

PAKIET MATERIAŁÓW DYDAKTYCZNYCH

do kształcenia na odległość –
szkoła ponadpodstawowa (klasa 2)
– informatyka na poziomie rozszerzonym

Województwo Wielkopolskie –
Centrum Doskonalenia Nauczycieli w Koninie

Projekt „Wsparcie placówek doskonalenia nauczycieli i bibliotek pedagogicznych w realizacji zadań związanych z przygotowaniem i wsparciem nauczycieli w prowadzeniu kształcenia na odległość”

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego
Materiał opracowany w ramach grantu „**Razem dla edukacji zdalnej – bezpiecznie, efektywnie, kreatywnie**”

SCENARIUSZ 1 z 1

informatyka (poziom rozszerzony)

SCENARIUSZ ZAJĘĆ DLA: Klasy II szkoły ponadpodstawowej

PROWADZONYCH PRZEZ: Nauczyciela informatyki

TEMAT: „Przetwarzanie plików tekstowych w Javie”

CELE OGÓLNE

Ogólne cele kształcenia zawarte w podstawie programowej informatyki związane z tematem lekcji (podstawa programowa dostępna jest pod adresem internetowym <https://podstawaprogramowa.pl/Liceum-technikum/Informatyka> [dostęp: 9.09.2021]).

- „Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.”
- „Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych układów cyfrowych: układanie i programowanie algorytmów, organizowanie wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.”

CELE SZCZEGÓŁOWE

Szczegółowe cele kształcenia zawarte w podstawie programowej informatyki związane z tematem lekcji (podstawa programowa dostępna jest pod adresem internetowym <https://podstawaprogramowa.pl/Liceum-technikum/Informatyka> [dostęp: 9.09.2021]).

Uczeń:

- „planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania)”;
- „do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych”;

- „objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność”;
- sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów.

Cele kształcenia zakładane przez nauczyciela; uczeń:

- posługuje się zintegrowanym środowiskiem programistycznym do redagowania i uruchamiania programów oraz korzysta z wbudowanej pomocy (podpowiedzi);
- pisze program komputerowy, który wczytuje dane z pliku tekstowego (niezbyt obszernego) i przetwarza poszczególne wiersze (z rekordami danych), dane umieszcza w wybranej strukturze danych (tablica, lista), sposób odczytu danych umożliwia podanie sposobu kodowania znaków w pliku (np. UTF-8);
- przetwarza odczytane z pliku dane (umieszczone w tablicy bądź liście), wykorzystując (w tej fazie nauki) nieskomplikowane algorytmy;
- formatuje dane przy konwersji na obiekt klasy *String* (*String.format*);
- zapisuje dane (wyniki przetwarzania) w pliku tekstowym o określonym kodowaniu znaków (np. UTF-8);
- zna i stosuje różne separatory linii (CR LF – Windows, DOS, LF – Unix, Linux);
- porządkuje (sortuje) listę (lub tabelę) według podanych kryteriów;
- korzysta z klas: *String*, *StringBuffer*, *File*, *Scanner*, *PrintWriter*, *List<T>*, *ArrayList<T>*, *Collections*, *Iterator<T>*, *Locale*, *Collator* oraz tworzy nieskomplikowane własne klasy; implementuje interfejsy *Comparable<T>* oraz *Comparator<T>*.

METODY PRACY

Metodę nauczania nazywam następująco: „monitorowana samodzielna zdalna praca warsztatowa uczniów wsparta wykładem on-line z elementami interakcyjnymi”.

Zalecana liczebność grupy zajęciowej – 5 uczniów, maksymalna liczebność grupy – 8 uczniów.

ŚRODKI DYDAKTYCZNE

- zintegrowane środowisko programistyczne *Eclipse z Javą* (ang. *Eclipse IDE for Java Developers*); bezpłatny program można pobrać ze strony internetowej „Eclipse Foundation” dostępnej pod linkiem <https://www.eclipse.org/downloads/packages/release/2021-09/r/eclipse-ide-java-developers> [dostęp: 9.09.2021];
- platforma Teams lub komunikator Google Meet.

PRZEWIDYWANY CZAS

6 godzin lekcyjnych (w tym 3 godziny samodzielnej pracy uczniów pod nadzorem nauczyciela).

PROPONOWANY PRZEBIEG ZAJĘĆ

CZYNNOŚCI NAUCZYCIELA	CZYNNOŚCI UCZNIÓW
Część 1. Czas trwania – 3 godz. lekcyjne	
CZĘŚĆ WSTĘPNA	
Powitanie uczniów. Upewnia się, że jest dobrze słyszany i widziany. Zwraca uwagę, że wiele programistycznych zadań maturalnych podpada pod schemat „wczytaj dane z pliku tekstowego, przetwórz je, wyniki zapisz w pliku tekstowym”.	Upewniają nauczyciela o dobrej jakości połączenia internetowego: widzialności i słyszalności.
CZĘŚĆ WŁAŚCIWA	
Przesyła uczniom plik z treścią zadania (treść zdania – załącznik 1.) za pomocą funkcji przesyłania plików. Prosi uczniów o jego pobranie i zapisanie na swoim komputerze oraz otwarcie. Czyta treść zadania i obdarza komentarzem – formułuje <i>specyfikację programu</i> .	Pobierają plik i otwierają treść zadania. Czytają tekst zadania i słuchają komentarzy nauczyciela.

Prosi uczniów o uruchomienie zintegrowanego środowiska programistycznym *Eclipse z Javą*.

Udostępnia uczniom swój ekran i również uruchamia zintegrowane środowisko programistyczne *Eclipse z Javą*.

Tworzy projekt Javy o nazwie „OdczytPrzetwarzanieZapis”, a następnie klasę Javy o nazwie „PrzetworzDane”.

Umieszcza w folderze projektu (znajdującym się w folderze *eclipse-workspace*) plik z danymi *dane.txt*.

Redaguje w wolnym tempie program stanowiący rozwiązanie zadania (tekst programu – załącznik 2.).

Podczas wprowadzania tekstu programu komentuje i objaśnia:

1. w jaki sposób półautomatycznie zaimportować potrzebne pakiety z klasami,
2. wykorzystanie automatycznych podpowiedzi;
3. użycie obiektu klasy *File* z obiektem klasy *Scanner* z uwzględnieniem sposobu kodowania znaków w pliku tekstowym;
4. tworzenie nowej klasy *Rekord* przechowującej dane z jednego wiersza pliku z danymi do programu i implementującej interfejs *Comparable<T>*;
5. sposób organizacji iteracji z użyciem metod klasy *Scanner*;
6. budowanie tekstu za pomocą obiektu i metod klasy *StringBuffer*;

Uruchamiają zintegrowane środowisko programistyczne *Eclipse z Javą*.

Tworzą projekt Javy i klasę – o podanych nazwach.

Redagują program. Ze zrozumieniem, aktywnie wsluchują się w objaśnienia i komentarze nauczyciela.

Zadają pytania dotyczące programu.

<p>7. jak zaimplementować interfejs <i>Comparable<T></i>;</p> <p>8. sposób konwersji danych z formatowaniem na obiekty klasy <i>String</i> (<i>String.format</i>);</p> <p>9. wykorzystanie iteratora (klasa <i>Iterator</i>) do przeglądania listy i usuwania elementów listy;</p> <p>10. użycie metod statycznych klasy <i>Collections</i> do porządkowania listy;</p> <p>11. zapis tekstu do pliku tekstowego za pomocą metod klasy <i>PrintWriter</i> z wyborem kodowania znaków.</p> <p>Komentuje użycie separatorów linii i pokazuje w programie, jak ustalić i wykorzystać wymagany przez system operacyjny separator linii.</p> <p>Tłumaczy, że ze względu na kształt języka <i>Java</i> (obecność metod-funkcji, brak procedur niefunkcyjnych), opisując program napisany w <i>Javie</i> nie stosujemy określenia „instrukcja wywołania funkcji (metody)” czy „wywołanie funkcji (metody)”, zamiast tego stosujemy określenie „użycie funkcji (metody) do konstrukcji wyrażenia”, „użycie funkcji (metody)”.</p> <p>Odpowiada na pytania uczniów dotyczące redagowanego programu.</p> <p>Wprowadza różnice w objaśnianiu, stosując określenia „użycie funkcji (metody) na rzecz obiektu pewnej klasy”, „użycie funkcji (metody) statycznej”.</p>	
<p>Uruchamia program. Pokazuje wyniki generowane przez program – zarówno te</p>	<p>Uruchamiają program, poprawiają ewentualne błędy.</p>

<p>pojawiające się w konsoli, jaki i w pliku (<i>wyniki.txt</i>).</p> <p>Udziela pomocy uczniom w uruchomieniu programu.</p> <p>Prosi uczniów o udostępnienie (po kolei) ekranów z tekstem programu.</p>	<p>Udostępniają ekran z programem nauczycielowi i uruchamiają program.</p> <p>Demonstrują wyniki generowane przez program.</p>
<p>Podsumowuje tę część zajęć ze wskazaniem na istotne części programu: wprowadzanie danych z pliku i umieszczenie w strukturze danych (liście) przetwarzanie danych i produkcja wyników, zapis wyników do pliku.</p>	<p>Utrwalają zagadnienia poruszane na zajęciach.</p>
<p>Część 2. Czas trwania – 3 godz. lekcyjne</p>	
<p>Poleca uczniom samodzielne zmodyfikowanie programu – treść zadania polecającego modyfikację znajduje się w załączniku 3.</p> <p>Sugeruje użycie w programie klasy <i>Collator</i> oraz interfejsu <i>Comparator<T></i> i implementującej go klasy anonimowej.</p>	<p>Uczniowie przeglądają Internet w celu znalezienia opisu interfejsu <i>Comparator<T></i> i przykładów jego zastosowań.</p> <p>Znajdują w Internecie opis i zastosowanie klasy <i>Collator</i>.</p> <p>Próbują wymyśleć rozwiązanie zadania.</p> <p>Testują swoje pomysły – tworzą kolejne wersje programu, aż do uzyskania poprawnego wyniku (rozwiązania).</p>
<p>Sprawdza zaproponowane rozwiązania uczniów. W przypadku podania nieoptymalnych rozwiązań przez uczniów – prezentuje rozwiązanie wzorcowe.</p>	<p>Udostępniają po kolei ekran nauczycielowi i demonstrują swój program. Omawiają działanie programu.</p>

EWALUACJA ZAJĘĆ

(Sposób sprawdzenia zakładanych osiągnięć ucznia.)

Sprawdzenie osiągnięć uczniów związanych z tematem zajęć następuje w drugiej części zajęć, kiedy uczniowie samodzielnie rozwiązują zadanie. Badaniu podlega

wynik samodzielnej pracy uczniów – program komputerowy o specyfikacji ujętej w treści zdania.

BIBLIOGRAFIA

- Barteczko K., (2015), *Java. Uniwersalne techniki programowania*, Warszawa: Wydawnictwo Naukowe PWIN SA;
- Schildt H., (2020), *Java. Kompendium programisty*, Wydanie XI, Gliwice: Helion SA.

ZAŁĄCZNIKI

(Załączniki stanowią integralną część niniejszego scenariusza.)

- Załącznik 1. Treść zadania „Psy” rozwiązywanego w 1. części zajęć – dostępny on-line pod linkiem <https://cdnkonin.pl/zalaczniki/mk/zalacznik1.pdf> [dostęp: 9.09.2021];
- Załącznik 2. Program – rozwiązanie zadania „Psy” – dostępny on-line pod linkiem <https://cdnkonin.pl/zalaczniki/mk/zalacznik2.pdf> [dostęp: 9.09.2021];
- Załącznik 3. Zadanie do samodzielnego wykonania przez uczniów – modyfikacja programu. Zadanie rozwiązywane w 2. części zajęć – dostępny on-line pod linkiem <https://cdnkonin.pl/zalaczniki/mk/zalacznik3.pdf> [dostęp: 9.09.2021];
- Załącznik 4. Przykładowe rozwiązanie uczniowskie modyfikacji programu – dostępny on-line pod linkiem <https://cdnkonin.pl/zalaczniki/mk/zalacznik4.pdf> [dostęp: 9.09.2021];
- Załącznik 5. Przykładowa zawartość pliku *dane.txt* z danymi do programu – dostępny on-line pod linkiem <https://cdnkonin.pl/zalaczniki/mk/zalacznik5.pdf> [dostęp: 9.09.2021];
- Załącznik 6. Wyniki generowane przez program bez uczniowskiej modyfikacji (plik *wyniki.txt*) – dostępny on-line pod linkiem <https://cdnkonin.pl/zalaczniki/mk/zalacznik6.pdf> [dostęp: 9.09.2021];

- Załącznik 7. Wyniki generowane przez program po uczniowskiej modyfikacji (stosowny fragment zawartości pliku *wyniki.txt*) – dostępny on-line pod linkiem <https://cdnkonin.pl/zalaczniki/mk/zalacznik7.pdf> [dostęp: 9.09.2021].

Opracowanie: Mariusz Kordylewski