

**PYTANIA NA EGZAMIN Z PRZEDMIOTU:
PROGRAMOWANIE KOMPUTERÓW**

POZIOM PODSTAWOWY

Programowanie proceduralne, obsługa pamięci dynamicznej i złożone typy danych

1. Wyjaśnij różnicę, między obiektami umieszczanymi w pamięci stosu i w pamięci sterty.
2. W jaki sposób w C++ alokujemy i zwalniamy pamięć dynamiczną.
3. Na czym polega „uziemiając wskaźnika” i jaką pełni rolę w programowaniu?
4. Kiedy następuje wyciek pamięci i czym się on objawia?
5. Jakie znasz sposoby na przekazywanie tablic do funkcji?
6. Jakie są reguły przeciążania nazw funkcji?
7. Czym są domyślne wartości argumentów funkcji i jakie są reguły ich używania?
8. Jakie znasz podstawowe operacje bitowe i jak one działają?
9. Czym są maski bitowe i jakie jest ich zastosowanie w programowaniu?
10. Czym są aliasy typów i jakie znasz sposoby ich tworzenia?
11. Czym jest typ wyliczeniowy i jak się z niego korzysta?
12. Czym jest struktura i czym różni się od tablicy?

Podstawy programowania obiektowego

1. Czym jest pole klasy, a czym metoda i jaka jest różnica między klasą a strukturą.
2. Na czym polega zasada enkapsulacji i jak ją realizujemy pisząc klasy?
3. Czym jest konstruktor, destruktor i jakie znasz typy konstruktorów?
4. Czym są składniki statyczne klasy i czym się różnią od składników zwykłych?
5. Czym różni się zwykła zależność od asocjacji?
6. Czym różni się asocjacja od agregacji?
7. Czym różni się agregacja od kompozycji?
8. Czym różni się kompozycja od dziedziczenia?
9. Jakie są sekcje dostępu do składników klasy i jaki wpływają na dziedziczenie?
10. Czym jest klasa abstrakcyjna i jakie ma zastosowanie w programowaniu?
11. Jakie zaznasz sposoby rzutowania danych w C++ i do czego służą?
12. Czym są metody wirtualne i jakie mają zastosowanie w programowaniu obiektowym?

Zarządzanie pamięcią w klasach, programowanie generyczne

1. Jakie znasz mechanizmy kopiowania i kiedy należy je definiować ręcznie?
2. Co jest kluczowym krokiem przy definiowaniu przenoszenia i do czego służy `std::move()`?
3. Czym różni się kopia płytka od kopii głębokiej?
4. Jakie znasz typy referencji w C++ i do czego możemy je przypisać?
5. Co to jest szablon i do czego służy?
6. Co to jest specjalizacja szablonu i jak ją zdefiniować dla funkcji/klasy?
7. Co może być parametrem szablonu?
8. Kiedy wolno nam przeprowadzić kopię binarną, instancji/tablicy, a kiedy musimy użyć kopii element-po- elemencie, lub pole-po-polu.
9. Jakie znasz typy kolekcji dostępne w bibliotekach standardowych C++?
10. Czym są inteligentne wskaźniki i jakie znasz ich rodzaje?
11. Czym są iteratory i do czego ich używamy?
12. Czym jest funktor i jakie ma zastosowanie w pracy z algorytmami standardowymi STL?

POZIOM ZAAWANSOWANY:

1. Omów jakie znasz sposoby alokacji dynamicznych tablic wielowymiarowych, przeprowadź dyskusję ich przydatności i wymień ograniczenia jakie wiążą się z ich używaniem.
2. W jaki sposób możemy przeprowadzić zmianę rozmiaru tablicy alokowanej dynamicznie oraz, jak odczytać rozmiar takiej tablicy. Wyjaśnij jakie mogą być konsekwencje zapisu poza zakresem przydzielonej pamięci alokowanej dynamicznie.
3. Omów jakie znasz zastosowania aliasów typów i na czym polega pośrednie upraszczanie definicji typów. Jaki są relacje między aliasami typów, a automatyczną dedukcją typów. Wyjaśnij jak działają: auto i i decltype i jakie są przesłanki do ich używania.
4. Omów działanie maszyny stanów i wskaż jakie są kluczowe aspekty projektowania takich maszyn w ujęciu programistycznym. Jakie znasz przykłady zastosowania maszyny stanów?
5. Wymień i krótko omów wszystkie poznane rodzaje polimorfizmu w C++. Wyjaśnij fundamentalną różnicę między polimorfizmem statycznym, a dynamicznym. Co jest kluczowym aspektem zadziałania polimorfizmu obiektowego w C++?
6. Przeprowadź dyskusję użyteczności akcesorów w programowaniu obiektowym. Jakie znasz sposoby definiowania seterów i w jakich przypadkach użyjesz poszczególnych typów?
7. Wymień poznane ci zastosowania dla składników statycznych klasy w programowaniu obiektowym i przedstaw bardziej szczegółowo jedno z nich.
8. Podaj przykłady realizacji zasady DRY przy tworzeniu implementacji klas i wyjaśnij jaka jest fundamentalna korzyść ze stosowania się do tej zasady w programowaniu obiektowym.
9. Czym jest klasa czysto statyczna i jaka jest jej przewaga nad zwykłą przestrzenią nazw. Podaj przykład zastosowania takiej klasy w programowaniu?
10. Przeprowadź dyskusję stosowalności różnych sposobów definiowania konwersji typów danych w programowaniu obiektowym w C++ , oraz wyjaśnij jak zachodzą standardowe konwersje między typami danych związanych relacją dziedziczenia.
11. Wyjaśnij czym są jawne i niejawne konwersje typów danych. Przedstaw jakie mogą być okoliczności i konsekwencje zajścia konwersji niejawnych, i jak możemy zapobiegać takim konwersjom.
12. Przedstaw zasady rządzące kolejnością wywoływania konstruktorów klas, gdy te związane są z innym klasami jednocześnie relacją kompozycji i dziedziczenia. Jaką rolę pełni lista inicjalizacyjna konstruktora?
13. Omów w jaki sposób i w jakiej kolejności zachodzi inicjalizacja pól klasy i jak możemy na nią wpływać. Ponadto wyjaśnij jak możemy zainicjalizować inicjalizujemy stałe pola klasy i do czego służy specyfikator mutable.
14. Wyjaśnij dokładnie jak działa relacja przyjaźni, kiedy jest uzasadnione jej użycie. Wyjaśnij z jaką fundamentalną zasadą wchodzi ona w konflikt i dlaczego? Spróbuj podać przykład, kiedy relacja przyjaźni do klasy może być uzasadniona?
15. W jaki sposób możemy dostać się do przesłoniętych przez dziedziczenie składników klasy, a jak do odziedziczonych składników prywatnych. Omów poznane sposoby dziedziczenia, oraz w jaki sposób możemy tworzyć od nich wyjątki.
16. Wymień i omów poznane hierarchie dziedziczenia, oraz wyjaśnij na czym polega dziedziczenie wirtualne i w jakich typach hierarchii ma zastosowanie. Wskaż, która klasa odpowiada za wywołanie konstruktora klasy korzenia w hierarchii diamentu?
17. Omów działanie wzorca Copy-On-Write. Jakie typy kopii są w nim używane i w jaki sposób można wykorzystać inteligentne wskaźniki w uproszczeniu implementacji tego wzorca.

18. Wyjaśnij na czym polega refaktoryzacja kodu i jaki jest jej związek z optymalizacją. Omów jaką rolę w przeprowadzaniu refaktoryzacji pełnią testy jednostkowe i podaj przykład jednego wybranego przekształcenia refaktoryzacyjnego
19. Omów działanie systemu sygnałów i slotów w Qt i podaj jakie są sposoby i reguły ich łączenia. Wyjaśnij, jaka jest podstawowa korzyść ze stosowania tego oryginalnego systemu w programowaniu aplikacji graficznych pisanych obiektowo.
20. Omów tryby komunikacji z plikami w C++ i wyjaśnij w jaki sposób, możemy pozycjonować „głowicę komunikacyjną” w pliku. Porównaj przechowywanie danych w pliku na dysku z przechowywaniem ich w pamięci operacyjnej komputera i przedstaw korzyści i problemy związane z oboma sposobami.
21. Omów trójwarstwowy model aplikacji i wyjaśnij jakie korzyści płynną ze stosowania modeli warstwowych w programowaniu obiektowym.
22. Omów podstawowe pojęcia programowania wielowątkowego takie jak: łączenie, rozłączenie, mutex i zakleszczenie. Przeprowadź dyskusję, w jakich sytuacjach użycie programowania wielowątkowego może przyczynić się do wzrostu wydajności obliczeniowej.