

Roczny plan dydaktyczny Informatyka zakres rozszerzony tom 2 (klasa 2 LO)

Temat (rozumiany jako lekcja) Rozdział z podręcznika	Liczba godzin	Treści (pkt podstawa programowej)	Cele ogólne	Cele szczegółowe	Kształcenie umiejętności	Propozycje metod nauczania	Propozycje środków dydaktycznych	Uwagi
I. Podstawy programowania								
Edytor i kompilator, czyli środowisko zintegrowane (IDE) (1)	1/2	5.22	Odpowiedź na pytanie – czym jest środowisko zintegrowane IDE.	Poznanie budowy i podstawowych możliwości edytora i kompilatora Turbo Pascal. Poznanie istoty kodu źródłowego, kompilacji.	Uczeń: – posługuje się menu edytora Turbo Pascal; – odnajduje opcje dotyczące edycji i kompilacji kodu.	Wykład ilustrowany. Pokaz i ćwiczenia z edytorem TP.	Środowisko zintegrowane Turbo Pascal. Rzutnik, duży ekran, np. LCD lub tablica interaktywna.	Do pokazu można użyć jednego z kodów programu używanych w następnych lekcjach.
Edytor programu Free Pascal (2)	1/2	5.22	Poznanie zasad edycji tekstu kodu źródłowego w środowisku Free Pascal.	Poznanie sposobu wyświetlania tekstu kodu w edytorze IDE z uwzględnieniem kolorów dla słów kluczowych. Poznanie podstawowych narzędzi edycyjnych takich jak Search, Replace z uwzględnieniem ich opcji.	Uczeń: – stosuje zasady edycji kodu z zachowaniem wcięć, podziału na linie.	Wykład ilustrowany i ćwiczenia z edytorem Free Pascal z zastosowaniem przykładowego kodu programu.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna.	Program proponowany przez autora (CD).
Tworzenie kodu źródłowego i budowa programu	1	5.22, 5.23	Poznanie struktury kodu programu	Analiza budowy programu na podstawie	Uczeń – umie wyróżnić poszczególne	Analiza przypadkui ćwiczenia z	Środowisko zintegrowane Free Pascal.	Program o typowej strukturze,

(3)			komputerowego w języku Pascal.	przykładowego kodu z wyróżnieniem jego najważniejszych elementów struktury i budowy (wcięcia, średniki, dwukropki, słowa kluczowe, komentarze itp.).	bloki kodu programu; – kompiluje program i analizuje wynik kompilacji; – stosuje komentarze w treści programie.	edytorem Free Pascal dotyczące komunikatów kompilatora w przypadku błędu w kodzie.	Rzutnik, duży ekran, np. LCD, lub tablica interaktywna.	np. proponowany przez autora (CD).
Identyfikatory w Turbo Pascalu (4)	1	5.22, 5.23	Poznanie pojęcia identyfikatora.	Poznanie definicji identyfikatorów w języku Pascal i zasad tworzenia ich nazw. Poznanie komunikatów kompilatora o błędach w nazwach identyfikatorów.	Uczeń: – umie poprawnie zapisywać nazwy identyfikatorów; – zna zasady i ograniczenia dotyczące identyfikatorów; – umie odczytywać komunikaty kompilatora dotyczące identyfikatorów.	Analiza przypadku na podstawie treści kodów programów z poprawnie i błędnie nazwanymi lub użytymi identyfikatorami.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Przykłady kodów programów.	Należy zwrócić uwagę na to, że identyfikatory mogą składać się tylko z liter, cyfr i znaku podkreślenia, muszą zaczynać się od litery lub podkreślenia, mogą być słowami kluczowymi.
Typy proste (5)	1	5.22, 5.23	Poznanie klasyfikacji typów.	Zdefiniowanie typu prostego. Poznanie cech i właściwości typów: całkowitych, rzeczywistych, znakowych i logicznych. Poznanie	Uczeń: – umie odróżnić typ prosty od złożonego; – umie określić właściwości poszczególnych typów prostych; – określa wpływ typu na zajętość	Wykład ilustrowany z przykładami niedostosowania typów do danych przez nie reprezentowanych. Tabela 5.1. z podręcznika. Proste	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Przykłady kodów programów. Tabela 5.1.	Podkreślić, że typy dzielimy na standardowe i te, które może tworzyć programista. Każdy typ zajmuje

				wpływu rodzaju typu na zajętość pamięci. Przypomnienie kodów ASCII.	pamięci; – wie, jak zapisać typ znakowy, by odróżnić go od identyfikatora; – posługuje się kodem ASCII.	ćwiczenia.		określoną wielkość pamięci.
Zmienne i deklaracje (6)	1	5.22, 5.23	Zapoznanie się ze sposobami deklaracji zmiennych.	Poznanie znaczenia zmiennych. Poprawna deklaracja różnych typów zmiennych. Sposoby przypisywania wartości zmiennym. Wyświetlanie zawartości zmiennej.	Uczeń: – umie deklorować zmienne różnych typów; – stosuje odpowiednie słowa kluczowe i znaki (; :) do deklaracji zmiennych; – wyświetla zawartość różnych zmiennych; – nadaje wartość zmiennej za pomocą procedury readln.	Wykład ilustrowany przykładami deklaracji zmiennych. Ćwiczenie z deklaracji zmiennych.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Fragmenty programów.	Zwrócić uwagę na to, że zmienne muszą mieć nadaną nazwę i określony typ i służą do przechowywania wartości. Sama deklaracja nie powoduje nadania wartości zmiennej.
Stałe i definicje (7)	1	5.22, 5.23	Zapoznanie się ze sposobem deklaracji stałej i jej znaczeniem w programie.	Poznanie celowości stosowania stałych. Poprawna deklaracja stałej z użyciem const identyfikatora stałej operatora = i wartości. Wyświetlanie zawartości stałej.	Uczeń: – umie deklorować stałe i przypisywać im wartości; – stosuje poprawny schemat deklaracji stałej; – wyświetla zawartość różnych stałych.	Analiza przypadku na podstawie treści kodów programów z poprawnie i błędnie zadeklarowanym i stałymi.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Fragmenty programów.	Zwrócić uwagę na to, że w definicji stałych można użyć operatorów arytmetycznych, tworząc wyrażenia i nazw innych

								stałych.
Operatory i wyrażenia (8)	1	5.22, 5.23	Zapoznanie się z rodzajami i funkcjami operatorów.	Zdefiniowanie pojęć, np. wyrażenie całkowite. Poznanie znaczenia i zastosowania operatorów.	Uczeń: – umie stosować operatory działające na różnych typach zmiennych i stałych; – umie zapisywać skomplikowane wyrażenia arytmetyczne za pomocą operatorów.	Wykład ilustrowany przykładami deklaracji stałych. Ćwiczenie z zapisywania wyrażeń arytmetycznych za pomocą operatorów.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zadania oceniane na zajęciach. Pod uwagę brana jest szybkość i dokładność wykonania.
Operatory i wyrażenia, typ znakowy i logiczny – ćwiczenia (9, 10)	1	5.22, 5.23	Praktyczne przećwiczenie zasad budowy wyrażeń z operatorami.	Poprawne używanie operatorów znakowych i logicznych. Używanie operatorów do obliczeń w programach.	Uczeń: – umie stosować operatory działające na typach logicznych i znakowych; – umie zapisywać skomplikowane wyrażenia logiczne za pomocą operatorów; – poprawnie zapisuje wyrażenia znakowe.	Praca indywidualna z podręcznikiem – ćwiczenie na ocenę.	Środowisko zintegrowane Free Pascal. Podręcznik i platforma e-learningowa. Przykłady kodów programów.	Zwrócić uwagę na to, że dla każdego typu wyniku przeznaczone są odpowiednie operatory.
Instrukcje proste – zmieniamy wartość zmiennych (11)	1	5.22, 5.23	Poznanie sposobów przypisywania wartości zmiennym.	Poznanie instrukcji przypisania. Poznanie warunku zgodności. Poznanie komunikatów	Uczeń: – stosuje instrukcje przypisania w odniesieniu do różnych typów i wyrażeń; – przypisuje	Analiza przypadku na podstawie treści kodów programów z poprawnie i błędnie użytym przypisaniem.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-	Zwrócić uwagę na to, że należy koniecznie sprawdzić, czy typ deklarowan

				<p>kompilatora dotyczących zgodności typów podczas przypisania i zmiany wartości zmiennych.</p>	<p>wartość jednej zmiennej do drugiej; – prawidłowo interpretuje komunikaty kompilatora; – wyświetla wartości, do których przypisano inne zmienne lub wyrażenia; – wskazuje źródła niezgodności typów przy przypisaniu; – zna pojęcie instrukcji pustej.</p>		<p>learningowa do przesłania wyników ćwiczenia.</p>	<p>ej zmiennej pomieści wartość, która może być przypisana zmiennej.</p>
<p>Instrukcje warunkowe – rozgałęziamy działania (12)</p>	1	5.22, 5.23	<p>Poznanie instrukcji warunkowej if.</p>	<p>Poznanie składni instrukcji warunkowej, warunkowej złożonej i instrukcji wyboru. Analiza przykładów instrukcji warunkowej if i instrukcji wyboru case. Programowa realizacja bloków warunkowych algorytmu.</p>	<p>Uczeń: – prawidłowo stosuje instrukcje warunkowe, warunkowe złożone i instrukcji wyboru; – wskazuje fragmenty algorytmów, które mogą być realizowane za pomocą instrukcji warunkowych i instrukcji wyboru.</p>	<p>Wykład ilustrowany i analiza przypadku na podstawie treści kodów programów z poprawnie i błędnie użytymi instrukcjami warunkowymi i instrukcji wyboru. Ćwiczenie na ocenę.</p>	<p>Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia. Przykłady kodów programów.</p>	<p>Zwrócić uwagę na to, że instrukcja złożona zawiera się w bloku.</p>

Instrukcja wyboru – case (13)	1	5.22, 5.23	Poznanie instrukcji wyboru case.	Poznanie składni instrukcji wyboru, analiza fragmentów kodu z zastosowaniem case, napisanie programu z wykorzystaniem instrukcji case na podstawie opisu sytuacji.	Uczeń: – prawidłowo stosuje instrukcję wyboru case; – zna różnicę pomiędzy instrukcją wyboru a instrukcją warunkową; – prawidłowo stosuje typy dla selektorów.	Wykład ilustrowany i analiza przypadku na podstawie treści kodów programów z poprawnie i błędnie użytymi instrukcją wyboru. Ćwiczenie na ocenę.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że do podejmowania decyzji w programie służą instrukcje warunkowe if then else oraz case.
Instrukcje warunkowe – ćwiczenia (12, 13)	1	5.22, 5.23	Praktyczne wykorzystanie instrukcji warunkowych.	Programowa realizacja bloków warunkowych algorytmu za pomocą instrukcji warunkowych if i warunkowych złożonych if then else oraz instrukcji wyboru case.	– prawidłowo zapisuje w języku programowania fragmenty algorytmów z blokami warunkowymi, stosując instrukcje warunkowe i instrukcję wyboru case.	Quiz dydaktyczny i ćwiczenie na ocenę.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia. Przykłady kodów programów.	Zwrócić uwagę na to, że w instrukcji case wyrażenie nie musi być wyrażeniem logicznym.
Instrukcje powtarzania – pętla for (14)	1	5.22, 5.23	Poznanie istoty instrukcji for i jej zastosowania.	Poznanie budowy instrukcji for. Poznanie działania instrukcji warunkowej krok po kroku. Poznanie różnicy pomiędzy <i>do a down to</i> .	Uczeń: – prawidłowo stosuje zmienne sterujące występujące w pętlach for; – opisuje kolejne działania podejmowane przez komputer w czasie wykonywania pętli for;	Wykład ilustrowany przykładami wykorzystania pętli for. Burza mózgów nad realizacją programu z użyciem pętli for.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia. Przykłady kodów	Zwrócić uwagę na to, że prawidłowa pętla for wykonuje się skończoną ilość razy.

					– układa programy realizujące proste algorytmy realizowane za pomocą pętli for.		programów.	
Instrukcje powtarzania – pętle repeat ... until (15)	1	5.22, 5.23	Zrozumienie istoty pętli wykonywanej dopóty, dopóki warunek nie jest spełniony.	Poznanie składni instrukcji pętli repeat ... until. Poznanie różnicy pomiędzy repeat ... until a for. Układanie programów z zastosowaniem pętli repeat ... until.	Uczeń: – wskazuje istotne różnice pomiędzy działaniem pętli for i repeat ... until; – układa programy z użyciem instrukcji repeat ... until.	Wykład ilustrowany przykładami wykorzystania pętli repeat ... until. Burza mózgów nad realizacją programu z użyciem pętli repeat ... until.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia. Przykłady kodów programów.	Zwrócić uwagę na to, że pętli repeat until używa się, gdy nie jest znana ilość powtórzeń, jednak musi ona być skończona.
Instrukcje powtarzania – pętle while ... do (16)	1	5.22, 5.23	Zrozumienie istoty pętli while ... do wykonywanej jeśli warunek został spełniony.	Poznanie składni instrukcji while ... do. Poznanie różnicy pomiędzy while ... do a poznanymi wcześniej instrukcjami pętli.	Uczeń: – wskazuje i opisuje istotne różnice pomiędzy pętlą while ... do a poznanymi wcześniej instrukcjami pętli; – prawidłowo układa programy z zastosowaniem instrukcji while ... do.	Metoda problemowa – np. wspólne rozwiązanie problemu liczenia głosów w wyborach.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Przykłady kodów programów. Platforma e-learningowa do przesłania wyników ćwiczenia.	Początek tworzenia programu liczącego głosy w wyborach do Samorządu Uczniowskiego.
	1	5.22, 5.23	Poznanie istoty typu złożonego tablicowego.	Poznanie sposobu deklarowania tablic. Zdobyć	Uczeń: – prawidłowo deklaruje tablice jedno- i wielowymiarowe;	Wykład ilustrowany przykładami.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD,	Kontynuacja tworzenia programu liczącego głosy w

				umiejętności używania tablic wielowymiarowych.	– tłumaczy zastosowanie tablic.		lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	wyborach do Samorządu Uczniowskiego.
Tablice – definiujemy własne typy złożone (18)	1	5.22, 5.23	Poznanie zmiennych tablicowych i ich zastosowania.	Deklarowanie zmiennych dla tablic.	Uczeń: – wymienia typy zmiennych dopuszczalne w tablicach; – wyjaśnia, na czym polega różnica pomiędzy zmiennymi tablicowymi a zmiennymi jednoelementowymi; – deklaruje zmienne tablicowe; – podaje przykłady zastosowania zmiennych tablicowych.	Ćwiczenia z zastosowaniem zmiennych tablicowych	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Przykłady kodów programów. Platforma e-learningowa do przesłania wyników ćwiczenia.	Kontynuacja i tworzenia programu liczącego głosy w wyborach do Samorządu Uczniowskiego.
Tablice – definiujemy własne typy złożone – ćwiczenia (19)	1	5.22, 5.23	Praktyczna realizacja programu ze zmiennymi zapisanymi w tablicach.	Poznanie następujących zagadnień: – zastąpienie pojedynczych zmiennych tablicowymi; – nadawanie wartości początkowych tablicy; – modyfikacja	Uczeń: – definiuje własne tablice zmiennych różnych typów; – prawidłowo zastępuje zmienną tablicową osobno występujące zmienne,	Wykład ilustrowany przykładami wykorzystania tablic.	Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Kontynuacja i tworzenia programu liczącego głosy w wyborach do Samorządu Uczniowskiego.

				<p>elementów tablicy. – zajętość pamięci.</p>	<p>uzasadniając swoją decyzję i wykazując zasadność takiego postępowania; – nadaje wartości początkowe zmiennym tablicowym; – prawidłowo dobiera rozmiary zmiennych tablicowych i typy zmiennych zapisywanych w tablicach.</p>			
Łańcuchy (20)	1	5.22, 5.23	<p>Poznanie zmiennych typu łańcuchowego.</p>	<p>Poznanie sposobu deklaracji zmiennej typu string. Opanowanie podstawowych operacji na zmiennych typu string. Sprawdzenie działania standardowych funkcji i procedur działających na stringach.</p>	<p>Uczeń: – poprawnie deklaruje zmienne typu łańcuchowego; – poprawnie dobiera długość deklarowanej zmiennej typu string; – dodaje zmienne łańcuchowe; – programuje wprowadzanie zmiennych łańcuchowych za pomocą klawiatury; – szacuje zajętą ilość pamięci na podstawie deklaracji</p>	<p>Wykład ilustrowany przykładami wykorzystania ze zmiennych łańcuchowych. Ćwiczenia.</p>	<p>Środowisko zintegrowane Free Pascal. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.</p>	<p>Zwrócić uwagę na to, że zmienna typu string może pomieścić maksymalnie 255 znaków. Jeśli wiadomo, że będzie ich mniej, należy ograniczyć ją w deklaracji, używając po słowie <i>string</i> nawias kwadratowy</p>

					zmiennej łańcuchowej; – stosuje standardowe funkcje i procedury Pascala działające na stringach.			z ilością znaków.
Procedury – piszemy własne podprogramy (21)	1	5.22, 5.23	Poznanie konstrukcji i wykorzystania procedur.	Poznanie budowy procedury i sposobów jej wywołania. Definiowanie procedur. Wywołanie procedur bez parametrów i z parametrami.	Uczeń: – zna budowę procedury; – omawia istotę deklaracji wewnętrznych zmiennych i stałych, typów i podprogramów lokalnych; – układa i wywołuje procedury z parametrami i bez parametrów.	Dyskusja i wykład ilustrowany przykładami procedur.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że procedurę wywołuje się przez podanie jej nazwy.
Procedury: piszemy własne podprogramy –zmienne lokalne i globalne, przestanianie zmiennych (22, 23)	2	5.22, 5.23	Poznanie zmiennych lokalnych o globalnych w odniesieniu do procedur. Zapoznanie z metodami przekazywania parametrów do podprogramów.	Umiejętność odróżniania zmiennych, stałych, typów i podprogramów lokalnych od globalnych. Zastosowanie zmiennych lokalnych i globalnych w programowaniu procedur. Poznanie metody przekazywania parametrów bez	Uczeń: – prawidłowo układa procedury wykorzystujące globalne i lokalne zmienne; – stosuje unikalne nazwy zmiennych wewnątrz bloku; – umie określić różnice pomiędzy przekazywaniem parametrów przez wartość i	Dyskusja i wykład ilustrowany przykładami procedur. Ćwiczenia z układania programów z procedurami.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że zmienne zadeklarowane w procedurze lub funkcji nazywa się lokalnymi i nie używa ich program główny.

				var – przez wartość. Poznanie metody przekazywania parametrów z var – przez zmienną.	zmienną; – układa procedury z przekazywaniem parametrów na oba sposoby; – wyjaśnia mechanizmy obu sposobów przekazywania parametrów w procedurze.			
Funkcje – piszemy własne podprogramy (24)	1	5.22, 5.23	Stosowanie funkcji w programach.	Poznanie budowy składniowej funkcji i jej charakterystycznych cech. Sposoby wywoływania funkcji. Znaczenie parametru formalnego.	Uczeń: – określa przypadki, w których niezbędne jest zastosowanie funkcji; – układa kod funkcji; – wywołuje funkcje; – analizuje działanie funkcji.	Wykład ilustrowany przykładami procedur. Ćwiczenia z układania programów z funkcjami.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że funkcje pod swoją nazwą zwracają obliczoną wartość, a procedury tego nie robią.
Złożone struktury danych (rekordy) – grupujemy dane (25)	1	5.22, 5.23	Poznanie zasad i celu grupowania danych.	Poznanie pojęcia rekordu. Poznanie struktury rekordu. Tworzenie rekordu.	Uczeń: – definiuje rekord i typ rekordowy; – podaje przykłady rekordów i typów rekordowych; – deklaruje rekordy składające się z różnych danych.	Wykład ilustrowany przykładami procedur. Ćwiczenia z układania struktur złożonych.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że rekordem nazywa się zarówno strukturę, jak i zmienną typu rekordowego.
Złożone struktury danych –	1	5.22,	Stosowanie	Definiowanie	Uczeń:	Praca	Środowisko	Zwrócić

zmienne rekordowe (26)		5.23	zmiennych rekordowych	zmiennych rekordowych. Struktura tablicowa zmiennych rekordowych. Poznanie pojęcia deskryptora pola.	– definiuje zmienne rekordowe; – podaje przykłady zmiennych rekordowych; – posługuje się deskryptorami pól; – podaje przykłady odwołania się do pola.	indywidualna z podręcznikiem – ćwiczenie na ocenę.	zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	uwagę na rolę kropki w odwoływaniu się do pól rekordu.
Operacje wejścia i wyjścia – zapoznujemy się z plikami (27)	1	5.22, 5.23	Poznanie struktury i dostępu do pliku.	Zdefiniowanie operacji wejścia wyjścia. Poznanie struktury pliku tekstowego i położenia elementu w pliku. Sposoby dostępu do elementu w pliku.	Uczeń: – opisuje strukturę pliku; – określa położenie elementu w pliku; – opisuje sekwencyjny dostęp do elementów pliku i dostęp swobodny do plików.	Dyskusja i wykład ilustrowany przykładami procedur. Ćwiczenia.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że do elementów pliku jest dostęp sekwencyjny
Operacje wejścia i wyjścia – wykonujemy działania na plikach (28)	1	5.22, 5.23	Poznanie podstaw obsługi plików w językach programowania.	Poznanie i stosowanie etapów działań z plikiem. Definiowanie typu pliku. Kojarzenie zmiennych plikowych z nazwą pliku. Tworzenie i otwieranie pliku.	Uczeń: – omawia poszczególne etapy działań na plikach; – definiuje typy plikowe; – deklaruje zmienną plikową w bloku deklaracji; – kojarzy zmienną plikową	Wykład ilustrowany przykładami procedur. Ćwiczenia z układania programów z procedurami.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników	Zwrócić uwagę na poszczególne etapy tworzenia pliku.

					z nazwą pliku; – otwiera i tworzy pliki.		ćwiczenia.	
Operacje wejścia i wyjścia – wykonujemy działania na plikach (29)	1	5.22, 5.23	Opanowanie operacji na plikach.	Poznanie podstawowych operacji zapisu i odczytu. Poznanie działania standardowych procedur dla plików tekstowych. Poznanie innych procedur do obsługi plików. Poznanie dyrektyw kompilatora.	Uczeń: – używa procedury write i writeln z parametrami (zmienną plikową, nazwą zmiennej); – używa procedury read readln z parametrami (zmienną plikową, nazwą zmiennej); – programuje pobranie, modyfikacje i zapis pliku; – odczytuje programowo dane z plików; – prawidłowo zamyka plik procedurą close; – posługuje się procedurą seek i funkcjami filesize, eof, eoln.	Praca indywidualna z podręcznikiem – ćwiczenie na ocenę.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na poszczególne etapy tworzenia pliku.
Wykrywamy błędy (debugowanie) (30)	2	5.22, 5.23	Diagnozowanie błędów w programach komputerowych.	Poznanie podstawowych metod i narzędzi odnajdowania błędów w programach komputerowych. Zapoznanie	Uczeń: – stosuje narzędzie Debugowanie do wykrywania błędów w programach komputerowych;	Dyskusja i wykład ilustrowany przykładami procedur. Ćwiczenia na ocenę.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica	Zwrócić uwagę na skróty klawiszowe i istotę debugowania.

				<p>ssięż pojęciem debugowania i śledzenia krokowego. Użycie tych opcji w środowisku Free Pascal. Zastosowanie pułapek w diagnozowaniu błędów.</p>	<p>– umie kontrolować wartości zmiennych w trakcie debugowania; – uruchamia program metoda krokową, kontrolując zawartości zmiennych; – umie uruchomić debugowanie od dowolnego miejsca programu; – debuguje lub pomija w tym procesie podprogramy; – prawidłowo ustawia pułapki i zna sposób ich wizualizacji w edytorze; – odnajduje błędy za pomocą debugera lub pracy krokowej.</p>		<p>interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.</p>	
<p>Rekurencja – wywołujemy samych siebie, „dziel i zwyciężaj” po raz pierwszy (31)</p>	1	5.22, 5.23	<p>Poznanie i zastosowanie w praktyce pojęcia rekurencji.</p>	<p>Poznanie istoty rekurencji i omówienie przykładów. Realizacja programowa algorytmów zawierających rekurencje. Poznanie</p>	<p>Uczeń: – zna i definiuje pojęcie autowywoływania podprogramu, rekurencji i iteracji, podaje przykłady; – układa program z</p>	<p>Metoda problemowa.</p>	<p>Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-</p>	<p>Zwrócić uwagę na to, że procedury i funkcje są rekurencyjne, gdy w ich definicjach występuje odwołanie</p>

				przykładu metody „dziel i zwyciężaj”.	zastosowaniem iteracji, obliczający silnie i dokładnie objaśnia jego działanie jako funkcji; – wie, jakie rodzaje zmiennych powstają w czasie wywoływania procedury lub funkcji; – wyjaśnia zasadę dostępu do poszczególnych zestawów zmiennych w procesach wykorzystujących rekurencję; – wyjaśnia pojęcie poziomu zagłębienia w procesach rekurencyjnych.		learningowa do przesłania wyników ćwiczenia.	do samych siebie.
Modularyzacja programu – grupujemy podprogramy (32)	1	5.22, 5.23	Sprawne i poprawne grupowanie własnych funkcji i tworzenie modułów predefiniowanych.	Poznanie i wykorzystanie modułów predefiniowanych, ich części i kolejności ich wykonywania.	Uczeń: – zna budowę i składnię modułu; – charakteryzuje części publiczną, implementacyjną i inicjującą modułu; – kompiluje moduły; – układa programy	Wykład ilustrowany przykładami procedur. Ćwiczenia z grupowania podprogramów. Ćwiczenia oceniane.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e.-learningowa do przesłania	Zwrócić uwagę na to, że część implementacyjna modułu jest widoczna tylko w samym module.

					korzystające z modułów.		wyników ćwiczenia.	
II. Dynamiczne struktury danych								
W świecie wskaźników (1)	1	5.22, 5.14	Poznanie typu wskaźnikowego oraz jego zastosowania.	Poznanie sposobu odwołania się do zmiennych dynamicznych za pomocą zmiennych wskaźnikowych. Definiowanie i przykłady zastosowania typu wskaźnikowego i wskaźników.	Uczeń: – wie jakie są mechanizmy odwoływania się do zmiennych dynamicznych za pomocą wskaźników; – podaje przykłady takich zastosowań typów wskaźnikowych.	Wykład ilustrowany przykładami procedur. Ćwiczenia ze wskaźnikami. Ćwiczenia oceniane.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Podkreślić, że wartości typu wskaźnikowego nazywa się wskaźnikami.
Deklarujemy zmienne wskaźnikowe (2)	1	5.22, 5.14	Opanowanie prawidłowej deklaracji zmiennej wskaźnikowej.	Deklaracja zmiennych wskaźnikowych	Uczeń: – deklaruje zmienne wskaźnikowe; – w swoich programach umie odwołać się do zmiennej dynamicznej za pomocą wskaźników.	Praca zespołowa nad programem ze zmiennymi wskaźnikowymi.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że w definicji typu wskaźnikowego można użyć nazwy typu, który nie został jeszcze zdefiniowany.
Tworzymy pierwsze zmienne dynamiczne (3)	1	5.22, 5.14	Tworzenie zmiennych dynamicznych w języku Pascal.	Poznanie procedur tworzenia zmiennych dynamicznych oraz funkcji działających na tych zmiennych.	Uczeń: – tworzy zmienne dynamiczne za pomocą procedury new; – posługuje się funkcjami assigned oraz	Analiza przypadku i ćwiczenia z edytorem Free Pascal dotyczące zmiennych dynamicznych.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna.	Zwrócić uwagę na znaczenie stałej nil.

					dispose; – wie, jak działają powyższe funkcje i procedura.		Platforma e-learningowa do przesłania wyników ćwiczenia.	
Co jeszcze powinieneś wiedzieć o wskaźnikach? (4)	2	5.22, 5.14	Unikanie błędów w definiowaniu i używaniu wskaźników.	Poznanie najczęściej stosowanych błędów w użyciu wskaźników w odniesieniu do zmiennych dynamicznych.	Uczeń: – umie analizować błędy powstałe podczas tworzenia programu ze zmiennymi dynamicznymi; – wie, czym jest spowodowana utrata dostępu do zmiennej dynamicznej.	Ćwiczenia z podręcznikiem i plikami dołączonymi do podręcznika.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na możliwość zagubienia zmiennej dynamicznej j tworzonej wcześniej.
Dynamiczne struktury danych (5)	1	5.22, 5.14	Poznanie istoty i zastosowania dynamicznych struktur danych.	Poznanie pojęcia <i>pole wskaźnikowe</i> i jego funkcji w dynamicznej strukturze danych. Poznanie graficznej metody początkującego programisty na prawidłowe programowanie dynamicznych struktur.	Uczeń: – zna znaczenie rekordu w tworzeniu struktur dynamicznych jako zmiennej dynamicznej; – wie, czym jest zmienna wskaźnikowa i umie opisać jej znaczenie w dynamicznej strukturze danych.	Wykład ilustrowany przykładami dynamicznych struktur danych. Analiza działających fragmentów kodu programu.		Zwrócić uwagę na to, że za pomocą zmiennych wskaźnikowych tworzy się i usuwa zmienne dynamiczne .
Dynamiczne struktury danych – zapis kodu (6)	1	5.22, 5.14, 5.25	Poznanie sposobu definiowania i deklaracji zmiennej	Posługiwanie się zmiennymi dynamicznymi i ich łączenie.	Uczeń: – definiuje i deklaruje w programie struktury	Wykład ilustrowany przykładami dynamicznych struktur danych.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów.	Zwrócić uwagę na wypełnianie wszystkich pól struktury

			wskaźnikowej.		dynamiczne, – łączy zmienne dynamiczne, – używa zmiennej wskaźnikowej do wypełniania pola rekordu dynamicznego.	Analiza działających fragmentów kodu programu.	Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	dynamiczne j w trakcie jej konstruowania.
Stos – ostatni wchodzi, pierwszy wychodzi (LIFO) (7)	1	5.22, 5.14, 5.25	Poznanie funkcji i znaczenia stosu oraz jego zastosowania w programowaniu.	Praktyczne wykorzystanie w programach właściwości stosu do wstawiania i usuwania zmiennych dynamicznych.	Uczeń: – zna budowę stosu i jego zastosowanie; – wie, jak zbudować stos dla zmiennych dynamicznych; – zna funkcję zmiennej wskaźnikowej w adresowaniu wierzchołka stosu; – wie, jak przeglądać stos i jak usuwać zmienne dynamiczne.	Wykład ilustrowany przykładami stosu. Ćwiczenia podręcznikiem.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Pamiętać o przesunięciu wierzchołka na aktualną górę stosu po usunięciu elementu stosu.
Kolejka – pierwszy wchodzi, pierwszy wychodzi (FIFO) (8)	1	5.22, 5.14, 5.25	Poznanie funkcji i znaczenia kolejki oraz jej zastosowania w programowaniu.	Tworzenie kolejek zmiennych dynamicznych, wstawianie do nich elementów i ich usuwanie.	Uczeń: – wie, czym różni się FIFO od LIFO; – umie tworzyć w swoich programach kolejkę zmiennych; – opisuje rolę wskaźników i zmiennych wskaźnikowych	Wykład ilustrowany przykładami kolejek. Ćwiczenia z podręcznikiem.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników	Zwrócić uwagę na to, że operację dodawania elementu kolejki wykonuje się na końcu kolejki.

					w tworzeniu i obsłudze kolejek; – wie, na czym polega wstawianie elementów do kolejki, jej przeglądanie i usuwanie elementów.		ćwiczenia.	
Lista jednokierunkowa (9)	1	5.22, 5.14, 5.25	Praktyczne wykorzystanie list jednokierunkowych – stosu i kolejki.	Analiza programów operujących na listach jednokierunkowych. Poznanie list uporządkowanych.	Uczeń: – zna dokładnie różnice pomiędzy kolejką a stosem; – umie porządkować listy jednokierunkowe	Wykład ilustrowany przykładami list jednokierunkowych. Ćwiczenia ze wskaźnikami. Ćwiczenia oceniane.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na zapamiętanie elementów listy znajdujących się za usuwanymi elementami.
Lista dwukierunkowa, cykliczna (10)	1	5.22, 5.14, 5.25	Poznanie funkcji i znaczenia listy dwukierunkowej jej zastosowania w programowaniu.	Definiowanie rekordu dla listy dwukierunkowej i operacje na jej elementach.	Uczeń: – wskazuje różnicę pomiędzy listą jedno- i dwukierunkową; – umie definiować w programie listy dwukierunkowe; – wstawia i usuwa elementy z listy dwukierunkowej; – wymienia cechy listy	Wykład ilustrowany przykładami list dwukierunkowych. Ćwiczenia ze wskaźnikami. Ćwiczenia oceniane.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że w liście dwukierunkowej pierwszy element wskazuje także na ostatni.

					cyklicznej; – analizuje przykład programu z listą cykliczną.			
Drzewo (11)	1	5.22, 5.14, 5.25	Poznanie funkcji i znaczenia drzewa oraz jego zastosowania w programowaniu.	Budowanie struktur drzewiastych i analogie do struktur folderów w systemie operacyjnym.	Uczeń: – definiuje rekordy dla struktury drzewiastej; – opisuje pojęcia <i>poddrzewo</i> , <i>węzeł</i> , <i>drzewo binarne</i> , <i>głębokość i wysokość drzewa</i> ; – omawia znaczenie i pełnioną funkcję zmiennej wskaźnikowej w drzewie wskaźnikowym;; – analizuje program wykorzystujący strukturę drzewiaste.	Ćwiczenia z podręcznikiem i plikami dołączonymi do podręcznika.	Środowisko zintegrowane Free Pascal. Przykłady kodów programów. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że drzewo jest strukturą rekurencyjną.
III. Bazy danych								
Tabele, wiersze i klucze (1)	1	2.1, 2.2, 2.3, 5.1	Przypomnienie, utrwalenie i poznanie nowych właściwości tabel.	Analiza przykładu tabel utworzonych na podstawie zadania pod kątem ich budowy i elementów z których się składają.	Uczeń: – wybiera odpowiednie nazwy dla pól tabeli; – operuje słownictwem znamionym dla baz danych takim jak <i>rekordy</i> , <i>pola</i> , <i>klucze</i> ;	Wykład z prezentacją i przykładami.	Plik z założeniami do bazy danych użytej w ćwiczeniach. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do	Zwrócić szczególną uwagę na budowę tabeli i ilość jej pól.

					<ul style="list-style-type: none"> – wskazuje pola, które jednoznacznie identyfikują rekordy, np. PESEL, numer, telefonu itp.; – prawidłowo ustala klucze i identyfikatory dla tabel. 		przesłania wyników ćwiczenia.	
Projektujemy bazę danych Pierwsza i druga postać normalna. (2)	1	2.1, 2.2, 2.3, 5.1	Poznanie pojęcia pierwszej i drugiej postaci normalnej tabel bazy danych.	Zapoznanie się z procesem normalizacji tabel (dla 1. i 2. postaci normalnej) i rolą różnych rodzajów kluczy w zależności od postaci tabeli.	<p>Uczeń:</p> <ul style="list-style-type: none"> – rozumie pojęcie normalizacji i opisuje ten proces; – wskazuje różnice pomiędzy 1. a 2. postacią normalną; – ustala klucz główny dla 1. postaci normalnej tabeli; – zna zależności pomiędzy polami niekluczowanymi a kluczem głównym; – wskazuje miejsca, w których następuje redundancja danych; – umie przeprowadzić proces normalizacji do 	Metoda projektowa oparta o zadanie z podręcznika.	Przykłady tabel w 1. i 2. postaci normalnej. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że proces normalizacji eliminuje redundancję.

					2. postaci normalnej; – omawia rolę klucza głównego w 2. postaci normalnej.			
Projektujemy bazę danych. Trzecia i czwarta postać normalna. (3)	1	2.1, 2.2, 2.3, 5.1	Poznanie pojęcia trzeciej i czwartej postaci normalnej tabel bazy danych.	Poznanie procesu normalizacji do postaci 3. i 4. oraz wpływu tego procesu na projektowanie baz danych.	Uczeń: – wskazuje różnice pomiędzy 3. a 4. postacią normalną; – normalizuje tabele do postaci 3. i 4.	Metoda projektowa oparta na zadaniu z podręcznika. Kontynuacja ćwiczenia.	Przykłady tabel w 3. i 4. postaci normalnej. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że możliwe jest od razu sprowadzenie tabeli do postaci 4.
Projektujemy bazę danych. Określamy relacje między tabelami. (4)	1	2.1, 2.2, 2.3, 5.1	Określanie relacji między tabelami w procesie projektowania bazy danych.	Ustalanie prawidłowych i skutecznych relacji między tabelami bazy. Rola kluczy w ustalaniu relacji. Integralność bazy danych	Uczeń: – odróżnia relacje 1–1 od 1–n; – określa prawidłowe relacje pomiędzy tabelami, używając kluczy; – wykazuje, że po poprawnym procesie normalizacji tabele są powiązane prawidłowymi relacjami; – umie określić integralność bazy danych na podstawie analizy tabel.	Metoda projektowa oparta na zadaniu z podręcznika. Kontynuacja ćwiczenia.	Przykłady relacji pomiędzy tabelami. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że po procesie normalizacji tabele są ze sobą powiązane relacjami.
Pierwsze chwile z bazą	1	2.1,	Wykorzystanie	Zapoznanie się z	Uczeń:	Metoda	Program	Omówić

danych – programy do tworzenia baz danych (5)		2.2, 2.3, 5.1	zdołanych umiejętności i wiedzy do budowy bazy danych w aplikacji bazodanowej.	kreatorem baz danych programu OpenOffice.org Base.	– umie uruchomić program do tworzenia relacyjnych baz danych, np. OpenOffice.org Base, i zna rozmieszczenie opcji menu; – umie posługiwać się kreatorem; – zna i omawia funkcję obiektów głównego ekranu programu OpenOffice Org Base.	projektowa oparta na zadaniu z podręcznika. Kontynuacja ćwiczenia z wykorzystaniem programu OpenOffice.org Base lub LibreOffice.org Base.	OpenOffice.org Base lub LibreOffice.org Base. Przykłady baz utworzonych za pomocą tego programu. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	wszystkie obiekty występujące w bazach danych takich jak OpenOffice.org Base.
Tworzymy tabele (6)	1	2.1, 2.2, 2.3, 5.1	Prawidłowe tworzenie tabel za pomocą kreatora.	Użycie kreatora do tworzenia tabel. Prawidłowe określanie typów pól tabel. Określanie kluczy. Użycie formularzy do wprowadzania danych do tabel.	Uczeń: – umie uruchomić i posługiwać się kreatorem tabel programu do tworzenia baz danych np. OpenOffice.org Base.	Metoda projektowa oparta na zadaniu z podręcznika. Kontynuacja ćwiczenia z wykorzystaniem programu OpenOffice.org Base lub LibreOffice.org Base.	Program OpenOffice.org Base lub LibreOffice.org Base. Przykłady tabel tworzonych za pomocą tego programu. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Podkreślić znaczenie kreatora.

<p>Tworzymy i modyfikujemy tabele (7)</p>	<p>1</p>	<p>2.1, 2.2, 2.3, 5.1</p>	<p>Prowadzenie modyfikacji tabel z kreatorem oraz tworzenie tabel bez użycia kreatorów.</p>	<p>Ćwiczenie wprowadzania zmian w tabelach. Tworzenie tabel bez użycia kreatorów w systemach tworzenia baz danych, np. OpenOffice.org Base. Ustanawianie kluczy.</p>	<p>Uczeń: – samodzielnie tworzy tabele z użyciem kreatora; – samodzielnie tworzy tabele bez użycia kreatora; – prawidłowo ustala typy pól w tabelach; – modyfikuje nazwy tabel, pól i typów pól; – ustala klucze dla tabel.</p>	<p>Metoda projektowa oparta na zadaniu z podręcznika. Kontynuacja ćwiczenia z wykorzystaniem programu OpenOffice.org Base lub LibreOffice.org Base.</p>	<p>Program OpenOffice.org Base lub LibreOffice.org Base. Przykłady modyfikacji tabel. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.</p>	<p>Zwrócić uwagę na to, że każda tabela powinna mieć swój klucz.</p>
<p>Indeksujemy i określamy relacji (8)</p>	<p>1</p>	<p>2.1, 2.2, 2.3, 5.1</p>	<p>Prawidłowe i skuteczne indeksowanie i wprowadzanie relacji.</p>	<p>Prawidłowe zaplanowanie indeksów i powiązań na podstawie kryteriów wyszukiwania ustalonych w trakcie projektowania bazy. Używanie klucza głównego jako indeksu. Tworzenie nowych indeksów. Ustanawianie relacji pomiędzy tabelami.</p>	<p>Uczeń: – planuje powiązania i indeksy zgodnie z założeniami wyszukiwania danych w bazie; – samodzielnie tworzy indeksy z wykorzystaniem kreatora i bez niego; – samodzielnie ustala i tworzy relacje; – wyjaśnia, w jakim celu wprowadza się indeksowanie i relacje między tabelami.</p>	<p>Metoda projektowa oparta na zadaniu z podręcznika. Kontynuacja ćwiczenia z wykorzystaniem programu OpenOffice.org Base lub LibreOffice.org Base.</p>	<p>Program OpenOffice.org Base lub LibreOffice.org Base. Przykłady indeksowania tabel i relacji między nimi. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.</p>	<p>Zwrócić uwagę na to, że klucze główne są indeksowane automatycznie.</p>

Budujemy kwerendy z kreatora (9)	1	2.1, 2.2, 2.3, 5.1	Zdobycie umiejętności formułowania prawidłowych i skutecznych kwerend za pomocą kreatora.	Użycie opcji kreatora w procesie tworzenia kwerend w tym: wyboru pól, porządku sortowania, warunków przeszukiwania, szczegółów i skrótów, aliasów. Sprawdzenie poprawności działania kwerendy.	Uczeń: – używa kreatora do formułowania kwerend; – świadomie używa wszystkich opcji kreatora w czasie tworzenia kwerendy; – używa odpowiednich słów do wypełniania pól kreatora; – przeprowadza testy poprawności działania kwerendy.	Metoda projektowa oparta na zadaniu z podręcznika. Kontynuacja ćwiczenia z wykorzystaniem programu OpenOffice.org Base lub LibreOffice.org Base.	Program OpenOffice.org Base lub LibreOffice.org Base. Przykłady kwerend. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na znaczenie kwerend dla prawidłowego i skutecznego wyszukiwania danych w bazie.
Budujemy kwerendy bez kreatora (10)	1	2.1, 2.2, 2.3, 5.1	Zdobycie umiejętności formułowania prawidłowych i skutecznych kwerend bez stosowania kreatora.	Tworzenie kwerend z zastosowaniem opcji <i>Utwórz projekt kwerendy</i> dla jednej tabeli lub kilku tabel.	Uczeń: – samodzielnie tworzy kwerendę dla jednej tabeli i większej ilości tabel ze wszystkimi jej cechami bez użycia kreatora.	Metoda projektowa oparta na zadaniu z podręcznika. Kontynuacja ćwiczenia z wykorzystaniem programu OpenOffice.org Base lub LibreOffice.org Base. Ocena wykonanych baz danych i ich zgodności z zadaniem z podręcznika.	Program OpenOffice.org Base lub LibreOffice.org Base. Przykłady kwerend. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że powiązania tabel można dokonać także podczas edycji.

Budujemy kwerendy za pomocą SQL (11)	1	2.1, 2.2, 2.3, 5.1	Użycie języka SQL do tworzenia kwerendy dla bazy danych.	Wykorzystanie edytora programu OpenOffice.org Base do tworzenia kwerendy w języku SQL. Poznanie najważniejszych poleceń języka SQL służących do tworzenia kwerend. Programowanie w SQL wyświetlania danych wyszukanych przez kwerendę.	Uczeń: – posługuje się edytorem SQL z programu OpenOffice.org Base; – układa kwerendę za pomocą poleceń języka SQL; – organizuje wydruk danych z kwerendy za pomocą SQL; – omawia działanie najważniejszych poleceń języka SQL.	Metoda projektowa oparta na zadaniu z podręcznika. Kontynuacja ćwiczenia z wykorzystaniem programu OpenOffice.org Base lub LibreOffice.org Base.	Program OpenOffice.org Base lub LibreOffice.org Base. Przykłady kwerend zbudowanych za pomocą SQL. Slajdy z omówieniem poleceń SQL. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na znaczenie języka SQL.
Kreujemy formularze (12)	1	2.1, 2.2, 2.3, 5.1	Tworzenie formularzy zgodnie z projektem bazy danych.	Wykorzystanie kreatora do tworzenia formularzy. Kształcenie umiejętności korzystania ze stylów.	Uczeń: – korzysta z opcji kreatora formularzy; – umiejętnie wybiera pola formularze; – definiuje podformularze; – prawidłowo wybiera tryb wprowadzania danych; – stosuje style kreatora dla formularzy; – prawidłowo	Metoda projektowa oparta na zadaniu z podręcznika. Kontynuacja ćwiczenia z wykorzystaniem programu OpenOffice.org Base lub LibreOffice.org Base.	Program OpenOffice.org Base lub LibreOffice.org Base. Przykłady różnych formularzy. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do	Zwrócić uwagę na to, że formularze służą do uporządkowania przedstawiania danych.

					określa nazwę formularza.		przesłania wyników ćwiczenia.	
Drukujemy raporty (13)	1	2.1, 2.2, 2.3, 5.1	Wykorzystanie kreatora do generowania raportów.	Prawidłowy wybór pól do raportu i dobór adekwatnego do danych wyglądu raportu.	Uczeń: – używa kreatora i wszystkich jego opcji do tworzenia raportów; – prawidłowo wybiera pola do raportu; – prawidłowo i zgodnie z charakterem danych nazywa pola; – grupuje dane w raporcie; – organizuje sortowanie danych w raporcie; – dobiera odpowiedni wygląd; – drukuje raporty.	Metoda projektowa oparta na zadaniu z podręcznika. Kontynuacja ćwiczenia z wykorzystaniem programu OpenOffice.org Base lub LibreOffice.org Base. Podsumowanie działu z powtórką.	Program OpenOffice.org Base lub LibreOffice.org Base. Przykłady wydruków z kwerend. Rzutnik, duży ekran, np. LCD, lub tablica interaktywna. Platforma e-learningowa do przesłania wyników ćwiczenia.	Zwrócić uwagę na to, że raporty mogą być statyczne i dynamiczne.

W planie nie przewidziano godzin na sprawdzanie wiedzy i umiejętności. Każdy nauczyciel wypracował swoje metody diagnostyczne, a te wymagają różnej ilości godzin i różnią się częstotliwością. Plan przewiduje 57 godzin zajęć. Pozostałe mogą być wykorzystane na sprawdzanie wiedzy i umiejętności. Plan zawiera także propozycję wykorzystania platformy e-learningowej do przesyłania i sprawdzania prac uczniów.