

Wymagania edukacyjne

Pracowania programowani obiektowego

Klasa II – technik programista

Sposoby sprawdzania i oceniania osiągnięć edukacyjnych uczniów. Ocenie podlega zarówno wiedza teoretyczna, jak i nabyte w trakcie nauki umiejętności.

Oceniane są:

- Ćwiczenia, zadania wykonywane na lekcji
Ocenie podlega: wykonanie wszystkich poleceń zgodnie z treścią, stopień samodzielności wykonywania zadania, pilność, końcowy efekt pracy (jakość pracy), umiejętność pracy w zespole.
- Aktywność podczas pracy na lekcji.
Ocenie podlega: aktywność ucznia w czasie zajęć, stopień zaangażowania podczas wykonywania zajęć, zainteresowanie tematem lekcji.
- Kartkówki, sprawdziany pisemne lub praktyczne.
Ocena z prac pisemnych zgodna ze Statutem Szkoły
- Zadania dodatkowe, prace projektowe

Wymagania na poszczególne oceny				
dopuszczający	dostateczny	dobry	bardzo dobry	celujący
Zasady programowania obiektowego				
Potrafi zainstalować środowisko do programowania obiektowego	Potrafi stosować specyfikatory dostępu	Potrafi utworzyć konstruktor domyślny i kopiujący	Potrafi jak projektować aplikację, stosując dziedziczenie, hermetyzację i polimorfizm	Umie zaprojektować aplikację z wykorzystaniem mechanizmów obiektowych Umie rozwiązać zaawansowane problemy korzystając z

<p>Potrafi utworzyć klasę reprezentującą konkretny byt</p> <p>Potrafi utworzyć obiekt na podstawie klasy</p> <p>Stosuje specyfikatory dostępu: private i public</p> <p>Potrafi zadeklarować klasę bazową i pochodną</p> <p>Potrafi zadeklarować klasę abstrakcyjną i interfejs</p> <p>Potrafi zadeklarować pole statyczne i statyczną metodę</p>	<p>Potrafi zadeklarować przeciążone konstruktory</p> <p>Potrafi pola i metody</p> <p>Potrafi zadeklarować własności, getter oraz setter</p> <p>Deklaruje obiekty reprezentujące klasy</p> <p>Deklaruje wiele obiektów i wywołuje odpowiednie</p> <p>Umie przechwycić prosty wyjątek</p>	<p>Potrafi utworzyć klasy bazowe i pochodne</p> <p>Potrafi odpowiednio wykorzystać specyfikator protected w dziedziczeniu</p> <p>Potrafi zdefiniować abstrakcyjne i metody wirtualne</p> <p>Potrafi wykorzystać interfejsy</p> <p>Umie zdefiniować i wykorzystać składniki statyczne klasy</p> <p>Potrafi umieścić instrukcje generujące wyjątki w bloku try wyjątki i umie je obsłużyć instrukcją catch</p> <p>Umie wykorzystać standardowe klasy szablone</p> <p>Potrafi napisać funkcję wykorzystującą mechanizm przyjaźni (tylko C++)</p>	<p>Potrafi klasy abstrakcyjne do budowy hierarchii klas potomnych</p> <p>Umie zdefiniować i zastosować interfejs jako zbiór zdarzeń</p> <p>Potrafi wykorzystać interfejsy standardowe do wykonania czynności takich jak sortowanie</p> <p>Potrafi zadeklarować klasę ze statycznymi polami i metodami</p> <p>Potrafi zdefiniować wyjątek instrukcją throw</p> <p>Umie przygotować klasę szablonową na przykładzie obsługi stosu lub kolejki</p>	<p>dokumentacji do języka programowania</p>
--	---	---	---	---

