

Карпаев Алексей, м.н.с. лаборатории физиологии человека, Биофармкластер «Северный»

Современные технологии объектно-ориентированного программирования численных методов

Лекция 1

Обзорная

Цель курса

- Недостаток времени на рассмотрение технологий программирования в курсе вычислительной математики
- Показать преимущества объектно-ориентированного подхода перед традиционным процедурным на примерах реализации численных методов:
 - Легкость поддержки программы
 - Возможность повторного использования кода
 - Приятный процесс разработки

Парадигмы программирования

- **Процедурное программирование:** разбиение программы на процедуры, использование ранее написанных процедур
- **Структурное программирование** как более развитая версия процедурного: тщательное проектирование структуры программы с целью создания полностью автономных в смысле создания и отладки процедур
- **Объектно-ориентированное программирование:** объединение переменных и методов, соответствующих определенным сущностям : человеку, машине, записной книжке, ...

Основы ООП: классы и объекты

Класс: **переменные + функции для работы с этими переменными.**

- Переменные класса — **поля (Data)**
- Функции класса для работы с полями — **методы (Methods)**
- Экземпляр класса — **объект.**

Пример

Объявление класса

```
class Human:  
Data:  
    Name; Age; weight; isMarried;  
Methods:  
    SetName(); setAge();  
    Setweight();  
    GetName(); GetAge();  
    GetMaritalstatus();
```

Использование класса

```
Human Pavel;  
Pavel.SetAge(21);  
Pavel.Setweight(70);  
Pavel.SetMaritalStatus(True);  
  
Human Natalia;  
Natalia.SetAge(22);  
Natalia.Setweight(50);  
Natalia.SetMaritalStatus(False);  
...
```

Основные принципы ООП

- Абстракция
- Инкапсуляция
- Наследование
- Полиморфизм

Абстракция

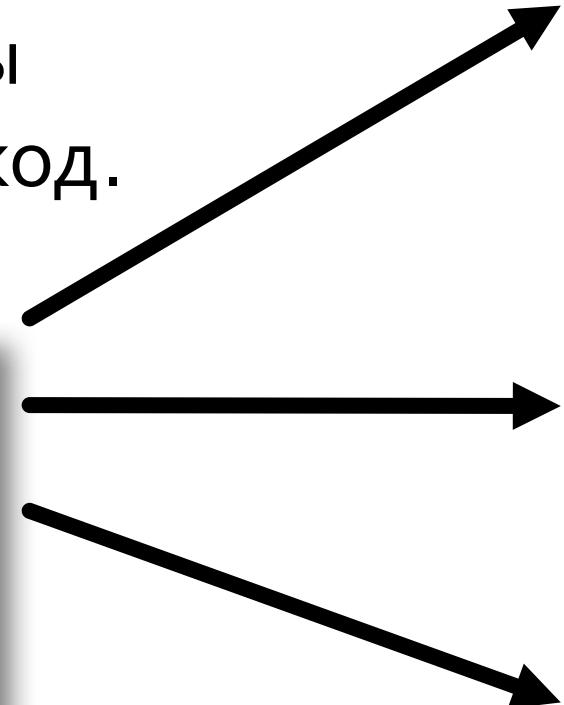
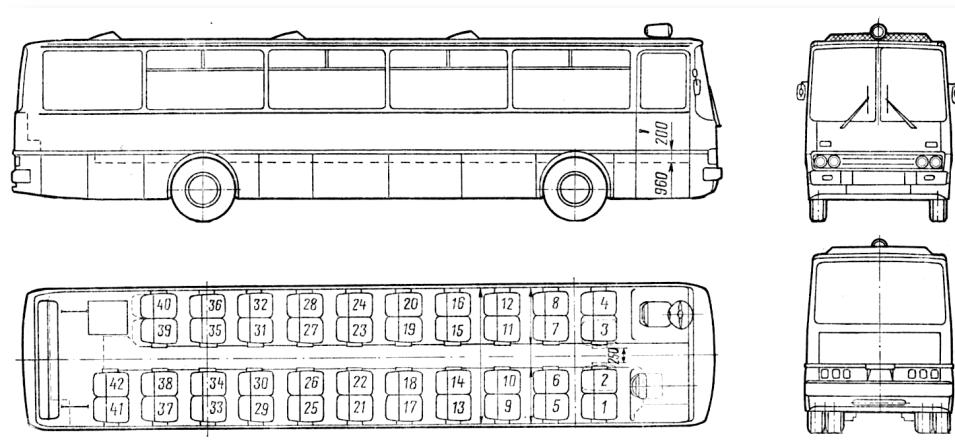
- Позволяет определить, какие поля и методы должен иметь класс, а какими из них можно пренебречь в контексте своего использования в конкретной программе
- **Пример:** человек как
 - работник в организации
 - студент в университете
 - гражданин в государстве.

Инкапсуляция

- Принцип сокрытия данных от внешнего воздействия.
Обеспечивает безопасность пользования программой
сторонними пользователями
- Разделение прав доступа к полям класса на **private** и
public.

Наследование

- Позволяет избежать повторения кода при расширении функциональности программы используя ранее написанный код.



Полиморфизм

- Способность объектов с одним названием иметь различную реализацию
- Примеры:
 - C++: *int add(int a, int b)* и *double add(double a, double b)*
 - Функция *MakeTimestep(AbstractSolver solver, Equation eq)* в решателе ОДУ: в качестве первого аргумента можно передать объект-наследник от класса *AbstractSolver*, например *ExplicitEulerSolver*, *RungeKuttaSolver*, *AdamsSolver*, ...

ИТОГ

- Познакомились с ООП
- Основные принципы ООП: **абстракция, инкапсуляция, наследование и полиморфизм**
- В курсе будет использоваться язык



Темы, которые будут рассмотрены

- Введение в синтаксис Python
- Тренировочные задачи по ООП
- ООП реализации стандартных задач вычислительной математики: **численное дифференцирование, интегрирование, решение ОДУ и УрЧП; реализация функциональных пространств**
- Обзор структуры крупных программных комплексов для решения вычислительных задач
- Использование средств ускорения Python.

Спасибо за внимание.

Спасибо за внимание.