Министерство науки и высшего образования РФ  
ФГБОУ ВО Тульский государственный университет  
Институт прикладной математики и компьютерных наук  
Программирование (ДЗ) 3 семестр

Вариант 1

Вопрос №1

Как добавить обработчик события к элементу управления?

1. element.EventName += new EventHandler(MethodName);

2. element.OnEventName += delegate { MethodName(sender, e); };

3. element.HandleEvent("EventName", MethodName);

4. element->EventName += gcnew EventHandler(this, &ClassName::MethodName);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №2

Как решить ромбовидную проблему в C++?

1. Удаление общих предков из иерархии наследования

2. Переименование методов в базовых классах

3. Использование виртуального наследования

4. Использование ключевого слова `override`

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №3

Что такое шаблон в C++?

1. Специальный вид функции, который позволяет создавать классы и функции общего вида

2. Спецификация для определения интерфейса класса

3. Ключевое слово для создания абстрактных классов

4. Механизм для обработки ошибок

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №4

В чем заключается основное отличие между классом и структурой в C++?

1. В структуре все члены имеют доступ по умолчанию public

2. Класс всегда наследуется от структуры

3. Структура не поддерживает полиморфизм

4. В классе нельзя использовать указатели

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №5

Можно ли перегружать операторы в C++?

1. Да, можно перегрузить большинство операторов

2. Нет, операторы являются встроенными и неизменяемыми

3. Можно перегрузить только арифметические операторы

4. Перегрузка возможна только для пользовательских типов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №6

Как решается проблема неоднозначности при множественном наследовании в C++?

1. Использование ключевого слова `override`

2. Использование ключевого слова `virtual`

3. Изменение порядка наследования

4. Использование квалификатора имени базового класса перед методом

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №7

Чем отличается структура от класса в C++?

1. Структура не поддерживает наследование

2. Все члены структуры по умолчанию public, а у класса private

3. В структуре нельзя определять методы

4. Структура предназначена только для хранения данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №8

Какая операция используется для доступа к членам класса через указатель на объект?

1. . (точка)

2. -> (стрелка)

3. :: (диапазон)

4. & (амперсанд)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №9

Какая ситуация приводит к проблеме "алмазного наследования"?

1. Два базовых класса наследуются от одного и того же класса

2. Один базовый класс наследуется от двух других классов

3. Три базовых класса наследуются друг от друга

4. Четыре базовых класса наследуются друг от друга

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №10

Какое ключевое слово используется для объявления класса, который нельзя наследовать?

1. final

2. sealed

3. locked

4. immutable

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №11

Как изменить цвет фона формы в приложении Windows Forms?

1. form->BackColor = Color::White;

2. form.BackgroundColor = Colors.White;

3. form.BackColor = "#FFFFFF";

4. form.Color = White;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №12

В каком случае используется множественное наследование в C++?

1. Когда класс наследует свойства и методы нескольких базовых классов

2. Когда несколько классов наследуют свойства и методы одного базового класса

3. Когда класс создает несколько экземпляров объектов

4. Когда класс содержит несколько методов с одинаковыми именами

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №13

Какое ключевое слово используется для создания приватных членов класса в C++?

1. private

2. hidden

3. secret

4. invisible

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №14

Что такое pure virtual function?

1. Функция, которая не может быть вызвана

2. Функция, которая доступна только через указатель

3. Функция, которая возвращает чистый результат

4. Функция, которая должна быть реализована в производных классах

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №15

Какая ситуация приводит к проблеме "алмазного наследования"?

1. Два базовых класса наследуются от одного и того же класса

2. Один базовый класс наследуется от двух других классов

3. Три базовых класса наследуются друг от друга

4. Четыре базовых класса наследуются друг от друга

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №16

Что такое ромбовидная проблема в контексте множественного наследования?

1. Ситуация, когда два базовых класса имеют общий предок, но каждый из них наследует от него отдельно

2. Ситуация, когда два базовых класса имеют разные имена методов, но одинаковые сигнатуры

3. Ситуация, когда два базовых класса имеют одинаковые имена методов, но разные сигнатуры

4. Ситуация, когда два базовых класса имеют общие данные-члены

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №17

Какое ключевое слово используется для объявления перегруженного оператора в C++?

1. overload

2. operator

3. redefine

4. override

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №18

Какое пространство имен используется для работы с Windows Forms?

1. System.Windows.Forms

2. . Windows.System.Forms

3. Forms.Windows.System

4. System.Console

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №19

Как добавить изображение на форму в приложении Windows Forms?

1. PictureBox^ pictureBox = gcnew PictureBox(); this->Controls->Add(pictureBox);

2. this->pictureBox = new PictureBox(); Controls.Add(pictureBox);

3. PictureBox\* pictureBox = new PictureBox(); Form.Controls.Add(pictureBox);

4. PictureBox pictureBox = new PictureBox(); AddControl(pictureBox);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №20

Что такое шаблон в C++?

1. Специальный вид функции, который позволяет создавать классы и функции общего вида

2. Спецификация для определения интерфейса класса

3. Ключевое слово для создания абстрактных классов

4. Механизм для обработки ошибок

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №21

Какое ключевое слово используется для вызова конструктора базового класса в конструкторе производного класса?

1. super

2. base

3. parent

4. ancestor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №22

Какая директива подключения необходима для использования Windows Forms?

1. #include <windows.h>

2. using namespace System::Windows::Forms;

3. #include <iostream>

4. #include <forms.h>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №23

Что такое copy constructor?

1. ) Конструктор, который создает копию существующего объекта

2. Конструктор, который копирует данные из другого класса

3. Конструктор, который используется для копирования файлов

4. Конструктор, который инициализирует объект значением по умолчанию

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №24

Какое ключевое слово используется для создания приватных членов класса в C++?

1. private

2. hidden

3. secret

4. invisible

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №25

Какое ключевое слово используется для обозначения дружественной функции в C++?

1. buddy

2. friend

3. external

4. related

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №26

Что такое pure virtual function?

1. Функция, которая не может быть вызвана

2. Функция, которая доступна только через указатель

3. Функция, которая возвращает чистый результат

4. Функция, которая должна быть реализована в производных классах

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №27

Как добавить текстовое поле на форму в приложении Windows Forms?

1. TextBox^ textBox = gcnew TextBox(); this->Controls->Add(textBox);

2. this->textBox = new TextBox(); Controls.Add(textBox);

3. TextBox\* textBox = new TextBox(); Form.Controls.Add(textBox);

4. TextBox textBox = new TextBox(); AddControl(textBox);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №28

Какая директива подключения необходима для использования Windows Forms?

1. #include <windows.h>

2. using namespace System::Windows::Forms;

3. #include <iostream>

4. #include <forms.h>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №29

Какое ключевое слово используется для определения конструктора копирования в C++?

1. copy constructor

2. clone

3. duplicate

4. Имя класса с аргументом типа ссылки на тот же класс

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №30

Какое ключевое слово используется для указания множественного наследования в C++?

1. multiple

2. inheritance

3. Запятая (,)

4. Нет специального ключевого слова

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №31

Когда используется static\_cast?

1. Для приведения типов без проверки во время выполнения

2. Для приведения типов с проверкой во время выполнения

3. Для преобразования строк в числа

4. Для изменения значений переменных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №32

Каково основное преимущество множественного наследования в C++?

1. Возможность повторного использования кода из нескольких источников

2. Упрощение структуры программы

3. Улучшение производительности

4. Уменьшение размера исполняемого файла

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №33

Как установить цвет фона формы?

1. form.BackgroundColor = Colors.White;

2. form.BackColor = "#FFFFFF";

3. form->BackColor = Color::White;

4. form.Color = White;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №34

Какое ключевое слово используется для объявления метода, который не может быть переопределен в производных классах?

1. final

2. sealed

3. locked

4. immutable

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №35

Какую роль играет ключевое слово `virtual` при множественном наследовании в C++?

1. Позволяет избежать дублирования данных-членов в производном классе

2. Позволяет вызывать методы базового класса непосредственно

3. Позволяет переопределять методы базового класса в производном классе

4. Позволяет определять абстрактные методы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №36

Что означает ключевое слово `override` в C++?

1. Указывает, что метод переопределяет виртуальный метод базового класса

2. Указывает, что метод является абстрактным

3. Указывает, что метод является конечным и не может быть переопределен

4. Указывает, что метод является дружественным

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №37

Что такое множественное наследование?

1. Наследование нескольких классов друг от друга

2. Создание нескольких экземпляров одного класса

3. Наследование класса от самого себя

4. Наследование одного класса от нескольких базовых классов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №38

Как называется проблема, возникающая при множественном наследовании, когда у двух базовых классов есть методы с одинаковыми именами?

1. Проблема неоднозначности

2. Проблема дублирования

3. Проблема перегрузки

4. Проблема полиморфизма

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №39

Как отобразить сообщение в диалоговом окне?

1. Dialog.Show("Сообщение");

2. Alert("Сообщение");

3. MessageBox::Show("Сообщение");

4. Popup("Сообщение");

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №40

Какое ограничение накладывается на виртуальные базовые классы при множественном наследовании?

1. Они должны иметь одинаковые конструкторы

2. Они должны быть определены только один раз

3. Они не могут содержать виртуальных методов

4. Они не могут содержать данных-членов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утверждено протоколом №2 от 01.11.2024

Министерство науки и высшего образования РФ  
ФГБОУ ВО Тульский государственный университет  
Институт прикладной математики и компьютерных наук  
Программирование (ДЗ) 3 семестр

Вариант 2

Вопрос №1

Что происходит, если в производном классе отсутствует конструктор, а в базовом классе есть конструктор с параметрами?

1. Компилятор автоматически создаст конструктор без параметров

2. Произойдет ошибка компиляции

3. Конструктор базового класса будет вызван автоматически

4. Ничего не произойдет, программа будет работать корректно

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №2

Как задать положение формы на экране в Windows Forms?

1. form.Position = {100, 50};

2. form.SetPosition(100, 50);

3. form.Location = new Point(100, 50);

4. form->Location = Point(100, 50);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №3

Какая операция используется для доступа к членам класса через объект?

1. . (точка)

2. -> (стрелка)

3. :: (диапазон)

4. & (амперсанд)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №4

Можно ли перегружать оператор присваивания в C++?

1. Да, это возможно

2. Нет, это запрещено языком

3. Только для встроенных типов

4. Только для классов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №5

Что такое полиморфизм в контексте ООП?

1. Возможность объекта изменять свой тип во время выполнения

2. Возможность существования различных форм одного и того же объекта

3. Возможность обращения к объектам разных типов через один интерфейс

4. Возможность изменения состояния объекта во времени

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №6

Какое ограничение накладывается на виртуальные базовые классы при множественном наследовании?

1. Они должны иметь одинаковые конструкторы

2. Они должны быть определены только один раз

3. Они не могут содержать виртуальных методов

4. Они не могут содержать данных-членов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №7

Какое ключевое слово используется для определения деструктора класса в C++?

1. destructor

2. ~Имя\_класса

3. delete

4. destroy

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №8

Как решить ромбовидную проблему в C++?

1. Удаление общих предков из иерархии наследования

2. Переименование методов в базовых классах

3. Использование виртуального наследования

4. Использование ключевого слова `override`

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №9

Какое ключевое слово используется для объявления абстрактного класса в C++?

1. abstract

2. pure virtual

3. interface

4. virtual

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №10

Почему важно использовать виртуальный деструктор в базовом классе?

1. Чтобы обеспечить безопасность многопоточного доступа

2. Чтобы оптимизировать производительность программы

3. Чтобы избежать утечек памяти при удалении объектов через указатели на базовые классы

4. Чтобы упростить синтаксис вызова деструктора

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №11

Какое ключевое слово используется для явного вызова конструктора базового класса в конструкторе производного класса?

1. base

2. super

3. Имя базового класса

4. parent

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №12

Можно ли перегружать оператор присваивания в C++?

1. Да, это возможно

2. Нет, это запрещено языком

3. Только для встроенных типов

4. Только для классов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №13

Какое ключевое слово используется для ограничения видимости членов класса внутри производных классов?

1. private

2. protected

3. internal

4. friend

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №14

Как изменить шрифт текста элемента управления?

1. control->Font = gcnew Font("Arial", 12);

2. control.Font = new Font("Arial", 12);

3. control.setFont("Arial", 12);

4. control.Style.Font = "Arial, 12pt";

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №15

Какая директива подключения необходима для использования Windows Forms?

1. #include <windows.h>

2. using namespace System::Windows::Forms;

3. #include <iostream>

4. #include <forms.h>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №16

Как изменить цвет фона формы в приложении Windows Forms?

1. form->BackColor = Color::White;

2. form.BackgroundColor = Colors.White;

3. form.BackColor = "#FFFFFF";

4. form.Color = White;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №17

Какое ключевое слово используется для уничтожения объекта в C++?

1. delete

2. remove

3. erase

4. destroy

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №18

Какие виды конструкторов вы знаете?

1. Пустой, полный, частичный

2. Глобальный, локальный, временный

3. По умолчанию, параметрический, копирующий

4. Основной, вспомогательный, служебный

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №19

Что такое operator overloading?

1. Перегрузка операторов для работы с пользовательскими типами

2. Замена стандартных операторов новыми функциями

3. Преобразование типов данных

4. Добавление новых операторов в язык

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №20

Как вызвать конструктор базового класса при множественном наследовании в C++?

1. `Derived() : Base1, Base2 {}`

2. `Derived() : Base1(), Base2() {}`

3. `Derived(Base1, Base2) {}`

4. `Derived : Base1(), Base2() {}`

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №21

Как передается конкретный тип параметру шаблона класса при создании объекта?

1. int MyClass obj;

2. MyClass<int> obj;

3. MyClass(obj<int>);

4. obj<MyClass<int>>;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №22

Какой тип доступа к членам класса является наиболее ограниченным?

1. public

2. protected

3. private

4. internal

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №23

Что происходит, если в производном классе отсутствует конструктор, а в базовом классе есть конструктор с параметрами?

1. Компилятор автоматически создаст конструктор без параметров

2. Произойдет ошибка компиляции

3. Конструктор базового класса будет вызван автоматически

4. Ничего не произойдет, программа будет работать корректно

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №24

Как задать положение формы на экране в Windows Forms?

1. form.Position = {100, 50};

2. form.SetPosition(100, 50);

3. form.Location = new Point(100, 50);

4. form->Location = Point(100, 50);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №25

В каком случае используется множественное наследование в C++?

1. Когда класс наследует свойства и методы нескольких базовых классов

2. Когда несколько классов наследуют свойства и методы одного базового класса

3. Когда класс создает несколько экземпляров объектов

4. Когда класс содержит несколько методов с одинаковыми именами

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №26

В каком случае используется множественное наследование в C++?

1. Когда класс наследует свойства и методы нескольких базовых классов

2. Когда несколько классов наследуют свойства и методы одного базового класса

3. Когда класс создает несколько экземпляров объектов

4. Когда класс содержит несколько методов с одинаковыми именами

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №27

Какое ключевое слово используется для защиты данных-членов класса от прямого доступа извне?

1. private

2. protected

3. restricted

4. hidden

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №28

Как изменить цвет фона формы в приложении Windows Forms?

1. form->BackColor = Color::White;

2. form.BackgroundColor = Colors.White;

3. form.BackColor = "#FFFFFF";

4. form.Color = White;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №29

Какую роль играет ключевое слово `virtual` при множественном наследовании в C++?

1. Позволяет избежать дублирования данных-членов в производном классе

2. Позволяет вызывать методы базового класса непосредственно

3. Позволяет переопределять методы базового класса в производном классе

4. Позволяет определять абстрактные методы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №30

Как передается конкретный тип параметру шаблона класса при создании объекта?

1. int MyClass obj;

2. MyClass<int> obj;

3. MyClass(obj<int>);

4. obj<MyClass<int>>;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №31

Как добавить изображение на форму в приложении Windows Forms?

1. PictureBox^ pictureBox = gcnew PictureBox(); this->Controls->Add(pictureBox);

2. this->pictureBox = new PictureBox(); Controls.Add(pictureBox);

3. PictureBox\* pictureBox = new PictureBox(); Form.Controls.Add(pictureBox);

4. PictureBox pictureBox = new PictureBox(); AddControl(pictureBox);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №32

Что такое иерархия классов в контексте ООП?

1. Структура данных, представляющая собой дерево

2. Последовательность вызовов функций

3. Система классов, связанных отношениями наследования

4. Список всех методов класса

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №33

Как передается конкретный тип параметру шаблона класса при создании объекта?

1. int MyClass obj;

2. MyClass<int> obj;

3. MyClass(obj<int>);

4. obj<MyClass<int>>;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №34

Как задать положение формы на экране?

1. form->Location = Point(100, 50);

2. form.Location = new Point(100, 50);

3. form.Position = {100, 50};

4. form.SetPosition(100, 50);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №35

Что такое конструктор класса?

1. Функция, которая удаляет объект из памяти

2. Специальный метод, который вызывается автоматически при создании объекта

3. Метод, который возвращает значение типа класса

4. Функция, которая проверяет корректность данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №36

Какое ключевое слово используется для указания, что метод принадлежит классу?

1. self

2. this

3. class

4. member

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №37

Какое ключевое слово используется для определения конструктора копирования в C++?

1. copy constructor

2. clone

3. duplicate

4. Имя класса с аргументом типа ссылки на тот же класс

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №38

Какое ключевое слово используется для создания виртуальной функции в C++?

1. dynamic

2. polymorphic

3. virtual

4. overridable

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №39

Что такое полиморфизм?

1. Способность объектов разных классов реагировать на одни и те же сообщения по-разному

2. Способность класса изменять свое поведение в зависимости от состояния

3. Способность функции принимать разные типы аргументов

4. Способность программы работать на разных платформах

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №40

Какое ключевое слово используется для явного вызова конструктора базового класса в конструкторе производного класса?

1. base

2. super

3. Имя базового класса

4. parent

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утверждено протоколом №2 от 01.11.2024

Министерство науки и высшего образования РФ  
ФГБОУ ВО Тульский государственный университет  
Институт прикладной математики и компьютерных наук  
Программирование (ДЗ) 3 семестр

Вариант 3

Вопрос №1

Чем отличается структура от класса в C++?

1. Структура не поддерживает наследование

2. Все члены структуры по умолчанию public, а у класса private

3. В структуре нельзя определять методы

4. Структура предназначена только для хранения данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №2

В каком случае используется множественное наследование в C++?

1. Когда класс наследует свойства и методы нескольких базовых классов

2. Когда несколько классов наследуют свойства и методы одного базового класса

3. Когда класс создает несколько экземпляров объектов

4. Когда класс содержит несколько методов с одинаковыми именами

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №3

Как изменить шрифт текста элемента управления?

1. control->Font = gcnew Font("Arial", 12);

2. control.Font = new Font("Arial", 12);

3. control.setFont("Arial", 12);

4. control.Style.Font = "Arial, 12pt";

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №4

Как реализуется метод шаблона класса вне тела класса?

1. template MyClass::myMethod<T>() {}

2. template<typename T> MyClass::myMethod() {}

3. template<typename T> void MyClass<T>::myMethod() {}

4. void MyClass::myMethod<T>() {}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №5

Какая операция используется для доступа к членам класса через указатель на объект?

1. . (точка)

2. -> (стрелка)

3. :: (диапазон)

4. & (амперсанд)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №6

Что такое overloading функций?

1. Переопределение функции в производном классе

2. Вызов одной и той же функции несколько раз

3. Изменение поведения функции в зависимости от контекста

4. Определение нескольких функций с одинаковым именем, но разными параметрами

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №7

Как передается конкретный тип параметру шаблона класса при создании объекта?

1. int MyClass obj;

2. MyClass<int> obj;

3. MyClass(obj<int>);

4. obj<MyClass<int>>;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №8

Что такое полиморфизм?

1. Способность объектов разных классов реагировать на одни и те же сообщения по-разному

2. Способность класса изменять свое поведение в зависимости от состояния

3. Способность функции принимать разные типы аргументов

4. Способность программы работать на разных платформах

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №9

Как объявляется полная специализация шаблона класса в C++?

1. template <> class MyClass<int> {};

2. class MyClass<int> {};

3. template<int> class MyClass {};

4. specialization MyClass<int> {}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №10

Для чего нужен виртуальный деструктор?

1. Чтобы правильно удалять объекты производных классов через указатели на базовые классы

2. Чтобы обеспечить работу с виртуальными методами

3. Чтобы избежать дублирования кода в деструкторах

4. Чтобы повысить производительность программы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №11

Что происходит, если в производном классе отсутствует конструктор, а в базовом классе есть конструктор с параметрами?

1. Компилятор автоматически создаст конструктор без параметров

2. Произойдет ошибка компиляции

3. Конструктор базового класса будет вызван автоматически

4. Ничего не произойдет, программа будет работать корректно

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №12

Как задать текст кнопки?

1. button->Text = "Кнопка"

2. SetText("Кнопка");

3. button.Text = "Кнопка";

4. button.SetText("Кнопка");

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №13

Чем отличается структура от класса в C++?

1. Структура не поддерживает наследование

2. Все члены структуры по умолчанию public, а у класса private

3. В структуре нельзя определять методы

4. Структура предназначена только для хранения данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №14

Какое ключевое слово используется для ограничения видимости членов класса внутри производных классов?

1. private

2. protected

3. internal

4. friend

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №15

Как объявляется шаблон класса в C++?

1. template<class T> class MyClass {};

2. class template<T> MyClass {};

3. <T> class MyClass {};

4. MyClass<T> {}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №16

Что такое operator overloading?

1. Перегрузка операторов для работы с пользовательскими типами

2. Замена стандартных операторов новыми функциями

3. Преобразование типов данных

4. Добавление новых операторов в язык

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №17

Как добавить кнопку на форму?

1. Button^ button = gcnew Button(); this->Controls->Add(button);

2. this->button = new Button(); Controls.Add(button);

3. Button\* button = new Button(); Form.Controls.Add(button);

4. . Button button = new Button(); AddControl(button);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №18

Что такое виртуальная функция в C++?

1. Функция, которая может быть вызвана до создания объекта

2. Функция, которая может быть переопределена в производных классах

3. Функция, которая всегда возвращает значение `true`

4. Функция, которая не имеет реализации

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №19

Что означает ключевое слово `override` в C++?

1. Указывает, что метод переопределяет виртуальный метод базового класса

2. Указывает, что метод является абстрактным

3. Указывает, что метод является конечным и не может быть переопределен

4. Указывает, что метод является дружественным

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №20

Что делает деструктор класса?

1. Создает копию объекта

2. Удаляет объект из памяти

3. Инициализирует объект

4. Проверяет состояние объекта

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №21

Как объявляется шаблон функции в C++?

1. generic<T> void func(T param);

2. void template<T> func(T param);

3. T func<T>(T param);

4. template<typename T> void func(T param);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №22

Что такое copy constructor?

1. ) Конструктор, который создает копию существующего объекта

2. Конструктор, который копирует данные из другого класса

3. Конструктор, который используется для копирования файлов

4. Конструктор, который инициализирует объект значением по умолчанию

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №23

Какой оператор используется для вызова базового конструктора в производном классе?

1. super()

2. base()

3. parent()

4. ::BaseClassName()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №24

В каком случае используется множественное наследование в C++?

1. Когда класс наследует свойства и методы нескольких базовых классов

2. Когда несколько классов наследуют свойства и методы одного базового класса

3. Когда класс создает несколько экземпляров объектов

4. Когда класс содержит несколько методов с одинаковыми именами

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №25

Какое ключевое слово используется для запрета дальнейшего наследования от класса в C++?

1. `sealed`

2. `final`

3. `locked`

4. `closed`

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №26

Для чего используются виртуальные функции?

1. Для реализации полиморфизма

2. Для работы с многопоточностью

3. Для оптимизации кода

4. Для повышения безопасности программы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №27

Какой метод используется для отображения формы?

1. Show()

2. Display()

3. Render()

4. View()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №28

Что делает деструктор класса?

1. Создает копию объекта

2. Удаляет объект из памяти

3. Инициализирует объект

4. Проверяет состояние объекта

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №29

Что такое шаблонный класс?

1. Класс, который определяет набор шаблонов для других классов

2. Класс, который создается на основе шаблона

3. Класс, который хранит данные в виде шаблонов

4. Класс, параметры которого задаются при его использовании

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №30

Как добавить изображение на форму в приложении Windows Forms?

1. PictureBox^ pictureBox = gcnew PictureBox(); this->Controls->Add(pictureBox);

2. this->pictureBox = new PictureBox(); Controls.Add(pictureBox);

3. PictureBox\* pictureBox = new PictureBox(); Form.Controls.Add(pictureBox);

4. PictureBox pictureBox = new PictureBox(); AddControl(pictureBox);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №31

Что такое наследование в контексте ООП?

1. Процесс передачи данных между объектами

2. Способность одного класса использовать код другого класса

3. Механизм обработки исключительных ситуаций

4. Процесс создания нового экземпляра класса

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №32

Какое ключевое слово используется для объявления чистого виртуального метода в C++?

1. `= 0`

3. empty

4. void

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №33

Как сделать форму полноэкранной?

1. form.WindowState = WindowState.Maximized;

2. form->WindowState = FormWindowState::Maximized;

3. form.FullScreen = true;

4. form.Maximize();

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №34

Как решить ромбовидную проблему в C++?

1. Удаление общих предков из иерархии наследования

2. Переименование методов в базовых классах

3. Использование виртуального наследования

4. Использование ключевого слова `override`

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №35

Что такое параметр шаблона по умолчанию в C++?

1. Значение, которое будет использоваться, если при вызове шаблона не указан конкретный тип

2. Имя переменной, используемой в шаблоне

3. Тип данных, возвращаемый функцией-шаблоном

4. Класс, от которого наследуется шаблон

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №36

Как изменить шрифт текста элемента управления?

1. control->Font = gcnew Font("Arial", 12);

2. control.Font = new Font("Arial", 12);

3. control.setFont("Arial", 12);

4. control.Style.Font = "Arial, 12pt";

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №37

Как объявляется полная специализация шаблона класса в C++?

1. template <> class MyClass<int> {};

2. class MyClass<int> {};

3. template<int> class MyClass {};

4. specialization MyClass<int> {}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №38

Как установить цвет фона формы?

1. form.BackgroundColor = Colors.White;

2. form.BackColor = "#FFFFFF";

3. form->BackColor = Color::White;

4. form.Color = White;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №39

Что такое наследование в контексте ООП?

1. Процесс передачи данных между объектами

2. Способность одного класса использовать код другого класса

3. Механизм обработки исключительных ситуаций

4. Процесс создания нового экземпляра класса

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №40

Что такое вариативные шаблоны в C++?

1. Шаблоны, которые могут изменяться во времени

2. Шаблоны, которые могут принимать произвольное количество аргументов

3. Шаблоны, которые могут быть использованы только в определенных условиях

4. Шаблоны, которые могут быть применены только к определенным типам данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утверждено протоколом №2 от 01.11.2024

Министерство науки и высшего образования РФ  
ФГБОУ ВО Тульский государственный университет  
Институт прикладной математики и компьютерных наук  
Программирование (ДЗ) 3 семестр

Вариант 4

Вопрос №1

Что такое параметр шаблона по умолчанию в C++?

1. Значение, которое будет использоваться, если при вызове шаблона не указан конкретный тип

2. Имя переменной, используемой в шаблоне

3. Тип данных, возвращаемый функцией-шаблоном

4. Класс, от которого наследуется шаблон

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №2

Какова основная причина использования виртуального наследования в C++?

1. Избежание дублирования данных-членов в производном классе

2. Повышение безопасности программы

3. Увеличение гибкости программы

4. Сокращение времени компиляции

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №3

Что такое composition в контексте ООП?

1. Механизм объединения объектов различных классов в одном объекте

2. Разделение классов на компоненты

3. Создание иерархии классов

4. Компилирование программы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №4

Как объявляется частичная специализация шаблона класса в C++?

1. partial specialization MyClass<T\*> {};

2. template<T\*> class MyClass {};

3. template<typename T> class MyClass<T\*> {};

4. template<pointer> class MyClass {};

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №5

Каково основное отличие между одиночным и множественным наследованием в C++?

1. Количество базовых классов, от которых наследуется производный класс

2. Способ вызова конструкторов базовых классов

3. Способ разрешения конфликтов имен методов

4. Наличие виртуальных методов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №6

Какое ключевое слово используется для создания защищенных членов класса в C++?

1. protected

2. guarded

3. safe

4. secure

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №7

Какое ключевое слово используется для создания виртуальной функции в C++?

1. dynamic

2. polymorphic

3. virtual

4. overridable

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №8

Какова основная причина использования виртуального наследования в C++?

1. Избежание дублирования данных-членов в производном классе

2. Повышение безопасности программы

3. Увеличение гибкости программы

4. Сокращение времени компиляции

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №9

Какой оператор используется для вызова базового конструктора в производном классе?

1. super()

2. base()

3. parent()

4. ::BaseClassName()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №10

Какая операция используется для доступа к членам класса через указатель на объект?

1. . (точка)

2. -> (стрелка)

3. :: (диапазон)

4. & (амперсанд)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №11

Какой метод используется для отображения формы?

1. Show()

2. Display()

3. Render()

4. View()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №12

Как задать текст кнопки?

1. button->Text = "Кнопка"

2. SetText("Кнопка");

3. button.Text = "Кнопка";

4. button.SetText("Кнопка");

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №13

Что такое полиморфизм?

1. Способность объектов разных классов реагировать на одни и те же сообщения по-разному

2. Способность класса изменять свое поведение в зависимости от состояния

3. Способность функции принимать разные типы аргументов

4. Способность программы работать на разных платформах

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №14

Как изменить размер формы?

1. form.Size = new Size(300, 200);

2. setSize(300, 200);

3. form->Size = System::Drawing::Size(300, 200);

4. Resize(300, 200);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №15

Что такое полиморфизм?

1. Способность объектов разных классов реагировать на одни и те же сообщения по-разному

2. Способность класса изменять свое поведение в зависимости от состояния

3. Способность функции принимать разные типы аргументов

4. Способность программы работать на разных платформах

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №16

Что означает ключевое слово virtual перед методом класса?

1. Метод может быть переопределен в производных классах.

2. Метод не может быть вызван напрямую

3. Метод доступен только внутри класса

4. Метод имеет динамическую память

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №17

Какое ключевое слово используется для уничтожения объекта в C++?

1. delete

2. remove

3. erase

4. destroy

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №18

Какое ключевое слово используется для определения оператора присваивания в C++?

1. operator=

2. assign

3. set

4. copy

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №19

Какое ключевое слово используется для вызова конструктора базового класса в конструкторе производного класса?

1. super

2. base

3. parent

4. ancestor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №20

Как установить заголовок формы?

1. form.Title = "Заголовок";

2. form->Title = "Заголовок";

3. setTitle("Заголовок");

4. form.Caption = "Заголовок";

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №21

Какое ключевое слово используется для создания класса в C++?

1. class

2. struct

3. interface

4. object

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №22

Какая цель перегрузки операторов в C++?

1. Упростить синтаксис программы

2. Изменить поведение стандартных операторов

3. Добавить новые операторы в язык

4. Реализовать полиморфизм

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №23

Какое ключевое слово используется для обозначения дружественной функции в C++?

1. buddy

2. friend

3. external

4. related

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №24

Как объявляется шаблон класса в C++?

1. template<class T> class MyClass {};

2. class template<T> MyClass {};

3. <T> class MyClass {};

4. MyClass<T> {}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №25

В каком случае используется множественное наследование в C++?

1. Когда класс наследует свойства и методы нескольких базовых классов

2. Когда несколько классов наследуют свойства и методы одного базового класса

3. Когда класс создает несколько экземпляров объектов

4. Когда класс содержит несколько методов с одинаковыми именами

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №26

Как называется проблема, возникающая при множественном наследовании, когда у двух базовых классов есть методы с одинаковыми именами?

1. Проблема неоднозначности

2. Проблема дублирования

3. Проблема перегрузки

4. Проблема полиморфизма

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №27

Что такое pure virtual function?

1. Функция, которая не может быть вызвана

2. Функция, которая доступна только через указатель

3. Функция, которая возвращает чистый результат

4. Функция, которая должна быть реализована в производных классах

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №28

Как изменить размер формы?

1. form.Size = new Size(300, 200);

2. setSize(300, 200);

3. form->Size = System::Drawing::Size(300, 200);

4. Resize(300, 200);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №29

Что такое конструктор класса?

1. Функция, которая удаляет объект из памяти

2. Специальный метод, который вызывается автоматически при создании объекта

3. Метод, который возвращает значение типа класса

4. Функция, которая проверяет корректность данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №30

Что происходит, если в производном классе отсутствует конструктор, а в базовом классе есть конструктор с параметрами?

1. Компилятор автоматически создаст конструктор без параметров

2. Произойдет ошибка компиляции

3. Конструктор базового класса будет вызван автоматически

4. Ничего не произойдет, программа будет работать корректно

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №31

Как задать текст кнопки?

1. button->Text = "Кнопка"

2. SetText("Кнопка");

3. button.Text = "Кнопка";

4. button.SetText("Кнопка");

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №32

Что такое конструктор класса?

1. Функция, которая удаляет объект из памяти

2. Специальный метод, который вызывается автоматически при создании объекта

3. Метод, который возвращает значение типа класса

4. Функция, которая проверяет корректность данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №33

Как объявляется частичная специализация шаблона класса в C++?

1. template<typename T> class MyClass<T\*> {};

2. partial specialization MyClass<T\*> {};

3. template<T\*> class MyClass {};

4. template<pointer> class MyClass {};

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №34

Что такое параметр шаблона по умолчанию в C++?

1. Значение, которое будет использоваться, если при вызове шаблона не указан конкретный тип

2. Имя переменной, используемой в шаблоне

3. Тип данных, возвращаемый функцией-шаблоном

4. Класс, от которого наследуется шаблон

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №35

Какое ключевое слово используется для указания, что метод не может быть переопределен в производных классах?

1. final

2. sealed

3. locked

4. closed

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №36

Как решается проблема неоднозначности при множественном наследовании в C++?

1. Использование ключевого слова `override`

2. Использование ключевого слова `virtual`

3. Изменение порядка наследования

4. Использование квалификатора имени базового класса перед методом

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №37

Что такое this pointer?

1. Указатель на первый элемент массива

2. Указатель на текущий объект внутри метода класса

3. Указатель на глобальную переменную

4. Указатель на родительский объект

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №38

Что такое pure virtual function?

1. Функция, которая не может быть вызвана

2. Функция, которая доступна только через указатель

3. Функция, которая возвращает чистый результат

4. Функция, которая должна быть реализована в производных классах

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №39

Какой тип доступа к членам класса является наиболее ограниченным?

1. public

2. protected

3. private

4. internal

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №40

Чем отличается структура от класса в C++?

1. Структура не поддерживает наследование

2. Все члены структуры по умолчанию public, а у класса private

3. В структуре нельзя определять методы

4. Структура предназначена только для хранения данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утверждено протоколом №2 от 01.11.2024

Министерство науки и высшего образования РФ  
ФГБОУ ВО Тульский государственный университет  
Институт прикладной математики и компьютерных наук  
Программирование (ДЗ) 3 семестр

Вариант 5

Вопрос №1

Как изменить цвет фона формы в приложении Windows Forms?

1. form->BackColor = Color::White;

2. form.BackgroundColor = Colors.White;

3. form.BackColor = "#FFFFFF";

4. form.Color = White;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №2

Почему важно использовать виртуальный деструктор в базовом классе?

1. Чтобы обеспечить безопасность многопоточного доступа

2. Чтобы оптимизировать производительность программы

3. Чтобы избежать утечек памяти при удалении объектов через указатели на базовые классы

4. Чтобы упростить синтаксис вызова деструктора

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №3

Как задать положение формы на экране?

1. form->Location = Point(100, 50);

2. form.Location = new Point(100, 50);

3. form.Position = {100, 50};

4. form.SetPosition(100, 50);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №4

Как закрыть форму программным способом?

1. Close();

2. Exit();

3. Terminate();

4. Destroy();

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №5

Каковы основные недостатки множественного наследования в C++?

1. Сложность понимания и поддержки кода

2. Увеличение времени компиляции

3. Уменьшение производительности

4. Все вышеперечисленное

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №6

Можно ли перегружать операторы в C++?

1. Да, можно перегрузить большинство операторов

2. Нет, операторы являются встроенными и неизменяемыми

3. Можно перегрузить только арифметические операторы

4. Перегрузка возможна только для пользовательских типов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №7

Что делает деструктор класса?

1. Создает копию объекта

2. Удаляет объект из памяти

3. Инициализирует объект

4. Проверяет состояние объекта

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №8

Какое ключевое слово используется для указания, что метод принадлежит классу?

1. self

2. this

3. class

4. member

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №9

Что такое overloading функций?

1. Переопределение функции в производном классе

2. Вызов одной и той же функции несколько раз

3. Изменение поведения функции в зависимости от контекста

4. Определение нескольких функций с одинаковым именем, но разными параметрами

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №10

Что делает деструктор класса?

1. Создает копию объекта

2. Удаляет объект из памяти

3. Инициализирует объект

4. Проверяет состояние объекта

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №11

Какой тип доступа к членам класса является наиболее ограниченным?

1. public

2. protected

3. private

4. internal

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №12

Каково назначение protected-членов класса?

1. Доступны только внутри класса и его потомков

2. Доступны только вне класса

3. Доступны только внутри класса и его друзей

4. Доступны везде, где есть ссылка на объект

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №13

Что такое интерфейс в C++?

1. Класс, определяющий внешний вид приложения

2. Протокол взаимодействия между модулями программы

3. Спецификация методов класса

4. Абстрактный класс, содержащий только pure virtual методы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №14

Какую роль играет ключевое слово `virtual` при множественном наследовании в C++?

1. Позволяет избежать дублирования данных-членов в производном классе

2. Позволяет вызывать методы базового класса непосредственно

3. Позволяет переопределять методы базового класса в производном классе

4. Позволяет определять абстрактные методы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №15

Какую роль играет ключевое слово `virtual` при множественном наследовании в C++?

1. Позволяет избежать дублирования данных-членов в производном классе

2. Позволяет вызывать методы базового класса непосредственно

3. Позволяет переопределять методы базового класса в производном классе

4. Позволяет определять абстрактные методы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №16

Какая ошибка возникает при попытке создать экземпляр абстрактного класса?

1. Ошибка времени выполнения

2. Компиляционная ошибка

3. Предупреждение компилятора

4. Никакой ошибки нет

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №17

Какие ключевые слова используются для указания типа наследования в C++?

1. `virtual`, `override`, `final`

2. `class`, `struct`, `union`

3. `private`, `protected`, `public`

4. `const`, `mutable`, `volatile`

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №18

Как называются шаблоны, которые могут принимать произвольное количество аргументов?

1. Вариативные шаблоны

2. Полиморфные шаблоны

3. Универсальные шаблоны

4. Множественные шаблоны

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №19

Какое ключевое слово используется для объявления метода, который не может быть переопределен в производных классах?

1. final

2. sealed

3. locked

4. immutable

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №20

Что такое Windows Forms?

1. Библиотека для создания консольных приложений

2. Платформа для разработки веб-приложений

3. Инструмент для работы с базами данных

4. Технология для создания графических пользовательских интерфейсов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №21

Каково основное преимущество множественного наследования в C++?

1. Возможность повторного использования кода из нескольких источников

2. Упрощение структуры программы

3. Улучшение производительности

4. Уменьшение размера исполняемого файла

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №22

Какое ключевое слово используется для определения оператора присваивания в C++?

1. operator=

2. assign

3. set

4. copy

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №23

Что такое overloading функций?

1. Переопределение функции в производном классе

2. Вызов одной и той же функции несколько раз

3. Изменение поведения функции в зависимости от контекста

4. Определение нескольких функций с одинаковым именем, но разными параметрами

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №24

Что такое this pointer?

1. Указатель на первый элемент массива

2. Указатель на текущий объект внутри метода класса

3. Указатель на глобальную переменную

4. Указатель на родительский объект

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №25

Как объявляется шаблон класса в C++?

1. template<class T> class MyClass {};

2. class template<T> MyClass {};

3. <T> class MyClass {};

4. MyClass<T> {}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №26

Как установить заголовок формы?

1. form.Title = "Заголовок";

2. form->Title = "Заголовок";

3. setTitle("Заголовок");

4. form.Caption = "Заголовок";

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №27

Какое ключевое слово используется для защиты данных-членов класса от прямого доступа извне?

1. private

2. protected

3. restricted

4. hidden

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №28

Как объявляется класс, наследующий от двух базовых классов в C++?

1. class Derived : public Base1, public Base2 {};

2. class Derived : Base1, Base2 {};

3. class Derived(Base1, Base2) {};

4. class Derived extends Base1, Base2 {};

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №29

Что такое наследование в контексте ООП?

1. Процесс передачи данных между объектами

2. Способность одного класса использовать код другого класса

3. Механизм обработки исключительных ситуаций

4. Процесс создания нового экземпляра класса

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №30

Что такое специализация шаблона в C++?

1. Объявление новой переменной в шаблоне

2. Создание конкретной версии шаблона для определенного типа данных

3. Переопределение метода в производном классе

4. Создание копии объекта

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №31

Что такое параметр шаблона по умолчанию в C++?

1. Значение, которое будет использоваться, если при вызове шаблона не указан конкретный тип

2. Имя переменной, используемой в шаблоне

3. Тип данных, возвращаемый функцией-шаблоном

4. Класс, от которого наследуется шаблон

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №32

Какая ошибка возникает при попытке создать экземпляр абстрактного класса?

1. Ошибка времени выполнения

2. Компиляционная ошибка

3. Предупреждение компилятора

4. Никакой ошибки нет

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №33

Для чего нужен виртуальный деструктор?

1. Чтобы правильно удалять объекты производных классов через указатели на базовые классы

2. Чтобы обеспечить работу с виртуальными методами

3. Чтобы избежать дублирования кода в деструкторах

4. Чтобы повысить производительность программы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №34

Как добавить изображение на форму в приложении Windows Forms?

1. PictureBox^ pictureBox = gcnew PictureBox(); this->Controls->Add(pictureBox);

2. this->pictureBox = new PictureBox(); Controls.Add(pictureBox);

3. PictureBox\* pictureBox = new PictureBox(); Form.Controls.Add(pictureBox);

4. PictureBox pictureBox = new PictureBox(); AddControl(pictureBox);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №35

Какое ключевое слово используется для определения оператора присваивания в C++?

1. operator=

2. assign

3. set

4. copy

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №36

Какое ключевое слово используется для защиты данных-членов класса от прямого доступа извне?

1. private

2. protected

3. restricted

4. hidden

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №37

Можно ли перегружать оператор присваивания в C++?

1. Да, это возможно

2. Нет, это запрещено языком

3. Только для встроенных типов

4. Только для классов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №38

Какое ключевое слово используется для объявления абстрактного класса в C++?

1. abstract

2. pure virtual

3. interface

4. virtual

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №39

Какое ключевое слово используется для защиты данных-членов класса от прямого доступа извне?

1. private

2. protected

3. restricted

4. hidden

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вопрос №40

Какой оператор используется для вызова базового конструктора в производном классе?

1. super()

2. base()

3. parent()

4. ::BaseClassName()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утверждено протоколом №2 от 01.11.2024

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вариант** | **#1** | **#2** | **#3** | **#4** | **#5** | **#6** | **#7** | **#8** | **#9** | **#10** | **#11** | **#12** | **#13** | **#14** | **#15** | **#16** | **#17** | **#18** | **#19** | **#20** |
| В1 | 4 | 3 | 1 | 1 | 1 | 4 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 2 | 1 | 1 | 1 |
| В2 | 2 | 4 | 1 | 1 | 3 | 2 | 2 | 3 | 2 | 3 | 3 | 1 | 2 | 1 | 2 | 1 | 1 | 3 | 1 | 2 |
| В3 | 2 | 1 | 1 | 3 | 2 | 4 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 1 | 1 | 1 | 2 | 1 | 2 |
| В4 | 1 | 1 | 1 | 3 | 1 | 1 | 3 | 1 | 1 | 2 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 2 | 2 |
| В5 | 1 | 3 | 1 | 1 | 4 | 1 | 2 | 2 | 4 | 2 | 1 | 3 | 4 | 1 | 1 | 2 | 3 | 1 | 1 | 4 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вариант** | **#21** | **#22** | **#23** | **#24** | **#25** | **#26** | **#27** | **#28** | **#29** | **#30** | **#31** | **#32** | **#33** | **#34** | **#35** | **#36** | **#37** | **#38** | **#39** | **#40** |
| В1 | 2 | 2 | 1 | 1 | 2 | 4 | 1 | 2 | 4 | 3 | 1 | 1 | 3 | 1 | 1 | 1 | 4 | 1 | 3 | 2 |
| В2 | 2 | 1 | 2 | 4 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 3 | 2 | 1 | 2 | 2 | 4 | 3 | 1 | 3 |
| В3 | 4 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 4 | 1 | 2 | 1 | 2 | 3 | 1 | 1 | 1 | 3 | 2 | 2 |
| В4 | 1 | 1 | 2 | 1 | 1 | 1 | 4 | 3 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 4 | 2 | 4 | 1 | 2 |
| В5 | 1 | 1 | 4 | 2 | 1 | 2 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 |