektowym, np. Metoda szablonowa (Template method), Fasada (Facade), Kompozyt (Composite)
			9) stosuje zagadnienia prawa autorskiego w dziedzinie programowania (ep)	1) rozróżnia autorskie prawa osobiste i majątkowe
				2) określa czas trwania praw autorskich
				3) określa konsekwencje naruszenia prawa autorskiego
				4) charakteryzuje elementy własności intelektualnej (dobra niematerialne, własności przemysłowe)
				5) rozróżnia typy licencji oprogramowania
Razem godzin:	0	75		
SUMA:	75			

2.3. Plan kursu umiejętności zawodowych

Kurs może być realizowany w formie:

- stacjonarnej – w czasie około 6 miesięcy (1 semestr), łącznie 75 godzin – zajęcia odbywają się 3 lub 4 dni w tygodniu po min. 8 godzin dziennie,
- zaocznej – w czasie około 6 miesięcy (65% z 75 godzin = 50 godzin) – zajęcia odbywają się co 2 tygodnie przez 2 dni po 10 godzin dziennie, a w uzasadnionych przypadkach – co tydzień przez 2 dni po 10 godzin dziennie.
- efekty kształcenia wskazane do realizacji w kształceniu teoretycznym mogą być (po spełnieniu wymagań określonych w aktualnych przepisach oświatowych) realizowane w formie kształcenia na odległość, przy czym zaliczenie tych zajęć nie może odbywać się w formie zdalnej. Przedmioty o kształceniu praktycznym ze względu na efekty uczenia wymagające fizycznej interakcji muszą być prowadzone w formie stacjonarnej.

Tabela 4. Plan zajęć kursu dodatkowych umiejętności zawodowych

Lp.	Powiązanie z podstawą programową	Przedmioty	Liczba godzin	Uwagi
1	INF.04.3. Projektowanie oprogramowania	Tworzenie aplikacji	75	Ze względu, iż w ramach kursu jest jeden przedmiot należy zwracać szczególną uwagę przy omawianiu tematyki zajęć na podstawie teoretyczną oraz zachęcać do korzystania z dokumentacji technicznej
		Łączna liczba godzin:	75	

3. Cele kształcenia kursu umiejętności zawodowych

Absolwent kursu powinien być przygotowany do wykonywania następujących zadań zawodowych w zakresie:

- znajomości i stosowanie różnych typów danych,
- znajomości i stosowanie różnych algorytmów,
- umiejętności tworzenia projektu aplikacji na podstawie wymagań klienta,
- planowania prac projektowych,
- licencjonowania.

4. Programy poszczególnych zajęć

4.1. Program nauczania dla przedmiotu: Tworzenie aplikacji

4.1.1 Cele ogólne przedmiotu to:

- Stosowanie różnych typów danych.
- Stosowanie różnych algorytmów.
- Tworzenie projektu aplikacji na podstawie wymagań klienta.
- Planowanie prac projektowych.
- Licencjonowanie.
- Nabywanie umiejętności pracy w grupach i zespołach.
- Nabycie kompetencji personalnych i społecznych.

4.1.2 Cele szczegółowe przedmiotu (słuchacz/uczestnik potrafi):

- zastosować różne typy danych,

- zastosować algorytmy,
- zastosować narzędzia wspomagające projektowanie aplikacji,
- współpracować z klientem,
- opracowywać funkcjonalność aplikacji,
- projektować aplikację,
- tworzyć harmonogramy pracy,
- określać zasady licencjonowania,
- zaplanować system zabezpieczeń licencyjnych w aplikacjach.

4.1.3 Materiał nauczania z uwzględnieniem opisu efektów kształcenia

Dział programowy	Tematy jednostek metodycznych	liczba godzin	Opis efektów kształcenia uwzględniający kryteria weryfikacji
Wstęp do programowania	Wstęp do programowania, typy danych	10	<ul style="list-style-type: none"> — rozróżniać typy numeryczne stałoprzecinkowe i zmiennoprzecinkowe — rozpoznawać typ logiczny — rozróżniać typy znakowe i łańcuchowe — posługiwać się tablicami jednowymiarowymi i dwuwymiarowymi — posługiwać się typem plikowym — posługiwać się typem wskaźnikowym — posługiwać się typami liczbowymi stałoprzecinkowe i zmiennoprzecinkowymi, typem logicznym, typem znakowym i łańcuchowym — rozróżniać rodzaje złożonych typów danych — posługiwać się tablicami dynamicznymi, asocjacyjnymi — posługiwać się typem rekordowym, np. struktura, unia — charakteryzować cechy kolekcji, w tym znaczenie iteratora — posługiwać się kolekcjami, np. listami, kolejkami, stosami, wektorami — projektować zestawy danych dla problemu programistycznego
Wstęp do programowania	Rozwiązywanie problemów przy użyciu algorytmów	15	<ul style="list-style-type: none"> — charakteryzować algorytmy iteracyjne, tekstowe i szyfrowania, tablicowe — charakteryzować algorytmy rekurencyjne — określać złożoność obliczeniową algorytmów — charakteryzować typy sortowania i ich złożoność obliczeniową — projektować algorytmy za pomocą różnych metod: schematów blokowych, listy kroków, drzew decyzyjnych, pseudokodu

Dział programowy	Tematy jednostek metodycznych	liczba godzin	Opis efektów kształcenia uwzględniający kryteria weryfikacji
			<ul style="list-style-type: none"> — charakteryzować problemy i metody ich rozwiązywania, np. algorytmy heurystyczne, problem komiwojażera — stosować różne typy sortowania, np. bąbelkowe, zachłanne, przez wstawianie, szybkie, metodą dziel i zwycięża — stosować algorytmy wyszukiwania dla tablic, list, kolejek, stosów
Projektowanie aplikacji	Narzędzia wspomagające projektowanie aplikacji	5	<ul style="list-style-type: none"> — określać funkcje narzędzi do zarządzania projektem — korzystać z programów wspierających zarządzanie projektami, np. Jira, Trello — stosować diagramy do zarządzania etapami projektu, zadaniami i czasem, np. diagram Gantta — korzystać z systemu kontroli wersji, np. Git
Projektowanie aplikacji	Projekt aplikacji na podstawie wymagań klienta	15	<ul style="list-style-type: none"> — tłumaczyć wymagania klienta na specyfikację techniczną dla zespołu programistów — identyfikować elementy interfejsu użytkownika, np. okna, dialogi, kontrolki, formularze, paski narzędziowe, widżety — projektować aplikacje w różnych paradygmatach programowania: strukturalnym, obiektowym — projektować funkcjonalność aplikacji — określać cel projektu — analizować wymagania klienta i tworzy zgodny z nimi projekt — projektować interfejs użytkownika i wygląd aplikacji — dostosowywać interfejs do różnych platform — projektować aplikację opartą na architekturze klient-serwer — projektować struktury danych dla aplikacji — planować system zabezpieczeń aplikacji — dobierać wzorzec projektowy do zadania programistycznego — stosować wzorce projektowe w programowaniu obiektowym, np. Metoda szablonowa (Template method), Fasada (Facade), Kompozyt (Composite)
Projektowanie aplikacji	Planowanie prac w projekcie	25	<ul style="list-style-type: none"> — korzystać z programów wspierających zarządzanie projektami, np. Jira, Trello — tłumaczyć wymagania klienta na specyfikację techniczną dla zespołu programistów — identyfikować elementy interfejsu użytkownika, np. okna, dialogi, kontrolki, formularze, paski narzędziowe, widżety

Dział programowy	Tematy jednostek metodycznych	liczba godzin	Opis efektów kształcenia uwzględniający kryteria weryfikacji
			<ul style="list-style-type: none"> — projektować aplikacje w różnych paradygmatach programowania: strukturalnym, obiektowym — projektować funkcjonalność aplikacji — określać cel projektu — określać zasoby ludzkie oraz ramy czasowe wykonania projektu — określać etapy tworzenia aplikacji — organizować prace projektowe — stosować harmonogram czynności w celu efektywnego osiągnięcia celów — projektować zestawy danych dla problemu programistycznego — stosować diagramy do zarządzania etapami projektu, zadaniami i czasem, np. diagram Gantta — korzystać z systemu kontroli wersji, np. Git — analizować wymagania klienta i tworzyć zgodny z nimi projekt — projektować interfejs użytkownika i wygląd aplikacji — dostosowywać interfejs do różnych platform — projektować aplikację opartą na architekturze klient-serwer — projektować struktury danych dla aplikacji — planować system zabezpieczeń aplikacji — określać fazy realizacji projektu — charakteryzować cykl życia projektu informatycznego i jego poszczególne etapy — korzystać z metodologii zarządzania projektem: model kaskadowy (waterfall), model przyrostowy, model prototypowy, metodyki zwinne (Agile oraz przynajmniej jedną z Scrum, Lean, Kanban) — dobierać optymalną metodologię zarządzania projektem — dobierać wzorzec projektowy do zadania programistycznego — stosować wzorce projektowe w programowaniu obiektowym, np. Metoda szablonowa (Template method), Fasada (Facade), Kompozyt (Composite)
Projektowanie aplikacji	Prawo autorskie i licencje	5	<ul style="list-style-type: none"> — rozróżniać autorskie prawa osobiste i majątkowe — określać czas trwania praw autorskich — rozróżniać typy licencji oprogramowania — planować system zabezpieczeń aplikacji \

Dział programowy	Tematy jednostek metodycznych	liczba godzin	Opis efektów kształcenia uwzględniający kryteria weryfikacji
			<ul style="list-style-type: none"> — określać konsekwencje naruszenia prawa autorskiego — charakteryzować elementy własności intelektualnej (dobra niematerialne, własności przemysłowe)

4.1.4 Procedury osiągnięcia celów kształcenia

Propozycje metod nauczania,

- metoda projektu (zalecana),
- metoda tekstu przewodniego,
- symulacje,
- gry dydaktyczne,
- pokaz z objaśnieniem,
- pokaz z instruktażem,
- ćwiczenia praktyczne (zalecana),
- praca w grupach (zalecana),
- użycie platform dydaktycznych,
- nauczanie na odległość z wykorzystaniem platform edukacyjnych.

Obudowa dydaktyczna

stanowisko komputerowe dla osoby prowadzącej kurs z:

- dostępem do internetu,
- pakietem programów biurowych
- projektorem multimedialnym lub tablicą multimedialną,
- urządzenie wielofunkcyjne sieciowe (drukarka z skanerem),
- monitorem minimum 21",
- zainstalowanymi środowiskami programistycznymi,
- zainstalowanym oprogramowaniem wspierającym projektowanie aplikacji.

stanowisko komputerowe dla kursanta (1 stanowisko na osobę) z:

- dostępem do internetu,
- pakietem programów biurowych
- monitorem minimum 21",

- zainstalowanymi środowiskami programistycznymi,
- zainstalowanym oprogramowaniem wspierającym testowanie.

W sali lekcyjnej powinny znajdować się e-booki, plansze dydaktyczne, schematy, grafiki i plansze interaktywne, gry edukacyjne, filmy instruktażowe (tutoriale), filmy edukacyjne, sekwencje filmowe, wizualizacje lub animacje 2D/3D, galerie zdjęć, symulatory, umożliwiające realizowanie treści w formie zdalnej i zawierające treści dotyczące:

- projektowania aplikacji,
- typów danych,
- algorytmów,
- frameworków.

Wypożyczenie sali powinno obejmować między innymi zestawy instrukcji eksploatacji używanych na zajęciach urządzeń i przyrządów, zestawy ćwiczeń, instrukcje do ćwiczeń, komputerowe programy demonstracyjne i symulacyjne, czasopisma branżowe, katalogi, schematy ideowe i montażowe, normy ISO i PN, pakiety edukacyjne dla słuchaczy, karty samooceny, karty pracy dla słuchaczy i prezentacje multimedialne.

Warunki realizacji

Zajęcia powinny być prowadzone z wykorzystaniem różnych form pracy aktywizującej słuchaczy np. praca w grupach po 2-3 słuchaczy. W trakcie prac ze słuchaczami należy pozostawiać im dodatkowy czas na własne prace związane z realizowanymi celami kształcenia. Dodatkowy czas należy też poświęcić na indywidualizowanie pracy słuchaczy w zależności od ich możliwości i potrzeb. W czasie prowadzenia zajęć w pracowni należy stosować zasadę, iż nieudane ćwiczenie też może być wysoko ocenione pod warunkiem, iż uczestnik kursu potrafi wyjaśnić przyczyny niepowodzenia oraz wskazać jak powinno ono przebiegać w prawidłowy sposób. Pozwoli to na indywidualizację prowadzonych działań przez słuchaczy oraz pokaże, iż doświadczenie można zdobyć nie tylko poprzez udane doświadczenia.

4.1.5 Proponowane metody sprawdzania osiągnięć edukacyjnych słuchacza/uczestnika

Sprawdzanie opanowania przez słuchacza/uczestnika wymagań programowych będzie przeprowadzone na podstawie wykonanych ćwiczeń. W ocenie należy uwzględnić następujące kryteria ogólne: zawartość merytoryczną ćwiczeń, ich poprawność, formy przedstawienia. Sprawdzanie osiągnięć powinno odbywać się przez cały okres realizacji programu zajęć na podstawie kryteriów przedstawionych na początku kursu.

5. Ewaluacja programu kursu dodatkowych umiejętności zawodowych

Efekt kształcenia z podstawy programowej (oznaczony w programie kursu jako kluczowy dla kwalifikacji lub jednostki efektów) (czy uczeń potrafi:)	Wskaźniki potwierdzające osiągnięcie efektu kształcenia	Metody/techniki badania	Termin badania
INF.04.3. Projektowanie oprogramowania			
1) posługuje się prostymi typami danych (ek)	1) rozróżnia typy numeryczne stałoprzecinkowe i zmiennoprzecinkowe 2) rozpoznaje typ logiczny 3) rozróżnia typy znakowe i łańcuchowe 4) posługuje się typami liczbowymi stałoprzecinkowe i zmiennoprzecinkowymi, typem logicznym, typem znakowym i łańcuchowym	wykład informacyjny, pokaz z objaśnieniem, wykład problemowy, film dydaktyczny, dyskusja dydaktyczna, burza mózgów, pokaz z instruktażem, pokaz z objaśnieniem, ćwiczenia przedmiotowe, ćwiczenia produkcyjne, metoda projektów, metoda przewodniego tekstu, metoda przypadków, metoda sytuacyjna, inscenizacja, dyskusja dydaktyczna, gry dydaktyczne, teksty zamknięte, próby pracy, testy zamknięte praca w grupie	W czasie realizacji programu nauczania podczas trwania kursu dodatkowych umiejętności zawodowych
2) posługuje się złożonymi typami danych (ek)	1) rozróżnia rodzaje złożonych typów danych 2) posługuje się tablicami jednowymiarowymi i dwuwymiarowymi 3) posługuje się tablicami dynamicznymi, asocjacyjnymi 4) posługuje się typem rekordowym, np. struktura, unia 5) posługuje się typem plikowym 6) posługuje się typem wskaźnikowym 7) charakteryzuje cechy kolekcji, w tym znaczenie iteratora 8) posługuje się kolekcjami, np. listami, kolejkami, stosami, wektorami 9) projektuje zestawy danych dla problemu programistycznego		
3) stosuje metody rozwiązywania problemów za pomocą algorytmów (ek)	1) projektuje algorytmy za pomocą różnych metod: schematów blokowych, listy kroków, drzew decyzyjnych, pseudokodu 2) charakteryzuje algorytmy iteracyjne, tekstowe i szyfrowania, tablicowe 3) charakteryzuje algorytmy rekurencyjne 4) charakteryzuje problemy i metody ich rozwiązywania, np. algorytmy heurystyczne, problem komiwojażera 5) określa złożoność obliczeniową algorytmów 1) charakteryzuje typy sortowania i ich złożoność obliczeniową		



Efekt kształcenia z podstawy programowej (oznaczony w programie kursu jako kluczowy dla kwalifikacji lub jednostki efektów) (czy uczeń potrafi:)	Wskaźniki potwierdzające osiągnięcie efektu kształcenia	Metody/techniki badania	Termin badania
	2) stosuje różne typy sortowania, np. bąbelkowe, zachłanne, przez wstawianie, szybkie, metodą dziel i zwycięża		
	3) stosuje algorytmy wyszukiwania dla tablic, list, kolejek, stosów		
6) projektuje aplikację (ek)	1) analizuje wymagania klienta i tworzy zgodny z nimi projekt		
	2) tłumaczy wymagania klienta na specyfikację techniczną dla zespołu programistów		
	3) identyfikuje elementy interfejsu użytkownika, np. okna, dialogi, kontrolki, formularze, paski narzędziowe, widgety		
	4) projektuje interfejs użytkownika i wygląd aplikacji		
	5) dostosowuje interfejs do różnych platform		
	6) projektuje aplikacje w różnych paradygmatach programowania: strukturalnym, obiektowym		
	7) projektuje aplikację opartą na architekturze klient-serwer		
	8) projektuje struktury danych dla aplikacji		
	9) projektuje funkcjonalność aplikacji		
	10) planuje system zabezpieczeń aplikacji		

6. Wykaz literatury oraz niezbędnych środków i materiałów dydaktycznych

6.1. Wykaz literatury

Proponowane podręczniki:

- Kwalifikacja INF.04. Projektowanie, programowanie i testowanie aplikacji. Część 1. Inżynieria programowania - projektowanie oprogramowania, testowanie i dokumentowanie aplikacji. Podręcznik do nauki zawodu technik programista,
- Testowanie oprogramowania. Podręcznik dla początkujących, Rafał Pawlak,
- Zarządzanie projektami informatycznymi. Subiektywne spojrzenie programisty, Joel Spolsky,
- Zarządzanie projektami IT, Joseph Philips,
- Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.

Literatura dodatkowa:

- Krzysztof Szczęch, Wanda Buwała, Bezpieczeństwo i higiena pracy, Podręcznik do kształcenia zawodowego. WSiP. Warszawa 2016.
- Marcin Czerwonka, Zenon Nowocień Kwalifikacja INF.02. Administracja i eksploatacja systemów komputerowych, urządzeń peryferyjnych i lokalnych sieci komputerowych. Część 1. Systemy komputerowe. Podręcznik do nauki zawodu technik informatyk, wyd. Helion,
- Jolanta Pokorska, Kwalifikacja INF.03. Tworzenie i administrowanie stronami i aplikacjami internetowymi oraz bazami danych. Część 2. Projektowanie i administrowanie bazami danych. Podręcznik do nauki zawodu technik informatyk i technik programista, wyd. Helion,
- Jolanta Pokorska, Podręcznik do zawodu technik informatyk, technik programista, część 1, Tworzenie stron internetowych, Kwalifikacja INF.03.Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych, wyd. Helion –podręcznik w przygotowaniu.
- Adam Freeman, HTML5. Przewodnik encyklopedyczny, wyd. Helion
- David Sawyer McFarland, CSS3 nieoficjalny podręcznik, wyd. Helion
- David Sawyer McFarland, JavaScript i jQuery. Nieoficjalny podręcznik, wyd. Helion,
- Eric T. Freeman, Elisabeth Robson, Programowanie w Javascript. Rusz głową!, wyd.Helion
- Luke Welling, Laura Thomson, PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty, wyd. Helion
- R. Sama, K. Sama, Język angielski zawodowy w branży informatycznej, wyd. WSiP, Warszawa 2016
- Bjarne Stroustrup, Język C++. Kompendium wiedzy, wyd. Helion,
- Stephen Prata, Język C++. Szkoła programowania. Wydanie VI, wyd. Helion,
- Grębosz Jerzy, Opus magnum C++11, Programowanie w języku C++ (komplet), Wyd.: Grębosz Jerzy,
- Python dla każdego. Podstawy programowania – Michael Dawson, wyd. Helion,
- Eric Matthes, Python. Instrukcje dla programisty, wyd. Helion,
- Al Sweigart, Automatyzacja nudnych zadań z Pythonem, wyd. Helion,
- Dawn Griffiths, David Griffiths, Android. Programowanie aplikacji. Rusz głową! Wydanie II, wyd. Helion,
- Marcin Płonkowski, Android Studio. Tworzenie aplikacji mobilnych. wyd. Helion,
- Matt Neuburg, iOS 12. Wprowadzenie do programowania w Swifcie. Wydanie V, wyd. Helion,
- Steven F. Daniel, Xamarin. Tworzenie interfejsów użytkownika, wyd. Helion.
- Gniewomir Sarbicki, Python. Kurs dla osoba prowadząca zajęcia i studentów, wyd. Helion,
- Adriaan de Jonge, Phillip Dutson, jQuery, jQuery UI oraz jQuery Mobile. Receptury, wyd. Helion,
- Jon Duckett, JavaScript i jQuery. Interaktywne strony WWW dla każdego. Podręcznik Front-End Developera wyd. Helion,
- David Herron, Platforma Node.js. Przewodnik webdevelopera. Wydanie III, wyd. Helion,
- Antonio Mele, Django 2. Praktyczne tworzenie aplikacji sieciowych. Wydanie II, wyd. Helion.
- <http://codecademy.com>
- <http://w3schools.com>
- <http://www.cplusplus.com/>
- <https://docs.python.org/3/tutorial/>

- <https://doc.qt.io/qtcreator/index.html>
- <http://www-cs.cuny.cuny.edu/~wolberg/cs221/qt/books/C++-GUI-Programming-with-Qt-4-1st-ed.pdf>
- https://qmlbook.github.io/assets/qt5_cadaques.pdf

Czasopisma branżowe:

- Programista (programista.pl),

6.2. Wykaz niezbędnych środków i materiałów dydaktycznych

stanowisko komputerowe dla osoby prowadzącej zajęcia z:

- dostępem do internetu,
- pakietem programów biurowych
- projektorem multimedialnym lub tablicą multimedialną,
- urządzenie wielofunkcyjne sieciowe (drukarka z skanerem),
- środowiska programistyczne,
- monitorem minimum 24”,

Stanowiska komputerowe kursantów (1 stanowisko na 1 osobę) powinny być wyposażone następująco:

- komputer z połączeniem do internetu z zainstalowanymi środowiskami programistycznymi i ekranem minimum 21”,
- zainstalowane stosowne środowiska programistyczne,
- zainstalowane oprogramowanie biurowe,
- zainstalowane oprogramowanie do obsługi zgłoszeń i błędów,
- zainstalowane oprogramowanie do zarządzania i kontroli wersji oprogramowania,
- zainstalowane oprogramowanie wspierające testowanie aplikacji,
- zainstalowane oprogramowanie wspierające projektowanie aplikacji,
- zainstalowanymi bibliotekami i frameworkami do tworzenia aplikacji webowych ,
- zainstalowanymi serwerami z obsługą PHP/Python/JS itp. i baz danych.
- zainstalowanymi bibliotekami i frameworkami do tworzenia aplikacji mobilnych ,
- zainstalowane emulatory urządzeń mobilnych z system Android.

W salach lekcyjnych oraz pomieszczeniach praktycznej nauki zawodu powinny znajdować się przepisy BHP dotyczące pracy z urządzeniami oraz instrukcje obsługi i konserwacji tych urządzeń. Niezbędne są również zestawy ćwiczeń, instrukcje do ćwiczeń, pakiety edukacyjne dla słuchaczy, karty samooceny, karty pracy dla słuchaczy i prezentacje multimedialne związane z treściami kształcenia, czasopisma branżowe, katalogi, normy ISO i PN.

Dodatkowo do dyspozycji wskazane są tematyczne e-booki, animacje 2D/3D, atlasy interaktywne, plansze interaktywne, gry edukacyjne, filmy edukacyjne, symulatory, wirtualne laboratoria, programy ćwiczeniowe do projektowania przez dobieranie umożliwiający realizowanie treści w formie zdalnej dostosowane treściami do poszczególnych przedmiotów teoretycznych i praktycznych.

7. Sposób i forma zaliczenia kursu

Kurs dodatkowych umiejętności zawodowych kończy się zaliczeniem w formie walidacji osiągnięć uczestnika kursu, polegającej na ocenie wykonywanych w trakcie nauki projektów i ćwiczeń oraz na podstawie uzyskanych w trakcie kursu ocen z poszczególnych przedmiotów.

Do oceny osiągnięć edukacyjnych słuchaczy proponuje się stosowanie testów wielokrotnego wyboru, zadań z luką, ocenę aktywności słuchacza podczas wykonywania zadań w grupie, ocenę jakości wykonania zadań przez słuchacza. Proponuje się, aby osiągnięcia słuchaczy oceniać w zakresie zaplanowanych, uszczegółowionych celów kształcenia na podstawie:

- obserwacji wykonanych ćwiczeń,
- testu pisemnego.

Umiejętności praktyczne proponuje się sprawdzać na podstawie obserwacji czynności wykonywanych przez słuchacza w trakcie realizacji ćwiczeń. Podczas obserwacji należy zwrócić uwagę na:

- wyszukiwanie i przetwarzanie rzetelnych informacji pozyskanych z różnych źródeł,
- poprawność merytoryczną wykonanych ćwiczeń praktycznych,
- umiejętność pracy w zespole.

Ważne kryteria oceny efektów kształcenia to: zaplanowanie wykonania zadania, dobór elementów oraz sporządzona dokumentacja techniczna. Możliwe są również inne sposoby i formy zaliczenia, takie jak: testy praktyczne, wykonanie projektów, próby pracy, aktywność uczącego się na zajęciach, prezentacje na forum grupy z przeprowadzonych prac. Osoba, która uzyskała zaliczenie, otrzymuje zaświadczenie o ukończeniu kursu dodatkowych umiejętności zawodowych.

W ostatnim zdaniu punktu 7. programu należy dopisać

Osoba, która uzyskała zaliczenie, otrzymuje zaświadczenie o ukończeniu kursu umiejętności zawodowych. Osoba, która ukończy również kursy umiejętności zawodowych z jednostek efektów kształcenia:

- INF.04.1. Bezpieczeństwo i higiena pracy
- INF.04.2. Podstawy informatyki
- INF.04.4. Programowanie obiektowe
- INF.04.5. Programowanie aplikacji desktopowych
- INF.04.6. Programowanie aplikacji mobilnych
- INF.04.7. Programowanie aplikacji zaawansowanych webowych
- INF.04.8. Testowanie i dokumentowanie aplikacji
- INF.04.9. Język obcy zawodowy

i otrzymała zaświadczenie o ich ukończeniu może przystąpić do egzaminu zawodowego potwierdzającego kwalifikację INF.04. Projektowanie, programowanie i testowanie aplikacji, organizowanego przez Okręgową Komisję Egzaminacyjną, po zdaniu którego otrzymuje certyfikat kwalifikacji zawodowej.

8. Sprawdzenie kompletności i poprawności opracowanego programu zajęć

Tabela 4. Tabela weryfikacji programu nauczania kursu dodatkowych umiejętności zawodowych pod kątem zgodności z przepisami prawa oświatowego

Lp.	Program kursu dodatkowych umiejętności zawodowych uwzględnia	Zawartość opracowanego programu zajęć (T/N)
1.	Cele kształcenia (zadania zawodowe)	T
2.	Efekty kształcenia	T
3.	Kryteria weryfikacji	T
4.	Warunki realizacji kształcenia w kwalifikacji (lub niezbędne do realizacji danej jednostki efektów)	T
5.	Minimalna liczba godzin kształcenia zawodowego dla kwalifikacji wyodrębnionej w zawodzie lub jednostki efektów	T

Tabela 5. Tabela weryfikacji programu kursu dodatkowych umiejętności zawodowych pod kątem kompletności efektów kształcenia

Efekty kształcenia oraz kryteria weryfikacji określone w podstawie programowej kształcenia w zawodzie		Zawartość opracowanego programu zajęć (temat zajęć)
INF.04.3. Projektowanie oprogramowania		
1) posługuje się prostymi typami danych (ek)	1) rozróżnia typy numeryczne stałoprzecinkowe i zmiennoprzecinkowe	Wstęp do programowania, typy danych
	2) rozpoznaje typ logiczny	Wstęp do programowania, typy danych
	3) rozróżnia typy znakowe i łańcuchowe	Wstęp do programowania, typy danych
	4) posługuje się typami liczbowymi stałoprzecinkowe i zmiennoprzecinkowymi, typem logicznym, typem znakowym i łańcuchowym	Wstęp do programowania, typy danych
2) posługuje się złożonymi typami danych (ek)	1) rozróżnia rodzaje złożonych typów danych	Wstęp do programowania, typy danych
	2) posługuje się tablicami jednowymiarowymi i dwuwymiarowymi	Wstęp do programowania, typy danych
	3) posługuje się tablicami dynamicznymi, asocjacyjnymi	Wstęp do programowania, typy danych
	4) posługuje się typem rekordowym, np. struktura, unia	Wstęp do programowania, typy danych
	5) posługuje się typem plikowym	Wstęp do programowania, typy danych
	6) posługuje się typem wskaźnikowym	Wstęp do programowania, typy danych
	7) charakteryzuje cechy kolekcji, w tym znaczenie iteratora	Wstęp do programowania, typy danych
	8) posługuje się kolekcjami, np. listami, kolejkami, stosami, wektorami	Wstęp do programowania, typy danych
	9) projektuje zestawy danych dla problemu programistycznego	Wstęp do programowania, typy danych Planowanie prac w projekcie



Efekty kształcenia oraz kryteria weryfikacji określone w podstawie programowej kształcenia w zawodzie		Zawartość opracowanego programu zajęć (temat zajęć)
3) stosuje metody rozwiązywania problemów za pomocą algorytmów (ek)	1) projektuje algorytmy za pomocą różnych metod: schematów blokowych, listy kroków, drzew decyzyjnych, pseudokodu	Rozwiązywanie problemów przy użyciu algorytmów
	2) charakteryzuje algorytmy iteracyjne, tekstowe i szyfrowania, tablicowe	Rozwiązywanie problemów przy użyciu algorytmów
	3) charakteryzuje algorytmy rekurencyjne	Rozwiązywanie problemów przy użyciu algorytmów
	4) charakteryzuje problemy i metody ich rozwiązywania, np. algorytmy heurystyczne, problem komiwojażera	Rozwiązywanie problemów przy użyciu algorytmów
	5) określa złożoność obliczeniową algorytmów	Rozwiązywanie problemów przy użyciu algorytmów
	1) charakteryzuje typy sortowania i ich złożoność obliczeniową	Rozwiązywanie problemów przy użyciu algorytmów
	2) stosuje różne typy sortowania, np. bąbelkowe, zachłanne, przez wstawianie, szybkie, metodą dziel i zwyciężaj	Rozwiązywanie problemów przy użyciu algorytmów
	3) stosuje algorytmy wyszukiwania dla tablic, list, kolejek, stosów	Rozwiązywanie problemów przy użyciu algorytmów
5) dobiera narzędzia i metodologie do planowania i zarządzania projektem (ew)	1) określa funkcje narzędzi do zarządzania projektem	Narzędzia wspomagające projektowanie aplikacji
	2) stosuje diagramy do zarządzania etapami projektu, zadaniami i czasem, np. diagram Gantta	Narzędzia wspomagające projektowanie aplikacji Planowanie prac w projekcie
	3) korzysta z programów wspierających zarządzanie projektami, np. Jira, Trello	Narzędzia wspomagające projektowanie aplikacji Planowanie prac w projekcie
	4) korzysta z systemu kontroli wersji, np. Git	Narzędzia wspomagające projektowanie aplikacji Planowanie prac w projekcie
6) projektuje aplikację (ek)	1) analizuje wymagania klienta i tworzy zgodny z nimi projekt	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie



Efekty kształcenia oraz kryteria weryfikacji określone w podstawie programowej kształcenia w zawodzie		Zawartość opracowanego programu zajęć (temat zajęć)
	2) tłumaczy wymagania klienta na specyfikację techniczną dla zespołu programistów	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie
	3) identyfikuje elementy interfejsu użytkownika, np. okna, dialogi, kontrolki, formularze, paski narzędziowe, widżety	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie
	4) projektuje interfejs użytkownika i wygląd aplikacji	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie
	5) dostosowuje interfejs do różnych platform	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie
	6) projektuje aplikacje w różnych paradygmatach programowania: strukturalnym, obiektowym	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie
	7) projektuje aplikację opartą na architekturze klient-serwer	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie
	8) projektuje struktury danych dla aplikacji	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie
	9) projektuje funkcjonalność aplikacji	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie
	10) planuje system zabezpieczeń aplikacji	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie Prawo autorskie i licencje
	7) planuje przedsięwzięcie programistyczne (ew)	
	1) określa cel projektu	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie



Efekty kształcenia oraz kryteria weryfikacji określone w podstawie programowej kształcenia w zawodzie		Zawartość opracowanego programu zajęć (temat zajęć)
	2) określa fazy realizacji projektu	Planowanie prac w projekcie
	3) charakteryzuje cykl życia projektu informatycznego i jego poszczególne etapy	Planowanie prac w projekcie
	4) określa zasoby ludzkie oraz ramy czasowe wykonania projektu	Planowanie prac w projekcie
	5) planuje etapy tworzenia aplikacji	Planowanie prac w projekcie
	6) korzysta z metodologii zarządzania projektem: model kaskadowy (waterfall), model przyrostowy, model prototypowy, metodyki zwinne (Agile oraz przynajmniej jedną z Scrum, Lean, Kanban)	Planowanie prac w projekcie
	7) dobiera optymalną metodologię zarządzania projektem	Planowanie prac w projekcie
	8) organizuje prace projektowe	Planowanie prac w projekcie
	9) stosuje harmonogram czynności w celu efektywnego osiągnięcia celów	Planowanie prac w projekcie
8) stosuje wzorce projektowe (ew)	1) dobiera wzorzec projektowy do zadania programistycznego	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie
	2) stosuje wzorce projektowe w programowaniu obiektowym, np. Metoda szablonowa (Template method), Fasada (Facade), Kompozyt (Composite)	Projekt aplikacji na podstawie wymagań klienta Planowanie prac w projekcie
9) stosuje zagadnienia prawa autorskiego w dziedzinie programowania (ep)	1) rozróżnia autorskie prawa osobiste i majątkowe	Prawo autorskie i licencje
	2) określa czas trwania praw autorskich	Prawo autorskie i licencje
	3) określa konsekwencje naruszenia prawa autorskiego	Prawo autorskie i licencje
	4) charakteryzuje elementy własności intelektualnej (dobra niematerialne, własności przemysłowe)	Prawo autorskie i licencje
	5) rozróżnia typy licencji oprogramowania	Prawo autorskie i licencje