**Вопросы**

1. **Для чего нужен конструктор?**

Это специальный метод, который служит для создания объекта (выделения под него памяти, инициализации значений).

1. **Сколько типов конструкторов существует в с++?**

4 типа: конструктор по умолчанию, копирования, перемещения, с параметрами.

1. **Для чего используется деструктор? В каких случаях деструктор описывается явно?**

Деструктор служит для освобождения ресурсов, занятых текущим объектом, вызывается автоматически, когда удаляется объект. Описывается явно для уничтожения объектов, созданных с использованием placement – new или allocator.

1. **Для чего используется конструктор без параметров? Конструктор с параметрами?**

Конструктор без параметров используется для создания пустых объектов для будущего их заполнения, либо для присвоения конкретных значений.

Конструктор с параметрами используют для инициализации полей объекта.

1. **В каких случаях вызывается конструктор копирования?**

Конструктор копирования в C++ используется для копирования данных одного объекта в другой.

1. **Перечислить свойства конструкторов.**

**-** Конструктор не возвращает значения, даже void. Также нельзя получить указатель на конструктор.

**-** Конструкторы можно перегружать, используя разные наборы параметров.

**-** Параметры конструктора могут иметь любой тип, кроме этого же класса.

**-** Если ни один конструктор не создан в коде, компилятор создает его автоматически.

**-** Конструкторы не наследуются.

**7. Перечислить свойства деструкторов.**

- Деструктор ничего не возвращает

- Деструктор не имеет параметров.

- Деструктор называется также, как и тип данных, но с тильдой («~») в начале.

**8. К каким атрибутам имеют доступ методы класса?**

По умолчанию – к атрибутам, описанным в public, однако для доступа к атрибутам в private и protected методы класса должны быть объявлены внутри самого класса или как дружественные функции класса.

**9. Что представляет собой указатель this?**

Указатель this указывает на текущий объект данного класса. С его помощью можно обратиться внутри класса к любым членам этого класса.

**10. Какая разница между методами определенными внутри класса и вне класса?**

Методы, описанные внутри класса имеют доступ к закрытым членам класса, к ним можно обращаться через указатель this, они могут использовать наследование и виртуальные функции.

**11. Какое значение возвращает конструктор?**

Не возвращает

**12. Какие методы создаются по умолчанию?**

Конструктор по умолчанию, конструктор копирования, присваивания, деструктор.

**13. Какое значение возвращает деструктор?**

Не возвращает

**14. Дано описание класса**

**class Student {**

**string name;**

**int group;**

**public:**

**student(string, int);**

**student(const student&)**

**~student();**

**}**

**Какой метод отсутствует в описании класса?**

Отсутствует конструктор копирования

**16. Какой метод будет вызван при выполнении следующих операторов:**

Конструктор с параметрами

**17. Какие методы будут вызваны при выполнении следующих**

**операторов:**

Student s1(“Ivanov”, 20)

Student s2 = s1;

Конструктор с параметрами и копирования

**18. Какие методы будут вызваны при выполнении следующих**

**операторов:**

Student s1(“Ivanov”, 20)

Student s2;

s2 = s1;

Конструктор с параметрами, без параметров и копирования.

**19. Какой конструктор будет использоваться при передаче параметра в**

**функцию print():**

Void print(student a)

{

a.show();

}

Конструктор копирования

**20. Класс описан следующим образом:**

Class Student

{

String name;

Int age;

Public:

Void set\_name(string);

Void set\_age(int);

…

}

Student p;

Каким образом можно присвоить новое значение атрибуту name

объекта p?

При помощи метода set\_name(string name) {this->name = name}

Int main()

{

p.set\_name(name);

}