

Przedmiotowy system oceniania

Zawód: Technik Informatyk

Nr programu: 312[01] /T,SP/MENiS/ 2004.06.14

Przedmiot: Programowanie Strukturalne i Obiektowe

Klasa: druga

Dział	Dopuszczający	Dostateczny	Dobry	Bardzo dobry	Celujący
Algorytmika i pseudoprogramowanie	<ul style="list-style-type: none"> zna podstawowe definicje co to jest algorytm i program komputerowy zna sposoby zapisu algorytmu zna bloki algorytmu zapisane w postaci graficznej potrafi odróżnić algorytm liniowy od algorytmu z warunkiem wie co to iteracja potrafi posłużyć programem edukacyjnym ELI do prezentacji algorytmów wie, że można zaprezentować działanie niektórych algorytmów w arkuszu kalkulacyjnym wie co to pseudojęzyk programowania, wie, że można zapisać algorytm pseudojęzyku programowania, wie do czego służy środowisko Algo 	<ul style="list-style-type: none"> wie jaki jest związek między problemem a algorytmem, zna sposoby zapisu algorytmów i potrafi je krótko opisać, potrafi przedstawić algorytm w postaci graficznej zna specyfikację algorytmu, potrafi zapisać algorytm jako listę kroków, zna różnice między algorytmem liniowym a z warunkiem, wie jak wygląda algorytm z warunkiem zagnieżdżonym, potrafi przedstawić algorytm w postaci drzewa decyzyjnego, potrafi wykorzystać program Eliz do sprawdzenia poprawności algorytmu potrafi zaprezentować prosty algorytm w arkuszu kalkulacyjnym potrafi zapisać algorytm w pseudojęzyku programowania, zna środowisko Algo 	<ul style="list-style-type: none"> potrafi dobrze zdefiniować związek pomiędzy algorytmem a problemem, podaje przykłady, zna wszystkie sposoby zapisu algorytmów i potrafi je opisać, potrafi zapisać algorytm w formie opisowej, graficznej, jako lista kroków i jako drzewo decyzyjne, zna podział algorytmów ze względu na ich strukturę, samodzielnie realizuje wcześniej poznane algorytmy w programie Eli(np. rozwiązanie równania kwadratowego, liniowego, wartość bezwzględna) potrafi samodzielnie zaprezentować poznane algorytmy w arkuszu kalkulacyjnym, wie na czym polega pseudoprogramowanie, dobrze zna środowisko Algo potrafi zapisać każdy algorytm w pseudojęzyku programowania, 	<ul style="list-style-type: none"> bardzo dobrze zna pojęcia i definicje związane z algorytmiką i pseudoprogramowaniem bezproblemowo posługuje się wszystkimi metodami zapisu algorytmu, bezproblemowo prezentuje poznane algorytmy przy użyciu programu Eli i arkusza kalkulacyjnego, pisze programy w pseudojęzyku programowania 	<p>Ocenę tę otrzymuje słuchacz, którego wiedza wykracza poza obowiązujący program nauczania, a ponadto spełnia jeden z podpunktów:</p> <ul style="list-style-type: none"> świetnie potrafi przedstawić zadany problem w postaci algorytmu (w dowolnej postaci graficznej, listy kroków itd.) zna przynajmniej jeden język programowania i potrafi zrealizować poznane algorytmy przy jego pomocy

Wprowadzenie do programowania strukturalnego Pascala	<ul style="list-style-type: none"> • wie, że są różne języki programowania, • zna podstawowe pojęcia związane z programowaniem: program komputerowy, języki programowania, kompilator, interpretator, konsolidator, • zna etapy tworzenia programu, • zna edytor środowiska Pascal, • zna podstawowe słowa kluczowe w języku Pascal, • zna podstawowe typy danych w języku Pascal (całkowity, rzeczywisty, znakowy, łańcuchowy, logiczny), • wie, że są instrukcje wyjścia/wejścia, • wie, że są funkcje i procedury, • wie, że są instrukcje warunkowe, iteracyjne i złożone w programach, • wie, że można realizować wcześniej poznane algorytmy przy pomocy języka Pascal 	<ul style="list-style-type: none"> • potrafi wymienić kilka języków programowania, • zna podstawowe pojęcia związane z programowaniem: program komputerowy, języki programowania, kompilator, interpretator, konsolidator, • zna etapy tworzenia programu, • wie na czym polega programowanie strukturalne, • zna edytor środowiska Pascal, • zna podstawowe typy danych w języku Pascal (całkowity, rzeczywisty, znakowy, łańcuchowy, logiczny) i wie kiedy je stosować, • wie, że są funkcje i procedury i wie kiedy je stosować, • wie, że są instrukcje warunkowe, iteracyjne i złożone w programach i wie kiedy je stosować, • wie, że można realizować wcześniej poznane algorytmy iteracyjne i z warunkiem przy pomocy języka Pascal 	<ul style="list-style-type: none"> • zna klasyfikację języków programowania, potrafi wymienić przykłady zastosowania poszczególnych języków programowania, • zna i rozumie pojęcia związane z programowaniem • zna i rozumie etapy tworzenia programu • wie na czym polega programowanie strukturalne, potrafi wymienić języki programowania strukturalnego, • dobrze zna edytor środowiska Pascal, • zna wszystkie typy danych w języku Pascal, odpowiednio je stosuje w trakcie pisania programów, • potrafi programować z użyciem funkcji i procedur, • potrafi programować wykorzystując instrukcje warunkowe i iteracyjne 	<ul style="list-style-type: none"> • biegle klasyfikuje języki programowania, • bezproblemowo potrafi napisać program w języku Pascal z zgodnie z wcześniej poznanymi algorytmami, • bezproblemowo pisze programy świadomie stosując odpowiednie typy danych, • biegle programuje stosując funkcje i procedury, • bezproblemowo wykorzystuje w trakcie programowania instrukcje warunkowe i iteracyjne, • potrafi zrealizować (napisać program) na dowolnie zadany problem 	<p>Ocenę tę otrzymuje słuchacz, którego wiedza wykracza poza obowiązujący program nauczania, a ponadto spełnia jeden z podpunktów:</p> <ul style="list-style-type: none"> • zna przeznaczenie i zastosowanie innych języków programowania nie omawianych na lekcjach • świetnie programuje w Pascalu, zna inne języki programowania i potrafi w nich napisać program • zna różnice pomiędzy programowaniem obiektowym a strukturalnym
---	--	--	--	--	--

Zawód: Technik Informatyk**Nr programu:** 312[01] /T,SP/MENiS/ 2004.06.14**Przedmiot: Programowanie Strukturalne i Obiektowe****Klasa:** trzecia

Dział	Dopuszczający	Dostateczny	Dobry	Bardzo dobry	Celujący
Podprogramy	<ul style="list-style-type: none"> • zapisuje w postaci programu algorytmy iteracyjne, w których nie jest z góry określona liczba kroków iteracji • wie, jak zadeklarować i wywołać funkcję w Turbo Pascalu; • potrafi przejść do trybu graficznego w Turbo Pascalu; • omawia i analizuje wybrane algorytmy sortowania (przez wybór, bąbelkowy, kubełkowy) przygotowane np. w postaci gotowych schematów blokowych lub programów komputerowych • wyjaśnia różnice między iteracją a rekurencją 	<ul style="list-style-type: none"> • definiuje prostą funkcję z jednym parametrem • omawia iteracyjną realizację algorytmów: Euklidesa, generowania liczb Fibonacciego, schemat Hornera; • wyjaśnia metodę „dziel i zwyciężaj” na przykładzie algorytmu znajdowania minimum i maksimum równocześnie • stosuje w programach podstawowe procedury graficzne, korzystając z podręcznika (rysuje punkty, proste i krzywe, wypełnia barwą wyznaczony obszar) • wie, na czym polega rekurencja i podaje przykłady zjawisk rekurencyjnych; 	<ul style="list-style-type: none"> • zna działanie instrukcji iteracyjnych while i repeat; • wyjaśnia, kiedy stosujemy funkcje, a kiedy procedury • potrafi zapisać w języku Turbo Pascal iteracyjne wersje algorytmów, • stosuje procedury graficzne w realizacji złożonych programów, np. animacji; • potrafi zapisać wybrane algorytmy sortowania w postaci programów w języku Turbo Pascal • potrafi zapisać algorytm rekurencyjny w postaci programu 	<ul style="list-style-type: none"> • samodzielnie dobiera odpowiednią instrukcję iteracyjną, zależnie od warunków zadania • pisze trudniejsze programy, definiując w nich odpowiednio funkcje i/lub procedury; • potrafi zapisać w języku Turbo Pascal iteracyjne wersje algorytmów: Euklidesa, generowania liczb Fibonacciego, schemat Hornera i algorytm znajdowania minimum i maksimum równocześnie • potrafi napisać program rysujący wykres funkcji, np. $y=\sin(x)$, spiralę Archimedesesa • potrafi zapisać wybrane algorytmy w postaci programów w języku Turbo Pascal w wersji rekurencyjnej, swobodnie zamienia wersję iteracyjną na rekurencyjną 	Ocenę tę otrzymuje słuchacz, którego wiedza wykracza poza obowiązujący program nauczania.

Typy strukturalne	<ul style="list-style-type: none"> • wie czym jest tablica – jako przykład typu strukturalnego • zna tablice jedno i dwu wymiarowe • zna procedury do wczytywania i wyprowadzania elementów tablicy • potrafi wymienić klasyczne algorytmy działające na tablicach • potrafi zapisać liczby w różnych systemach liczbowych • wie do czego służy Sito Eratostenesa • zna sposoby sortowania tablicy (bąbelkowe, szybkie, przez wstawianie, przez wybór) • zna szyfr cezara 	<ul style="list-style-type: none"> • potrafi przedstawić tablicę jako przykład typu strukturalnego • wie czym jest typ strukturalny • potrafi zastosować procedury do wczytywania i wyprowadzania elementów tablicy • potrafi zastosować algorytmy działające na tablicach • zna metody sortowania tablicy i umie je zastosować (bąbelkowe, szybkie, przez wstawianie, przez wybór) • wykorzystuje tablice tekstowe • potrafi wyszukać i policzyć znaki w tekście 	<ul style="list-style-type: none"> • dobrze rozumie zagadnienia tablic jako przykładu typu strukturalnego • potrafi omówić typ strukturalny i wie kiedy się go stosuje • zna dobrze klasyczne algorytmy działające na tablicach i potrafi je wykorzystać radzi sobie z wyszukiwaniem elementu maksymalnego (minimalnego) w tablicy. • zapisuje liczby w różnych systemach liczbowych • radzi sobie z tablicami tekstowymi (Czy dany ciąg jest palindromem) 	<ul style="list-style-type: none"> • bardzo dobrze rozumie zagadnienia tablic • wykorzystuje w programach tablice i algorytmy przeszukiwania i sortowania elementów tablicy 	<p>Ocenę tę otrzymuje słuchacz, którego wiedza wykracza poza obowiązujący program nauczania, a ponadto spełnia jeden z podpunktów:</p> <ul style="list-style-type: none"> • zna inne niż poznane na zajęciach metody przeszukiwania tablic
--------------------------	---	--	---	---	---

<p style="text-align: center;">Dynamiczne struktury danych</p>	<ul style="list-style-type: none"> • zna typ wskaźnikowy i jego zastosowanie • wie jak zadeklarować zmienne wskaźnikowe • potrafi wykonać podstawowe operacje na wskaźnikach • wie, że można wykorzystać wskaźniki do przeszukiwania tablic • zna stos i kolejkę jako przykład struktury danych • wie, że można tworzyć i przetwarzać dynamiczne struktury danych takie jak (listy drzewa grafy) 	<ul style="list-style-type: none"> • potrafi zdefiniować typ wskaźnikowy • deklaruje zmienne wskaźnikowe • wykonuje podstawowe operacje na wskaźnikach • potrafi zastosować wskaźniki w tablicach • wykorzystuje stos i kolejkę (jako struktury danych) • próbuje tworzyć i przetwarzać dynamiczne struktury danych takie jak: listy, drzewa, grafy • wykonuje losowanie danych 	<ul style="list-style-type: none"> • rozumie i poprawnie deklaruje zmienne wskaźnikowe • poprawnie wykonuje operacje na wskaźnikach • dobrze stosuje wskaźniki w tablicach • potrafi rozwiązać zadanie z wykorzystaniem wskaźników • stosuje stos – odwrotna notacja polska(ONP) • potrafi losować dane i sprawdzać ich poprawność 	<ul style="list-style-type: none"> • bardzo dobrze wie jak wykorzystać poprawnie wskaźniki w odniesieniu do tablic • zna i stosuje dynamiczne struktury danych (stos, kolejka) • bardzo dobrze wykorzystuje tworzenie i przetwarzanie dynamicznych struktur danych takich jak listy, drzewa, grafy • umiejętnie stosuje stos – odwrotna notacja polska(ONP) • bezbłędnie potrafi losować dane i sprawdzać ich poprawność 	<p>Ocenę tę otrzymuje słuchacz, którego wiedza wykracza poza obowiązujący program nauczania.</p>
<p style="text-align: center;">Elementy zaawansowanego programowania w Pascalu 18</p>	<ul style="list-style-type: none"> • zna obsługę kompilatora • wie, że można programować urządzenie peryferyjne • zna podstawowe instrukcje służące do komunikacji z urządzeniami peryferyjnymi • wie czym są rejestry procesora • wie do czego służy debugger • zna definicję złożoności obliczeniowej algorytmu • wie na czym polega optymalizacja programów 	<ul style="list-style-type: none"> • poprawnie posługuje się kompilatorem • potrafi napisać prosty program komunikujący się z urządzeniami peryferyjnymi • zna rejestry procesora • potrafi wykorzystać debugger do wykrycia błędów • wie jak wykonać dokumentację programu 	<ul style="list-style-type: none"> • zna dyrektywy kompilatora. • potrafi programować obsługę urządzeń peryferyjnych (drukarka, skaner) • wykazuje się dobrą znajomością rejestrów procesora • potrafi wykorzystać debugger do wykrywania i usuwania błędów w programach • wie na czym polega złożoności obliczeniowa algorytmów • potrafi zoptymalizować działanie programów • bezproblemowo radzi sobie z wykonaniem dokumentacji programu 	<ul style="list-style-type: none"> • świetnie opanował dyrektywy kompilatora. • potrafi programować obsługę urządzeń peryferyjnych (drukarka, skaner) • bardzo dobra znajomość rejestrów procesora • bezproblemowo posługuje się debuggerem do wykrywania i usuwania błędów w programach • rozumie istotę złożoności obliczeniowej i pamięciowej algorytmów • potrafi zoptymalizować działanie programów • bezproblemowo radzi sobie z wykonaniem dokumentacji programu 	<p>Ocenę tę otrzymuje słuchacz, którego wiedza wykracza poza obowiązujący program nauczania.</p>