3. лабораторная работа

Композиция (агрегация). Статика. Обработка исключений.

Спецификации варианта, создать новый класс с массивом Private из объектов подкласса, реализованного в второй лабораторной работе (композиция). В классе предусмотреть:

1. Использование статических (Static) атрибутов и методов работы с размером массива “по умолчанию”.
2. Метод добавления сведений Add (<objekts>).
3. Метод вывода информации в Print ().
4. Для уничтожения деструкторного массива (для освобождения памяти).
5. Контроль длины массива при помощи исключительной обработки (exception handling). В функции main () должны быть контролируемые блоки try и обработчики исключений catch. Исключение предлагается в методе нового класса с помощью throw.
6. Метод из списка вариантов.

Список вариантов:

1. Класс фирм (фирма). В основе - массив Employee. Найти максимальную зарплату в фирме (метод Double GetMaxSalary ()).
2. Класс State (страна). В основном - массив городских City. Найти минимальное количество населения в городе (метод Long GetMinInhabitants ()).
3. Класс CarPark (автопарк). В основном - массив “Car”. Найти максимальную мощность двигателя (метод Int GetMaxPower ()).
4. Класс BookShop (книжный). В основном - массив “Book”. Найти максимальную цену книги (метод float GetMaxPrice ()).
5. Класс RailBridgesList (список железных мостов). В основном - массив железных дорог RailwayBridge. Найти максимальную высоту моста (метод Int GetMaxHeight ()).
6. Класс Zoo (зоопарк). В основном - массив животных ZooAnimal. Найти минимальный год рождения (получения) (метод short GetMinBirth ()).
7. Класс ComputerNetwork (компьютерная сеть). В основном - массив NetComputer. Найти максимальный объем оперативной памяти (метод short GetMaxRAM ()).
8. Класс ScreenWindows (окна экрана). В основном - массив TitledWindow. Найти максимальную площадь окна (метод Int GetMaxWinArea ()).

4. лабораторная работа

Шаблон и параметрический полиморфизм

Описать все классы (кроме класса OverflowException) из первой, второй и третьей лабораторной работы в качестве шаблонов (template) с одним параметром.

template **<** class **T >**

Создать два класса из третьей лаборатории на основе шаблона. Создание объектов обоих классов, применение некоторых объектов объектов и выведение сведений о состоянии каждого объекта.

Список вариантов:

1. Суперкласс: Хуман (человек). Подкласс: Employee (сотрудник). Параметр: тип пола. Значения: char, Unsigned char.
2. Суперкласс: settlement (населенный пункт). Подкласс: City (город). Параметр: тип кода территории. Значения: Long, Unsigned Long.
3. Суперкласс: Vehicle (транспортное средство). Подкласс: Car (автомобиль). Параметр: тип производственного года. Значения: Unsigned short, Unsigned Int.
4. Суперкласс: Article (товар). Подкласс: Book (книга). Параметр: цены на тип участков. Значения: флот, Double.
5. Суперкласс: Bridge (мост). Подкласс: RailwayBridge (железнодорожный мост). Параметр: тип грузоподъемности. Значения: Unsigned Int, Unsigned Long.
6. Суперкласс: Animal (животное). Подкласс: ZooAnimal (животное в зоопарке). Параметр: тип года рождения (получения). Значения: Unsigned short, Unsigned Int.
7. Суперкласс: computer (компьютер). Подкласс: NetComputer (сетевой компьютер). Параметр: тип оперативной памяти. Значения: Unsigned short, Unsigned Int.
8. Суперкласс: Window (окно на экране компьютера). Подкласс: TitledWindow (окно с заголовком). Параметр: тип координат местонахождения (x, y). Значения: short, Unsigned short.

5. лабораторная работа

Основы языка Java

Исполнять программу с 1 ¬ й лабораторной работы на рабочем языке. Можно использовать любую среду разработки: Eclipse, NetBeans, IntelliJIDEA и другие.

На языке Java:

1. Методы реализуются внутри класса (реализация не отделена от интерфейса).
2. Нет секций Private, public, protected: модификатор доступа независимо указывается по каждому атрибуту и каждому методу.
3. Нет деструктора.
4. Нет инициализаторов.
5. Методам не является пост ¬ фикса const.
6. Нет внедренных методов (Inline).
7. Атрибуты - константы описываются как Final (а не const).
8. Строки символов являются объектами класса String.
9. Информация об объекте обычно выводится с помощью метода toString ().

Примечания:

1. Самостоятельно написать конструктор копии (конструкторский параметр является объектом того же класса). В программе создать объект-копию:

Coordpoint cpCopy = New CoordPoint (cp);

1. Вызвать конструктор с тремя параметрами из остальных двух конструкторов, в которых должна быть только одна строка кода. Применить в программе все три конструктора при создании трех объектов.
2. Подсчет созданных объектов. Для таких целей следует написать статический метод getObjectsCount (), который возвращает частную статическую переменную objectsCounter. Эта переменная увеличивается только в одном конструкторе. Вызвать этот метод из главной программы класса.

6. лабораторная работа

Друзья. Перезарядка операторов.

Основой работы является работа лаборатории 3, индивидуальные варианты задач не меняются. Изменить код своей объективной программы по аналогии с изменениями в тесте. Конечный результат: оператор < < (3 экземпляра) и + = (1 экземпляр) перезарядка.

1. Во всех классах замещать метод Print () с перезаряженным оператором вывода < <, являющийся другом класса. Речь идёт о суперклассе с 1 ¬ й лабораторной работы (на тестовом примере - CoordPoint), подклассе от работы 2 ¬ й лаборатории (на тестовом примере - DisplayPoint) и классическо ¬ композиции из работы лаборатории 3 (на испытательном примере - DisplayBrokenLine).
2. В классической композиции (в тестовом примере - DisplayBrokenLine) замещение метода AddNode (...) с перезаряженным оператором присоединения + =, являющимся членом класса. В завещании это было бы (но нет):

void = = (const DisplayPoint &);

1. Применить обе операторы к главной программе. Возможный код:

\* DL + = \* D1;

\* DL + = D2; cout < < \* DL;