

ПРАКТИЧЕСКАЯ РАБОТА №1

Изучение структур и разработка структур, членов-структуры и статических методов структуры на динамической памяти.

Цель работы: изучить новые способы представления данных с помощью создания переменных типа структур. Познакомиться с созданием базовых методов структур позволяющих реализовывать хранение структуры в динамической памяти компьютера. Изучить принципы методов: конструктор, “копирующий” конструктор, конструктор “по умолчанию”, деструктор.

Задание: реализовать структуру (*struct*) в соответствии с вариантом. Структура должна обеспечивать набор методов для работы с данными размещая объекты в динамической памяти компьютера. Структура должна быть описана с использованием *typedef*. Создать методы: конструктор, конструктор “копирования”, конструктор “по умолчанию”. Реализовать указанные методы с динамическим выделением памяти для хранения перечисленных в соответствии с вариантом полей. Внимательно рассмотреть цикл работы с динамической памяти и обратить внимание на вложенность. Создать метод деструктор для освобождения памяти. Посмотреть, как вызываются конструкторы и деструкторы. Реализовать структуру отдельным модулем (*struct.h*, *struct.c*). Где *struct* название структуры.

Собрать демонстрационную программу. Для сборки демонстрационной программы использовать *makefile*.

В результате написать подробный отчет, описывающий выполнение описанных задач, демонстрируя вставками изображений самые важные моменты.

Вариант 1. Точка в трехмерном пространстве (*point3d*):

Задание на 3.

Создайте структуру *point3d*, представляющий точку в трехмерном пространстве. Данные точки могут быть представлены как целыми, так и вещественными значениями не более чем 32bit. Реализуйте методы для:

- Сравнения двух точек ($==$, $!=$, $>$, $<$, $>=$, $<=$)
- Ввода/Вывода координат
- Изменение данных структуры ($++$, $--$, *setter's*).

Задание на 4.

- Вычисления расстояния между двумя точками.
- Смещения точки на заданный вектор.
- Генерация точки в случайном месте

Задание на 5.

- Поворот точки в пространстве Эйлера

Вариант 2. Двумерная матрица (*matrix2d*):

Задание на 3.

Создайте структуру *matrix2d*, представляющую двумерную матрицу. Значения матрицы могут быть представлены как целыми, так и вещественными значениями не

более чем 32bit. Матрицы могут быть с главной диагональю и без нее. Реализуйте методы для:

- Сравнения двух матриц (`==`, `!=`, `>`, `<`, `.>=`, `<=`)
- Ввода/Вывода матриц
- Изменение данных структуры (`++`, `--`, `setter's`).
- Заполнение матрицы случайными значениями

Задание на 4.

- Получение столбца/строки
- Транспонирование матрицы
- Нахождение определителя (детерминант)

Задание на 5.

- Вычисление обратной матрицы

Вариант 3. Дата и Время (*datetime*)

Задание на 3.

Создайте структуру *datetime*, представляющую день, месяц, год, часы, минуты. Реализуйте методы для:

- Сравнения двух объектов структур *datetime* (`==`, `!=`, `>`, `<`, `.>=`, `<=`)
- Ввода/Вывода даты и/или отдельных полей структуры.
- Изменение данных структуры (`++`, `--`, `setter's`).

Задание на 4.

- Форматированный строковый ввод/вывод объектов структур *datetime*.
- Нахождение временных интервалов между двумя объектами структур *datetime* (в годах, месяцах, днях, часах, минутах).

Задание на 5.

- Представление объекта структуры *datetime* как количества секунд с начала эпохи Unix.

Вариант 4. Банковский счёт (*aCash*):

Задание на 3.

Создать структуру *aCash* для моделирования работы банковского счёта. У *aCash* поля: идентификатор, номер расчётного счёта, типа карты (дебетовая (пополнять и снимать может только клиент), кредитная (пополнять и снимать может только клиент, разрешен отрицательный баланс, но при этом есть фиксированный размер отрицательного значения), эскроу(пополнять и снимать может только клиент, все могут просматривать баланс, не только клиент)), баланс счёта, валюта, дата создания (день, месяц, год) и дата закрытия (день, месяц, год; p.s. Дата закрытия это +6 лет от даты открытия).

Реализуйте методы для:

- Ввода/Вывода информации по счёту и/или отдельных полей структуры.
- Сравнения двух объектов структур *aCash* (`==`, `!=`, `>`, `<`, `.>=`, `<=`) (по датам, открытию, закрытию, балансу, типу карты)

- Реализовать операции пополнения/внесения и снятия/оплаты.

Задание на 4.

- Строковое представление объекта
- Расчёт списаний и накоплений счёта: фиксированной стоимости в валюте счёта и в процентном отношении.
- Реализовать операции перевода денежных средств с одного счёта на другой, со взятием фиксированной комиссии при переводе денег со счёта с одной валютой, на счёт с другой валютой.

Задание на 5.

- Реализовать проверку, закрыт счёт или нет, в указанной временной точке.
- Создайте структуру для хранения информации о бонусах клиента, которая содержит текущее количество бонусов, процент начисления бонусов за каждую операцию.
- Реализуйте отдельные методы для начисления бонусов за определённые операции (например, пополнение баланса).
- Обеспечьте возможность использования бонусов для частичного погашения операций.