ООП

# Лекция 1

Проблема? Проблема.

## Введение.

Генерал: «Здесь шлагбаум, или майора толкового поставить?»

Структурное программирование появилось в 61-м году, идеи ООП зародились в 66-м году. В рамках курса будет изучаться язык «C++», как расширение языка «C». Будет проведено знакомство с шаблонами (паттернами) проектировать, и способ строительства объектной модели.

Лабораторных работ будет 5.

Умер священник. Несут его ангелы к вратам рая, и вдруг вперёд его проносят водителя автобуса. Священник спрашивает: «А почему так?» Ангелы ему и говорят: «Ну так у вас на проповедях люди спали, а у водителя в салоне – молились».

Помимо ООП, существуют так же:

* Процедурный подход
* Функциональный подход
* Декларативный подход

## Структурное программирование

Технология (по Хоару) – это систематическое использование абстракций для управления массы деталей и способы проектирования, которые помогают проектировать программу.

50-й год. Возникают высокоуровневые ЯП. Возникает потребность составлять большие программы. Основной проблемой являлось сложность планирования сроков, надёжности и повторного использования и поддержки кода. IBM в 61-м году разработала технологию, позволяющую на порядки повысить производительность труда программиста и создавать программы без ошибок (но это, конечно, не правда. Ну почти, да, американцы рекламируют так свой продукт). Технология эта сейчас называется технологией структурного программирования.

Основные идеи структурного программирования:

1. Нисходящая разработка
2. Использование базовых компонентов
3. Сквозной структурный контроль

Этапы разработки проекта:

Анализ (на выхлопе должно быть ТЗ). После чего, циклически по каждому структурному элементу:

1. Проектирование
2. Кодирование
3. Тестирование

# Лекция 2

Раздвинь ножки…

Надо готовиться к праздникам. Особенно женщинам.

Приёмы сквозного структурного контроля, предложенные IBM:

1. Взаимодействие программистов между собой
2. Расположение руководителя в одном ряду с подчинёнными
3. Аналитиков и проектировщиков выращивать в своём коллективе
4. Организация контрольных сессий. Не приглашается руководитель. Приглашается представитель заказчика.

Есть люди, в каждой бочке затычка. Они всегда добавят свою бочонок дёгтя в большую ложку мёда.

Первая лабораторная работа будет посвящена технологии структурного программирования.

Сильные и слабые стороны структурного программирования

|  |  |
| --- | --- |
| Плюсы | Минусы |
| Логические ошибки исправляются на ранних стадиях разработки ПО | Тяжело модифицировать код. Очень дорого поддерживать код |
| Объединение этапов проектирования, кодирования и тестирования |  |
| Нет необходимости писать отладочные программы для каждой функции и для каждого модуля |  |
| Взаимодействие с заказчиком с ранних этапов разработки |  |
| Контроль продвижения проекта. Множество естественных контрольных точек |  |
| Код прозрачный, удобно выявлять ошибки |  |
| Крайне удобно распределять задание между программистами |  |
| Отказ проекта сводится к нулю |  |
| На низком уровне выделяются простые функции. Очень легко выделяются библиотеки |  |
| Нет работы «в холостую». Практически нет кода «в корзину». |  |
| Плавное распределение ресурсов. Основная нагрузка приходится примерно на середину проекта. Нет понятия «аврал» |  |
| Использование базовых логических структур. Стандартизация. Проще писать и читать код |  |

Вот сколько плюсиков. Или крестиков? Мы договорились, да? Что крестики у нас на кладбище.

Любое изменение написанного кода – понижение его надёжности

Городничему надо выдавать дочь замуж. Решил собрать всю знать города, и пригласил дьячка. Накрывается стол, гостей ещё нет, а дьячок уже припёрся. Накрыли на стол, поставили миску с чёрной икрой. Дьячок взял, и начал есть. Городничий у ужасе. Подходит к дьячку, и говорит:  
- Отец дьякон, это же чёрная икра!  
- Икра, икра  
- Отец дьякон, она же стоит рубль!  
- Стоит рубль, да  
- Отец дьякон, её же вредно сразу много есть!  
- А я буду есть её не сразу, а постепенно

Так называемый «код BS». Bowl of Spaghetti

У немцев есть такая поговорка: «Любая машина, в конце концов, становится Opel’ем»

В 66-м году Хоар на летней школы НАТО выступил с лекцией «Совместное использование записей». Он предложил 3 идеи:

1. Инкапсуляция. Выделение действий, которые можно проводить с данными, сразу. Таким образом, данные могут меняться, а действия - нет.
2. Наследование. Выполнение надстройки над объединением, которая будет заменять действия над данными, добавлять новые действия, и добавлять новые данные.
3. «Фундаментальная особенность нашего понимания мира заключается в том, что мы систематизируем наш жизненный опыт, представляя его в виде отдельных объектов…» - Хоар, 1966-й год. А почему бы в программу не перенести то, как объекты взаимодействуют в физическом мире?

Хоар выделил два вида взаимодействия объектов:

* Синхронное (или аксессорное). Аксессор – метод, позволяющий получать и изменять информацию. Get и set
* Асинхронное (или событийное взаимодействие)

Советских времён анекдот.  
Грузинский мальчик. Папа его отправил учиться в Москву, в институт. Купил ему «Волгу». Сын пишет в письме: «Учусь хорошо, нормально, только вот все ездят на троллейбусе, а я вот на «Волге». Отец ему в ответном письме: «Не волнуйся, сынок. Будешь хорошо учиться, сдашь сессию – и я куплю тебе троллейбус»

У меня был дипломник, который писал диплом. Он мне тогда сказал: «Я начал писать диплом, и наконец-то понял, что вы нам рассказывали на первом курсе»

Задержка применения ООП задержалась, так как событийный подход должен поддерживаться и на уровне ОС, и на уровне аппаратного обеспечения.

Революционные идеи? Революционные идеи.

Про студента ИУ7.  
Жене сказал, что пошёл к любовнице. Любовнице, что с женой, а сам кодить, кодить и кодить

- Доктор, я умру?  
- А куда вы денетесь?

## Основные понятия в ООП

Объект – конкретная реализация абстрактного понятия, обладающая характеристиками состояния, поведения и индивидуальности. С каждым состоянием так же будет связано действие, задачей которого будет являться перевод действия в это состояние.

Выделяют категории объектов:

* Реальные объекты. Абстракции фактического существования каких-то предметов в реальном мире.
* Роли. Абстракции цели или назначения человека, оборудования, организации, и так далее.
* Инциденты.
* Объекты взаимодействия. Получаются в результате отношений между другими объектами.
* Объекты спецификации.

Мур – это фамилия, а не Московский Уголовный Розыск

Как мы это будем реализовывать? Ну мы посмотрим

Выделяют два вида отношений между объектами:

* Отношение старшинства или использования. Любой объект выступает в трёх видах:
  + Воздействия. Могут воздействовать на другие, но не могут быть подвержены воздействию. Активные
  + Исполнители. Подвержены воздействию, но никогда не оказывают воздействие сами. Пассивные
  + Посредники. Могут подвергаться воздействию, как и оказывать его.
* Отношение включения. Объект может включать в себя другие объекты.

# Лекция 3

Знаете что такое синдром зеркала? Это когда своё мужское достоинство можно увидеть только в зеркало. Есть ещё синдром арбуза. Арбуз растет – кончик сохнет.

## Класс

Класс – это такая абстракция множества предметов, что все предметы в этом множестве имеют одни и те же характеристики, и у которых одно и то же поведение.

Отношения между классами:

* Наследование
* Использования (агрегация)
* Включения (композиция)
* Доменное

Понятно, да?

Домен – это отдельный реальный, гипотетический или абстрактный мир, населённый определённым числом объектов, ведущих себя в рамках определённых правил.

Недостатки ООП, в сравнении с структурным подходом

* Обязателен этап проектирования, на котором будет построена объектная модель
* Высокий порог вхождения
* Количество кода увеличивается. Время написания такого кода тоже увеличивается. Объём необходимой памяти увеличивается. Резко увеличивается время выполнения программы
* Сложно контролировать ресурсы (во всех современных языках эта проблема решена)
* Мёртвый код

Непонятно. Понятно, да?

Все эти недостатки перекрываются большим плюсом: в ООП можно легко модифицировать программы.

Со временем пришло понимание, что это не просто плохо, а очень плохо

## Лабораторная работа №1

Вьювер каркасной модели.

# Лекция 4

## C++

Для языка C++ было очень важно сохранять обратную совместимость с C. Помимо struct и union, была добавлена структура данных class.

Все женщины делятся на «дам», «не дам», «дам, но не вам»

Только 20% мужчин любят толстых женщин. Остальные только очень толстых

Базовый класс ещё называют «папой». Он ничего не знает о своих производных

# Лекция 5

Любой объект надо инициализировать

Сделал дело – прикрой веткой

Создание конструктора переноса ClassName(ClassName &&a);

Создание конструктора по умолчанию ClassName() = default;

Удаление конструктора: ClassName() = delete;

Explicit – модификатор, запрещающий неявный вызов конструктора

Раньше в бауманской столовой был комплексный обед. Состоящий из несовместимых компонентов

# Лекция 6

С программистами на Java будем бороться. Это ненормальное программирование

База может быть прямой (непосредственное наследование), или косвенной (наследование опосредовано промежуточным классом)

Если вы знакомитесь с девушкой, вас кто интересует? Девушка, правильно? Нет, ну есть такие люди, которые хотят сразу познакомиться с родителями. Как говорится, яблоко от яблони недалеко падает. Или наоборот?

Это только индусы дважды пишут в регистр. Чтоб записалось так записалось

Вы познакомились с девушкой, пригласили её в ресторан: что вы ей скажете? Среди всех вариантов ответа, мне понравился следующий: «мне нужно отойти ненадолго, встретиться со своим лучшим другом, с которым я вас познакомлю попозже»