

Wymagania edukacyjne i PSO

Przedmiot: Programowanie obiektowe – klasa 4C i 4GH (OOP + RAD)

Kwalifikacja: INF.03 – Tworzenie i administrowanie stronami i aplikacjami internetowymi oraz bazami danych

Rodzaj zajęć: zajęcia praktyczne w pracowni komputerowej

Nauczyciel: Michał Bednarek

1. Cele kształcenia

Uczeń powinien:

- rozumieć działanie wskaźnika **this** i stosować go w praktyce,
- stosować **hermetyzację** w klasach (settery, gettery),
- tworzyć aplikacje obiektowe w konsoli (np. kalendarz wydarzeń, quiz, „niezbędnik wędkarza”),
- rozumieć zasady **dziedziczenia, przesłaniania składowych i polimorfizmu**,
- pisać programy z użyciem **szablonów klas (wzorców)**,
- stosować mechanizmy **obsługi wyjątków (try-catch)**,
- projektować i realizować zadania końcowe w konsoli z wykorzystaniem OOP,
- znać środowisko programistyczne **RAD** i podstawy programowania wizualnego,
- tworzyć proste aplikacje okienkowe („Hello world”, zmiana kolorów, menu, lista zadań, koszyk zakupowy, notatnik),
- wykorzystywać podstawowe kontrolki (Label, Button, Image, ListBox),
- stosować timery, walidację formularzy, obsługę zdarzeń w GUI.

2. Wymagania edukacyjne

Ocena celująca (6)

Uczeń:

- projektuje i implementuje zaawansowane aplikacje obiektowe i graficzne,
- stosuje dziedziczenie wielopoziomowe, polimorfizm, szablony w praktycznych projektach,
- tworzy kompletne aplikacje w RAD z wieloma oknami, menu, walidacją danych i obsługą plików,
- samodzielnie rozszerza projekty o dodatkowe funkcjonalności (np. zapisywanie ustawień, obsługa błędów użytkownika),
- przygotowuje pełną dokumentację (UML, opis, instrukcję obsługi).

Ocena bardzo dobra (5)

Uczeń:

- opanował wszystkie treści programowe,

- samodzielnie realizuje projekty konsolowe i wizualne,
- poprawnie stosuje hermetyzację, dziedziczenie, polimorfizm, szablony i obsługę wyjątków,
- potrafi stworzyć aplikację w RAD z użyciem różnych kontrolerek i elementów GUI,
- kod jest czytelny i dobrze udokumentowany.

Ocena dobra (4)

Uczeń:

- poprawnie wykonuje większość ćwiczeń, niekiedy z pomocą nauczyciela,
- zna i stosuje dziedziczenie, hermetyzację, polimorfizm w prostych programach,
- korzysta z podstawowych elementów RAD (przyciski, pola tekstowe, menu, ListBox),
- rozwiązuje typowe zadania konsolowe i graficzne, ale projekty nie są w pełni dopracowane.

Ocena dostateczna (3)

Uczeń:

- zna podstawowe elementy OOP i GUI, ale stosuje je w ograniczonym zakresie,
- wykonuje proste ćwiczenia (np. klasa z setterami/getterami, aplikacja okienkowa typu „Hello world”),
- potrafi uruchomić i zmodyfikować gotowy projekt, ale z pomocą nauczyciela,
- ma trudności z większymi projektami (np. koszyk zakupowy, notatnik).

Ocena dopuszczająca (2)

Uczeń:

- zna jedynie elementarne pojęcia programowania obiektowego i wizualnego,
- wykonuje bardzo proste zadania przy dużym wsparciu nauczyciela,
- nie potrafi samodzielnie wykorzystać dziedziczenia ani polimorfizmu,
- obsługuje tylko najprostsze elementy środowiska RAD.

Ocena niedostateczna (1)

Uczeń:

- nie opanował podstawowych umiejętności,
- nie potrafi wykonać prostego programu ani w konsoli, ani w GUI,
- nie rozumie idei programowania obiektowego,
- nie przestrzega zasad pracy w pracowni komputerowej.

3. Obszary oceniania

1. **Umiejętności praktyczne** – projekty obiektowe w konsoli i aplikacje wizualne w RAD.
2. **Dokumentacja i jakość kodu** – czytelność, komentarze, nazewnictwo, UML.
3. **Postawa i aktywność** – praca na lekcji, systematyczność, przestrzeganie zasad BHP.

4. Kryteria oceniania

- Zadania praktyczne i projekty – **80%**
- Dokumentacja, aktywność i praca na lekcji – **20%**

5. Zasady poprawy i zaliczeń

- Uczeń ma prawo poprawić ocenę ze sprawdzianu lub projektu w terminie ustalonym z nauczycielem.
- Nieobecność na sprawdzianie należy usprawiedliwić i napisać w dodatkowym terminie przewidzianym w Statucie Zespołu Szkół nr 4 w Nowym Sączu
- Brak wykonania zadania praktycznego skutkuje oceną niedostateczną, którą można poprawić.

6. Metody sprawdzania osiągnięć uczniów

- sprawdziany praktyczne,
- ocena programów konsolowych i wizualnych,
- analiza dokumentacji i schematów UML,
- obserwacja pracy ucznia w trakcie zajęć.

7. Szczegółowe wymagania do tematów

Uczeń powinien umieć:

- **OOP w konsoli:**
 - stosować wskaźnik `this`, hermetyzację (setter, getter),
 - implementować dziedziczenie, przesłanianie, polimorfizm, szablony klas, obsługę wyjątków,
 - realizować zadania praktyczne (kalendarz wydarzeń, quiz, niezbędnik wędkarza),
- **Projekty końcowe:** napisać program obiektowy w konsoli łączący poznane elementy,
- **RAD (programowanie wizualne):**
 - znać środowisko RAD i jego podstawowe komponenty,
 - tworzyć aplikacje okienkowe z użyciem kontrolerek (Label, Button, Image, ListBox),

- implementować zdarzenia i logikę aplikacji (menu, koszyk zakupowy, notatnik),
- stosować walidację formularzy i obsługę timera.