Plan wynikowy z wymaganiami edukacyjnymi przedmiotu informatyka dla klasy II liceum ogólnokształcącego i technikum  
w zakresie podstawowym, uwzględniający kształcone umiejętności i treści podstawy programowej

Uwaga!

W planie pominięto podstawowe umiejętności, które uczeń powinien już posiadać, np. zachowywanie plików projektów, wczytywanie dokumentów do edycji i posługiwanie się systemem operacyjnym.

Kryteria danej oceny opracowano z założeniem, że zostały spełnione kryteria ocen niższych.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Temat** | **Ocena dopuszczająca**  **Uczeń:** | **Ocena dostateczna**  **Uczeń:** | **Ocena dobra**  **Uczeń:** | **Ocena bardzo dobra**  **Uczeń:** | **Ocena celująca**  **Uczeń:** |
| **I. Opracowania rozwiązań problemów wybranymi aplikacjami** | | | | | |
| 1. Zespół, czyli realizujemy projekty w chmurze | – omawia budowę przykładowej chmury, wymieniając jej podstawowe składniki. | – przedstawia i omawia przykładowy schemat struktury chmury informatycznej;  – wymienia najczęściej spotykane w chmurach narzędzia (dysk, edytor, arkusz, kalendarz itp.);  – wie, że chmurę można wykorzystać do pracy zespołowej. | – omawia przeznaczenie elementów chmury użytej na zajęciach;  – porównuje działanie programów z chmury z ich odpowiednikami z aplikacji komputerowych;  – posługuje się chmurą w stopniu wystarczającym do wykonywania edycji dokumentu, kopiowania plików itp.;  – podaje, które aplikacje można wykorzystać do organizacji pracy zespołu. | – biegle posługuje się programami i dyskami sieciowymi;  – przygotowuje chmurę do pracy zespołu, konfigurując kalendarz, udostępniając foldery i pliki itp. | – planuje i organizuje pracę zespołu w chmurze;  – posługuje się kilkoma chmurami (np. Google i OneDrive). |
| 2. Tutorial, czyli jak tworzyć pomoce i instrukcje obsługi | – wie, czym są style i szablony;  – zmienia styl w trakcie edycji dokumentu. | – prawidłowo dobiera style do treści zawartej w dokumencie;  – wie, że można samodzielnie definiować style i szablony;  – uruchamia edycję wykorzystującą gotowy szablon oferowany przez edytor. | – dobiera styl i szablon zgodnie z planowaną zawartością i tematem dokumentu;  – na podstawie podręcznika definiuje szablon i styl dokumentu;  – odnajduje w sieci (np. w chmurze) szablony dla różnych dokumentów. | – definiuje szablon i styl dokumentu;  – zgodnie z treścią przyszłego dokumentu projektuje szablon;  – modyfikuje i projektuje nowe style. | – projektuje szablony dla różnych edytorów (np. LibreOffice Draw) i ich używa. |
| 3. Rozbudowana struktura, czyli korzystamy z konspektu w edytorze tekstu | – rozumie i omawia pojęcie konspektu;  – wie, czym jest akapit i jaką pełni funkcję w edycji i formatowaniu tekstu. | – na podstawie podręcznika tworzy konspekt dokumentu;  – na podstawie podręcznika dzieli dokument na sekcje i kolumny. | – samodzielnie tworzy konspekt dokumentu;  – samodzielnie dzieli dokument na sekcje i kolumny;  – tworzy spis treści na podstawie konspektu. | – uzasadnia stosowanie podziału dokumentu tekstowego na sekcje lub kolumny;  – stosuje różny podział na niektórych stronach (sekcjach) dokumentu. | – stosuje podziały i sekcje w różnych edytorach tekstu. |
| 4. Broszura, czyli jak projektować duże dokumenty | – wstawia do dokumentów SmartArt i Kształty. | – uzasadnia wybór danego kształtu lub ilustracji SmartArt;  – na podstawie podręcznika tworzy spisy ilustracji i tabel. | – na podstawie podręcz- nika lub tutoriali zmienia domyślne opcje edytora oraz automatyczne spisy treści, tabel i ilustracji;  – na podstawie podręcznika lub tutoriali aktualizuje spisy po zmianach. | – samodzielnie zmienia domyślne opcje edytora oraz automatyczne spisy treści, tabel i ilustracji;  – samodzielnie aktualizuje spisy po zmianach;  – decyduje, które z opcji domyślnych zmienić i uzasadnia swój wybór. | – spełnia kryteria oceny bardzo dobrej dla innych edytorów (np. LibreOffice). |
| 5. Recenzja, czyli proponujemy poprawki w tekście | – odczytuje i interpretuje dokumenty, w których zastosowano tryb recenzji;  – odczytuje notatki i zaznaczenia w Adobe Acrobat Reader DC. | – na podstawie podręcznika używa opcji Recenzja w Word i wstawiania komentarzy w dokumencie PDF. | – na podstawie podręcznika lub tutoriali używa opcji Recenzja do wpisywania komentarzy w Word oraz opcji wstawiania komentarzy i zaznaczania fragmentów w dokumencie PDF;  – uruchamia opcję śledzenia zmian. | – samodzielnie używa opcji Recenzja do proponowania zmian w dokumencie;  – samodzielnie używa opcji wstawiania komentarzy i zaznaczania fragmentów w dokumencie PDF;  – analizuje wynik działa- nia opcji porównywania dokumentów. | – spełnia kryteria oceny bardzo dobrej dla innych edytorów (np. LibreOffice);  – przedstawia dokument z innego przedmiotu lub projektu, w którym użył opcji Recenzja dla dokumentu Word lub Notatka dla PDF. |
| **II. Rozwijanie kompetencji społecznych** | | | | | |
| 6. Informatyka pokonuje schody, czyli nikt nie powinien być wykluczony | – wymienia cechy aplikacji ratujących życie lub ułatwiających funkcjonowanie osobom niepełnosprawnym. | – znajduje i instaluje wiarygodne aplikacje ratujące życie lub zdrowie;  – wymienia urządzenia peryferyjne do monitorowania parametrów organizmu;  – posługuje się translatorem języków. | – ocenia jakość aplikacji ratujące życie lub zdrowie;  – ocenia jakość aplikacji ułatwiających funkcjo- nowanie osobom niepeł- nosprawnym i starszym;  – wymienia cechy dobrej strony dostosowanej do potrzeb osób niepełnosprawnych;  – używa peryferiów monitorujących stan organizmu. | – dobiera peryferia i aplikacje pomagające w ratowaniu życia lub zdrowia;  – omawia cechy dobranych przez siebie aplikacji i porównuje ich jakość;  – pokazuje przykłady stron przystosowanych do potrzeb osób niepełnosprawnych. | – projektuje strony przystosowane do potrzeb osób niepełnosprawnych;  – korzysta z translatora w trybie offline. |
| 7. Platformy uczą, czyli rola e-learningu w nauce i pracy | – wymienia podstawowe różnice między nauczaniem zdalnym a klasycznym. | – wymienia zalety i wady e-learningu;  – korzysta z gotowych lekcji e-learningowych. | – opisuje przykładową strukturę lekcji e-learningowej;  – opisuje przykładowy cykl nauczania e-learningowego. | – wymienia właściwości wybranego systemu do tworzenia i prowadzenia kursów e-learningowych;  – porównuje zalety i wady e-learningu i nauczania tradycyjnego;  – znajduje w sieci kursy e-learningowe na dany temat i odczytuje, na jakich warunkach można w nich uczestniczyć. | – układa scenariusz lekcji e-learningowej i wskazuje różnice w porównaniu z tradycyjną lekcją. |
| 8. Możesz być administratorem, czyli jak zarządzać platformą e-learningową | – omawia warunki, jakie musi spełniać platforma  e-learningowa. | – uruchamia aplikację Classroom;  – wymienia elementy, które powinny się znaleźć w strukturze kursu e-learningowego;  – na podstawie opisu z podręcznika tworzy lekcje w Classroom. | – projektuje przykładową strukturę kursu e-learningowego;  – z niewielką pomocą podręcznika tworzy lekcję, dodaje temat zajęć w Classroom. | – projektuje strukturę kursu e-learningowego na zadany temat (np. konkursu matematycznego lub tematu informatycznego) i tworzy ją w Classroom;  – testuje poprawność działania kursu. | – układa kurs e-learningowy w innej platformie niż Classroom. |
| 9. Zasoby i testy, czyli wypełniamy kursy treścią | – wymienia rodzaje zasobów, które można umieszczać w kursach e-learningowych. | – objaśnia rolę poszczególnych rodzajów zasobów kursów e-learningowych;  – wymienia rodzaje pytań testowych, jakie mogą się znaleźć w testach online. | – dzieli zasoby kursów według różnych kryteriów (np. funkcji pełnionej w kursie);  – z niewielką pomocą podręcznika umieszcza w Classroom gotowe zasoby edukacyjne i testy;  – zaprasza użytkowników do korzystania z kursu. | – tworzy część zasobów, które umieści w Classroom;  – tworzy krótkie testy w Classroom. | – umieszcza zasoby w innej platformie niż Classroom. |
| **III. Bazy danych** | | | | | |
| 10. Jak gromadzić informacje, czyli komputerowe bazy danych | – podaje przykłady baz danych, z których korzysta (np. w telefonie);  – definiuje pojęcie bazy danych. | – wymienia wszystkie elementy bazy danych opisane w podręczniku. | – omawia przeznaczenie poszczególnych elementów bazy danych. | – opisuje czynności wykonywane w trakcie projektowania bazy danych. | – opisuje zależności między poszczególnymi elementami bazy danych. |
| 11. Tabele i formularze, czyli jak utworzyć bazę danych | – wymienia przykładowe dane, które powinna zawierać baza uczestników projektu. | – uzasadnia wybór danych dla bazy uczestników projektu;  – na podstawie podręcznika tworzy tabele bazy danych zgodnie ze zdefiniowanymi danymi dla uczestników projektu;  – rozumie znaczenie tabel w bazie danych. | – samodzielnie tworzy kolejne tabele projektu;  – korzystając z kreatora odnośników, tworzy je dla pól tabeli, wybierając wskazane w podręczniku;  – umieszcza dane w tabelach;  – omawia znaczenia klucza;  – posługuje się pojęciami związanymi z bazami danych. | – tworzy klucze;  – tworzy odnośniki dla pól tabel;  – tworzy formularze. | – wykonuje czynności opisane dla Access także w innym systemie baz danych (np. LibreOffice Base). |
| 12. Relacje i pytania, czyli jak uczynić bazę użyteczną | – wyjaśnia znaczenie relacji między tabelami bazy. | – wyjaśnia różnice między rodzajami relacji. | – tworzy klucz zewnętrzny;  – używa kreatora relacji między tabelami;  – wypełnia pola tabel powiązanych za pomocą kreatora;  – na podstawie podręcznika tworzy relacje za pomocą narzędzia Relacje. | – samodzielnie tworzy relacje za pośrednictwem kreatora i narzędzia Relacje;  – tłumaczy sens tworzenia relacji i efekt ich działania w przykładzie z ćwiczeń. | – wykonuje czynności opisane dla Access także w innym systemie baz danych (np. LibreOffice Base). |
| 13. Kto pyta nie błądzi, czyli jak korzystać z baz danych Access | – wyjaśnia znaczenie pojęcia *kwerenda*;  – wyjaśnia, na czym polega filtrowanie w informatyce. | – na podstawie podręcznika lub tutoriali tworzy kwerendę, używając kreatora;  – na podstawie podręcznika lub tutoriali modyfikuje kwerendę, dodając filtrowanie;  – na podstawie podręcz- nika lub tutoriali sortuje dane i tworzy raport za pomocą kreatora. | – samodzielnie tworzy kwerendę, używając kreatora;  – samodzielnie modyfikuje kwerendę, dodając filtrowanie;  – samodzielnie sortuje dane i tworzy raport za pomocą kreatora. | – planuje kwerendę i ją opracowuje;  – prawidłowo decyduje o wyborze filtrowania, sposobie sortowania i raportowania;  – wykonuje eksport tabel do innych formatów (np. Excel). | – wykonuje czynności opisane dla Access także w innym systemie baz danych (np. LibreOffice Base). |
| 14. Czy to możliwe, czyli baza danych w arkuszu Excel | – omawia analogie w tworzeniu tabel w Access i Excel. | – na podstawie podręcznika lub tutoriali tworzy tabele w Excel z nagłówkami;  – na podstawie podręcznika lub tutoriali używa formularzy do wprowadzania danych do tabel;  – na podstawie podręcznika lub tutoriali tworzy listę rozwijaną. | – samodzielnie tworzy tabele w Excel z nagłówkami;  – samodzielnie używa formularzy do wprowa- dzania danych do tabel;  – samodzielnie tworzy listę rozwijaną;  – stosuje sortowanie według wskazanych kryteriów;  – wstawia narzędzia do paska Szybki dostęp. | – planuje i wykonuje czynności tworzenia bazy w Excel;  – dobiera kryteria sortowania;  – przenosi tabele z bazy danych do Excel. | – wykonuje czynności opisane dla Excel także w innym arkuszu (np. LibreOffice Calc). |
| 15. Wiedza w sieci, czyli internet mądrych ludzi | – wyszukuje informacje w domyślnej wyszukiwarce przeglądarki internetowej;  – omawia znaczenie zachowania praw autorskich i podstawy licencji CC. | – na podstawie podręcznika zmienia domyślną wyszukiwarkę w przeglądarce Firefox,  – na podstawie podręcznika uszczegółowia dane w wyszukiwarce w celu zwiększenia prawdopodobieństwa dotarcia do szukanej informacji;  – na podstawie podręcznika wyszukuje za pomocą obrazu. | – uzasadnia celowość korzystania z różnych wyszukiwarek;  – pozyskuje informacje z baz danych (np. europeana.eu);  – tłumaczy podstawy i cel działania botów indeksujących;  – tłumaczy na przykładzie konieczność uszczegółowiania zapytań do wyszukiwarki;  – korzysta z zaawansowanych opcji wyszukiwarek;  – ocenia wiarygodność źródeł, wymieniając cechy o niej świadczące. | – planuje użycie odpowiedniej wyszukiwarki;  – świadomie wybiera bazę danych do wyszukiwania konkretnych informacji;  – trafnie dobiera słowa kluczowe zarówno w opcjach głównych, jak i zaawansowanych wyszukiwarek;  – zmienia wyszukiwarki domyślne w różnych przeglądarkach internetowych. | – omawia cechy różnych wyszukiwarek internetowych i ocenia ich przydatność do konkretnego zadania. |
| **IV. Programowanie stron internetowych** | | | | | |
| 16. HTML, czyli przeglądarka interpretuje język programowania stron | – omawia rolę znaczników w języku HTML. | – omawia rolę przeglądarki internetowej w kontekście języka HTML;  – instaluje, uruchamia i korzysta ze wskazanego edytora, np. Notepad+ skonfigurowanego dla HTM5;  – korzysta z opisów najczęściej używanych znaczników (np.  z podręcznika). | – na podstawie podręcznika lub innych wiarygodnych źródeł konfiguruje edytor do układania programów w HTML5;  – na podstawie podręcznika układa proste programy stron, np. wyświetlające tekst z zachowaniem prawidłowej struktury programu;  – samodzielnie objaśnia rolę CSS w projektowaniu wyglądu strony. | – dobiera odpowiedni edytor do edycji programów w HTML5 i uzasadnia swój wybór;  – konfiguruje edytor;  – tworzy proste strony internetowe zawierające sformatowany tekst. | – tworzy rozbudowane strony internetowe w języku HTML5. |
| 17. Budujemy stronę, czyli tabele, listy i inne elementy dobrej strony | – omawia znaczenie list w programie HTML5;  – uruchamia w przeglądarce gotowy program strony w HTML5 np. z przykładu z podręcznika. | – na podstawie podręcznika uruchamia programy z przykładów;  – na podstawie przykładów omawia rolę list, definicji i tabel;  – tłumaczy istotę i przeznaczenie hipertekstu. | – na podstawie podręcznika lub innych wiarygodnych źródeł tworzy w kodzie HTML listy uporządkowane i nieuporządkowane oraz omawia ich znaczenie;  – na podstawie podręcznika lub innych wiarygodnych źródeł tworzy listy definicji z dodanymi nagłówkami;  – na podstawie podręcznika lub innych wiarygodnych źródeł tworzy tabele w HTML. | – samodzielnie tworzy w kodzie HTML listy uporządkowane i nieuporządkowane oraz omawia ich znaczenie;  – samodzielnie tworzy listy definicji z dodanymi nagłówkami;  – samodzielnie tworzy tabele w HTML;  – planuje użycie odpowiednich konstrukcji do projektowanej strony;  – używa atrybutów w konstrukcji hipertekstu. | – projektuje i tworzy rozbudowane strony internetowe w języku HTML5. |
| 18. Tabele i grafika, czyli kolejne składowe stron internetowych w HTML | – wymienia zasady przygotowania grafiki do publikacji na stronie   uwzględnieniem wymiarów i rozdzielczości. | – na podstawie podręcznika wstawia przygotowaną przez nauczyciela grafikę do gotowego kodu strony,  – na podstawie podręcznika tłumaczy znaczenie wymiarów i skalowania dla szybkości wczytywania strony. | – określa parametry zdjęcia, biorąc pod uwagę jego miejsce na stronie;  – zmienia parametry zdjęcia przeznaczonego na stronę za pomocą prostych narzędzi systemu Windows. | – planuje miejsce na grafikę;  – przygotowuje grafikę do publikacji na stronie;  – umieszcza zdjęcia w planowanym miejscu i formacie na stronie w HTML;  – tworzy odnośnik z elementu graficznego umieszczonego na stronie w HTML. | – projektuje i tworzy rozbudowane strony internetowe w języku HTML5 z zastosowaniem elementów graficznych. |
| 19. Składnia stylów, czyli jak CSS pomaga w programowaniu wyglądu strony | – objaśnia rolę CSS w kreowaniu wyglądu strony. | – objaśnia istotę pliku stylów i jego połączenie z HTML;  – objaśnia rolę CSS w projekcie strony. | – na podstawie podręcznika lub innych wiarygodnych źródeł omawia zasady łączenia CSS z HTML;  – na podstawie podręcznika lub innych wiarygodnych źródeł układa proste pliki stylów CSS formatujące sposób wyświetlania tekstu;  – na podstawie podręcznika lub innych wiarygodnych źródeł wyjaśnia znaczenie nagłówków i bloków w stosowaniu CSS;  – na podstawie podręcznika lub innych wiarygodnych źródeł wyjaśnia znaczenie sekcji i selektora. | – samodzielnie omawia zasady łączenia CSS z HTML;  – samodzielnie układa proste pliki stylów CSS formatujące sposób wyświetlania tekstu;  – samodzielnie wyjaśnia znaczenie nagłówków i bloków w stosowaniu CSS;  – samodzielnie wyjaśnia znaczenie sekcji i selektora;  – projektuje użycie CSS do formatowania stylu tekstu na stronie. | – projektuje i tworzy rozbudowane strony internetowe w języku HTML5 z zastosowaniem CSS. |
| 20. Pliki stylów, czyli CSS w akcji | – objaśnia działanie i znaczenie odsyłaczy na stronie w HTML;  – omawia rolę menu na stronie. | – na podstawie podręcznika lub innych wiarygodnych źródeł omawia znaczenie selektorów i ich atrybutów w pliku CSS;  – na podstawie podręcznika lub innych wiarygodnych źródeł omawia pojęcie *walidacja*. | – na podstawie podręcznika lub innych wiarygodnych źródeł tworzy pliki HTML i CSS z menu z przyciskami, wykorzystując przykłady;  – na podstawie podręcznika lub innych wiarygodnych źródeł używa pseudoklas z przykładu do określenia interakcji klawisza z kursorem,  – na podstawie podręcznika lub innych wiarygodnych źródeł rozmieszcza elementy, tworząc prosty layout strony. | – samodzielnie tworzy pliki HTML i CSS z menu z przyciskami, wykorzystując przykłady;  – samodzielnie używa pseudoklas z przykładu do określenia interakcji klawisza z kursorem,  – samodzielnie rozmieszcza elementy, tworząc prosty layout strony;  – zmienia wygląd i atrybuty menu z przykładu;  – zmienia położenie elementów strony;  – stosuje różną orientację elementów menu;  – przeprowadza walidację strony. | – projektuje i tworzy rozbudowane strony internetowe w języku HTML5 z zastosowaniem CSS, zawierające menu i efekty interakcji z kursorem. |
| 21. Treści mogą się zmieniać, czyli elementy dynamiczne na stronie internetowej | – wskazuje elementy istniejących stron, które zmieniają się dynamicznie;  – na podstawie podręcznika wskazuje różnice między statyczną a dynamiczną stroną internetową. | – omawia sposoby wklejania w kod strony gotowych odnośników do elementów dynamicznych;  – na podstawie podręcznika umieszcza w kodzie strony elementy dynamiczne. | – tworzy element dynamiczny zewnętrzny za pomocą darmowego edytora aplikacji (np. learningapps.com);  – pobiera i umieszcza na stronie kod uruchamiający widżety lub aplikacje internetowe z learningapps.com. | – planuje umieszczanie na stronie widżetów i innych elementów dynamicznych;  – dokonuje prawidłowego wyboru widżetów;  – projektuje elementy dynamiczne (np. krzyżówki w learningapps.com). | – projektuje i tworzy strony z własnymi elementami dynamicznymi. |
| 22. Widoczna w internecie, czyli jak opublikować stronę | – omawia funkcje domeny internetowej;  – wskazuje przynajmniej jedną firmę (stronę internetową) zajmującą się rejestracją domen. | – omawia proces wczytywania strony internetowej do przeglądarki i rolę domeny;  – omawia znaczenie niektórych domen (np. .com, .pl, .edu);  – wskazuje, jakie usługi musi uruchomić lub zamówić, by strona była widoczna w sieci. | – na podstawie podręcznika lub innych wiarygodnych źródeł opisuje na przykładzie proces rezerwacji domeny;  – na podstawie podręcznika lub innych wiarygodnych źródeł zakłada konta na darmowym serwerze z usługą hostingową (np. cba.pl);  – na podstawie podręcznika lub innych wiarygodnych źródeł wysyła pliki strony do serwera www. | – samodzielnie opisuje na przykładzie proces rezerwacji domeny;  – samodzielnie zakłada konta na darmowym serwerze z usługą hostingową (np. cba.pl);  – samodzielnie wysyła pliki strony do serwera www;  – sprawdza zajętość domen, którymi jest zainteresowany;  – korzysta z klienta FTP nie tylko do przesyłania plików strony. | – wskazuje, jak zarządzać domeną i usługą hostingową na przykładzie swojego konta. |
| 23. CMS, czyli system zarządzania treścią strony internetowej | – omawia definicję CMS;  – wskazuje, do czego służy CMS i wymienia kilka najpopularniejszych (w tym Joomla i WordPress). | – omawia proces przygotowań do instalacji CMS;  – na podstawie podręcznika lub innych wiarygodnych źródeł instaluje środowisko serwerowe (np. Laragon), a w nim CMS (np. WordPress);  – na podstawie podręcznika lub innych wiarygodnych źródeł loguje się do panelu sterowania CMSem (np. WordPress). | – samodzielnie instaluje środowisko serwerowe (np. Laragon), a w nim CMS (np. WordPress);  – samodzielnie loguje się do panelu sterowania CMSem (np. WordPress). | – dobiera odpowiedni do tematu strony CMS i uzasadnia swój wybór;  – wyjaśnia konieczność instalacji bazy danych dla CMS;  – używa wirtualnego środowiska serwera do testowania działania różnych stron internetowych. | – buduje ciekawe strony internetowe za pomocą różnych CMS. |
| 24. Panel i skórki, czyli tworzymy stronę w CMS | – wyjaśnia znaczenie szablonu (skórki) w systemie CMS;  – przegląda oferowane w sieci skórki dla WordPress. | – odróżnia motywy od szablonów;  – omawia znaczenie i funkcję motywów w szablonach;  – na podstawie podręcznika wypełnia treścią stronę zbudowaną na bazie szablonu. | – na podstawie podręcznika lub innych wiarygodnych źródeł instaluje w CMS wskazany szablon;  – na podstawie podręcznika lub innych wiarygodnych źródeł dobiera motyw i zmienia go w razie potrzeby;  – na podstawie podręcznika lub innych wiarygodnych źródeł modyfikuje motyw skórki;  – na podstawie podręcznika lub innych wiarygodnych źródeł wypełnia treścią stronę, tworząc nowe wpisy;  – na podstawie podręcznika lub innych wiarygodnych źródeł umieszcza na stronie elementy graficzne. | – samodzielnie instaluje w CMS wskazany szablon;  – samodzielnie dobiera motyw i zmienia go w razie potrzeby;  – samodzielnie modyfikuje motyw skórki;  – samodzielnie wypełnia treścią stronę, tworząc nowe wpisy;  – samodzielnie umieszcza na stronie elementy graficzne;  – trafnie dobiera szablon i motyw do treści przyszłej strony i uzasadnia swój wybór;  – administruje stroną, wykorzystując panel sterowania;  – dodaje i usuwa użytkowników CMS i nadaje im uprawnienia;  – umieszcza na stronie filmy i inne elementy multimedialne. | – planuje i realizuje własne projekty stron na bazie CMS. |
| 25. Szybkie i łatwe, czyli programy do tworzenia stron internetowych | – omawia, jak zapisać tekstowy dokument Word w formacie HTML;  – wymienia kilka systemów darmowych blogów. | – na podstawie podręcznika lub innych wiarygodnych źródeł eksportuje do formatu HTML dokumenty Word i Excel;  – na podstawie podręcznika lub innych wiarygodnych źródeł zakłada konto w systemie darmowych blogów wskazanym przez nauczyciela. | – samodzielnie eksportuje do formatu HTML dokumenty Word i Excel;  – samodzielnie zakłada konto w systemie darmowych blogów wskazanym przez nauczyciela;  – formatuje dokumenty Word i Excel w taki sposób, by po eksporcie do HTML powstała estetyczna i funkcjonalna strona;  – wypełnia treścią blog utworzony w darmowym systemie blogów. | – prawidłowo i estetycznie rozmieszcza elementy graficzne, wzory itp. w eksportowanym dokumencie;  – testuje w środowisku serwerowym (np. Laragon) strony utworzone podczas eksportu dokumentów do HTML, zachowując przy tym układ folderów;  – wypełnia ciekawą treścią własnego bloga. | – prowadzi własnego bloga na ciekawy temat. |
| **V. Algorytmy i programowanie** | | | | | |
| 26. Powtarzanie w pętli i wywołanie siebie, czyli iteracja i rekurencja w algorytmach | – omawia na prawdziwych przykładach różnicę między rekurencją a iteracją. | – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje przykładowe algorytmy iteracyjne i rekurencyjne;  – na podstawie podręcznika lub innych wiarygodnych źródeł rozpoznaje procesy rekurencyjne i iteracyjne. | – samodzielnie analizuje przykładowe algorytmy iteracyjne i rekurencyjne;  – samodzielnie rozpoznaje procesy rekurencyjne i iteracyjne;  – wskazuje w przykładowych algorytmach miejsca, które decydują o iteracyjności lub rekurencyjności opisywanego procesu;  – analizuje przykładowy program. | – układa algorytmy z podejściem iteracyjnym i rekurencyjnym i układa na ich podstawie programy;  – wskazuje instrukcje, które decydują o iteracyjności lub rekurencyjności podejścia do realizacji algorytmu. | – rozwiązuje problemy obiema metodami i ocenia ich skuteczność. |
| 27. Sortowanie bąbelkowe, czyli każda liczba jest mniejsza od maksymalnej lub jej równa | – na podstawie podręcznika lub innych wiarygodnych źródeł omawia budowę tablicy jednowymiarowej;  – na podstawie podręcznika lub innych wiarygodnych źródeł omawia istotę sortowania bąbelkowego. | – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje działanie algorytmu sortowania bąbelkowego w postaci listy kroków i schematu blokowego;  – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje przykład sprawdzający poprawność działania algorytmu. | – omawia istotę metody sortowania bąbelkowego;  – omawia działanie przykładowego algorytmu opartego o metodę sortowania bąbelkowego;  – sprawdza działanie algorytmu na przykładach. | – układa algorytm sortowania bąbelkowego;  – weryfikuje poprawność działania programu na przykładach. | – układa program sortujący metodą bąbelkową w innym języku niż C++ (np. Java). |
| 28. Przez wstawianie, czyli jeszcze o porządkowaniu liczb | – na podstawie podręcznika lub innych wiarygodnych źródeł omawia istotę sortowania „przez wstawianie”. | – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje działanie algorytmu sortowania bąbelkowego w postaci listy kroków i schematu „przez wstawianie”;  – analizuje przykład sprawdzający poprawność działania algorytmu. | – samodzielnie na przykładzie omawia istotę metody sortowania „przez wstawianie”;  – samodzielnie omawia działanie przykładowego algorytmu opartego o metodę sortowania „przez wstawianie”,  – sprawdza działanie algorytmu na przykładach. | – układa algorytm sortowania „przez wstawianie” w postaci listy kroków i schematu blokowego;  – weryfikuje poprawność działania programu na przykładach. | – układa program sortujący metodą „przez wstawianie” w innym języku niż C++ (np. Java). |
| 29. Komputer porządkuje, czyli układamy programy sortujące | – na podstawie podręcznika lub innych wiarygodnych źródeł weryfikuje działanie przykładowych, gotowych programów. | – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje przykładowe programy i wskazuje miejsca, w których są wykonywane najważniejsze dla metody działania. | – na podstawie podręcznika lub innych wiarygodnych źródeł układa programy sortujące;  – na podstawie podręcznika lub innych wiarygodnych źródeł uzasadnia użycie danych instrukcji w przykładowych programach. | – układa programy sortujące według obu metod i weryfikuje poprawność ich działania;  – wskazuje instrukcje i rozkazy realizujące istotę metody. | – proponuje inne od przykładowych rozwiązania programowe;  – układa programy realizujące algorytmy sortowania bąbelkowego i „przez wstawianie” według własnego pomysłu. |
| 30. Fibonacci i jego wzór, czyli generujemy kolejne liczby ciągu | – wie kim był i kiedy żył Fibonacci;  – zna zasługi Fibonacciego dla rozwoju cywilizacji. | – na podstawie podręcznika lub innych wiarygodnych źródeł omawia na gotowym przykładzie istotę ciągu Fibonacciego;  – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje działanie przykładowego algorytmu obliczający kolejne elementy ciągu, zapisanego w postaci schematu blokowego. | – samodzielnie omawia na gotowym przykładzie istotę ciągu Fibonacciego;  – samodzielnie analizuje działanie przykładowego algorytmu obliczający kolejne elementy ciągu, zapisanego w postaci schematu blokowego;  – układa algorytm obliczający określoną liczbę liczb ciągu Fibonacciego;  – analizuje program w języku C++ ułożony według przykładowego algorytmu. | – układa program w języku C++ obliczający *n* kolejnych elementów ciągu Fibonacciego;  – testuje poprawność działania swojego programu na przykładzie. | – proponuje rozwiązanie problemu obliczania *n*-tego elementu ciągu Fibonacciego. |