

# Семестр 2. ОСНОВЫ C++

Лекции[1][3].get() // Принципы ООП

# 3 принципа

- ▶ Инкапсуляция
- ▶ Наследование
- ▶ Полиморфизм

# Перед тем, как приступить к инкапсуляции. Исключения

Исключение - средства обработки нежелательных (иногда непредвиденных) ситуаций для обычной работы программы без «падений». Вступает при возникновении какой-либо ошибки. Такая ошибка может обозначаться как самим разработчиком, так и системно (деление на ноль, переполнение стека и т.д.)

# Синтаксис в C++

Блок обработки ошибок

```
try {  
    //Исполняемый блок, где может возникнуть ошибка  
}  
catch() //Этот блок ловит исключение и  
обрабатывает  
{  
    ...  
}
```

С помощью слова **throw** можно отправить в блок catch исключение и прервать работу блока try.

# Виды блоков catch

- ▶ `catch(type)`

Где `type` - тип исключения

- ▶ `catch(type arg)`

Где `arg` - значимая переменная

- ▶ `catch(...)`

Обработывает все исключения, если остальные не сработали.

# Пример

```
int x = 0;
try {
    cout << 2/x; //Здесь произойдет выброс
    исключения
    // Последующие операторы выполняться не будут
}
catch (...) {
    cout << "Division by zero" << std::endl;
}
```

# Инкапуляция

Это свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе и скрыть детали реализации от пользователя.

Пример - дописать класс DateTime

# Наследование

Это свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым или родительским. Новый класс - потомком, наследником или производным классом.

Например, от класса целого числа можно создать класс вещественного числа.



# СИНТАКСИС

```
class A{
    int x;
public:
    A(){x=0;}
    void print(){cout<<x;}
}
class B: public A{
    int y;
public:
    B(){y=0;}
    void print2(){cout<<y;}
}
```

При наследовании от класса, можно поставить модификатор чтобы определить, как будут наследоваться остальные члены класса предка.

Модификатор в базовом классе	public	protected	private
public	public	protected	private
protected	protected	protected	private
private	private	private	private

# Задание

Создать класс целого числа. От него унаследовать класс вещественного числа. Сократить поля данных, осуществить сложение как для базового, так и для производного класса.