Методические указания к выполнению домашнего задания

ДЗ 3 "Объектно-ориентированное программирование, библиотеки классов, исключения, потоковая многозадачность"

Студенты, номера которых в списке журнала больше 21, выбирают номер варианта по формуле N % 21, где N – номер студента в списке журнала.

Первым шагом в выполнении ДЗ должно быть **осмысление** поставленной задачи, и, в частности, выделение новых типов объектов, для которых нужно разработать обобщенные описания объектов в виде классов на языке Си++. Нужно определить свойства объектов каждого класса и его поведение. Необходимо первоначально выделить основные данные для объектов класса и перечень методов/функций, которые эти данные изменяют. На первом этапе это делается на абстрактном уровне.

Под предметной областью понимается совокупность понятий, объектов реального мира, их свойств, а также что с этими объектами можно делать в программе и делается в жизни. Для одной и той же системы классов может быть выделено много различных предметных областей, которые, в конечном счете, зависят от решаемой (поставленной) задачи автоматизации. Поэтому первым шагом нужно задать одну или несколько предметных областей, для которых будет разрабатываться система классов.

Пример. Рассмотрим пример разработки системы классов улиц и классов домов, которые могут располагаться на конкретных улицах. Возможны следующие задачи, для которых разрабатывается такая система классов: контролирование оплаты жителями коммунальных услуг, проведение выборов, ремонт строений на улице и самой улицы, ведение паспортного учета жителей (или прописки), оценка числа жителей на улице для планирования социальных услуг, построение карт города с улицами (детализация по домам), оценка движения автомобилей по улице и т.д. Из этого перечня задач видно, что от выбора конкретной задачи зависят свойства будущих объектов и набор операций (методов), которые над этими объектами предусматриваются. Определим, для примера, предметную область так: учет числа жителей на улице и оценка необходимости ремонта строений улицы.

Далее для каждого объекта предметной области, на содержательном уровне, нужно более детально задать перечень свойств (данных) и возможных операций над этими свойствами.

Пример. Например, свойства для дома: номер дома, этажность дома, число жителей в доме, признак ремонта, тип строения и т.д. Операции для дома: установка числа жителей, изменение типа дома, установка или сброс признака необходимости ремонта дома. Для улицы можно выделить свойства: название улицы, тип улицы, число домов на улице, число жителей, признаки ремонта улицы и т.д.

Далее целесообразно проанализировать предметную область и поставленную задачу и определить необходимость дополнительных классов для реализации. Эти классы могут быть технологическими (вспомогательными) и содержательными. Технологические классы могут быть выделены, например, для реализации контейнеров элементных объектов (списки, массивы, множества и т.д.) и для выполнения других функций решения задачи. Эти классы не имеют явных аналогов в предметной области и выделяются на основе опыта программиста. Дополнительные содержательные классы могут потребоваться для комплексного решения задачи. Например, Для нашего случая это могут быть: класс городов (при построении карт), класс планов ремонта улиц (в ремонтной задаче), классы бюджетов оплаты жилья (для контроля оплаты квартир) и т.д.

Более важным является определение перечня возможных операций над объектами. Не всегда удается сразу определить его полностью: он наращивается и видоизменяется в процессе проектирования и реализации программной системы.

Пример. Предположим тема задания следующая - разработать систему классов для описания объектов строений/домов улиц (элементные объекты) и объектов типа улица, содержащих в себе элементные объекты (дома). Улица - это контейнерный объект. Задачи, для которых разрабатывается система классов, заданы выше. Для домов целесообразно выделить свойства создания дома (построение), удаления дома (снос дома), изменения свойств дома (установка признаков необходимости ремонта, числа жителей и т.д.). Для улиц возможные операции: добавление и удаление домов, переименование улицы, поверка признака ремонта улицы и домов на улице. Предположим, что нужно предусмотреть операции для контейнерного объекта: добавления строений, переименования улиц, сноса строений, печати списка домов, добавления строений и т.д. Более детально это расшифровано ниже, включая и раскрытие понятий предметной области:

1. На первом этапе необходимо выделить следующие понятия: дома/строения и улицы, как упорядоченной совокупности домов.
2. Понятие дома - объекта, возможно, будет обладать следующими свойствами: номер дома, этажность дома, число жителей дома, расположение на левой или правой стороне улицы, название дома, характеристика строения и т.д.
3. Понятие улицы – объекта, возможно, будет содержать следующие свойства: название улицы, перечень домов улицы, число проживающих на улице, число домов, район расположения, город улицы и т.д.
4. Поведение дома – набор методов, возможно, будет содержать следующие действия: создание дома, разрушение дома, чтение параметров, изменение номера дома и название, установка число проживающих, изменение этажности – перестройка дома, распечатка дома и т.д.
5. Поведение улицы – набор методов, возможно, будет содержать следующие действия: создание улицы, удаление улицы, добавление дома на улицу, удаление дома с улицы, новая нумерация домов на улице, распечатка домов улицы, изменение названия улицы, чтение параметров улицы, объединение двух улиц, деление улиц на две и т.д.
6. Уже сейчас, можно придумать название для классов этих объектов: дом **Home**, а улица – **Street**. Так как это описание классов на языке программирования, то мы должны давать названия латиницей. Лучше эти названия дать осмысленно.
7. В качестве технологических классов можно выделить классы: список (**List**).

Пример. Материальный объект – дом/строение. Контейнерный объект улица, как совокупность домов. На первом этапе необходимо на понятийном уровне продумать смысл каждого свойства и каждого действия над выделенными типами объектов. Ниже приведены некоторые свойства классов с описание содержания:

1. Номер дома порядковый – целочисленная переменная (int **HomeNumber**), а операции по изменению номера дома и чтению этих параметров (**SetHomeNumber** и **GetHomeNumber**) и т.д. Придуманные обозначения желательно поместить в таблицу, в которой будут перечислены все понятия их свойства и методы работы с данными объекта, их названия и типы.
2. Номер дома символьный – символьная переменная (char ***Home_Number**), потребуется, так как дома часто имеют различные индексы в номере (Например: “Дом 5а-стр.2”).

3. Название улицы – символьный массив (char **NameStreet**[30]) или указатель на имя (char * **pNameStreet**). Для добавления дома на улицу придумаем метод (**AddHome**), а для удаления дома с улицы (**DeleteHome**). И т.д. Придуманные обозначения желательно поместить в таблицу, в которой будут перечислены понятия/методы, их названия и типы.

Таблица с описанием классов домов может выглядеть примерно так (расширенный вариант):

Класс домов - Home

Смысловое описание класса: класс домов используется для создания объектов типа дом для хранения информации о конкретном доме: номер, число жителей, число квартир, тип дома, требования к ремонту, число этажей.

Базовые классы класса **Home: AbstrHome**

Свойства класса **Home**

№ п/п	Содержание свойства	Тип данных и название	Примечание
1.	Символьный номер дома	char ***Home_Number**;	
2.	Номер дома числовой	int **iHome**;	
3.	Число этажей в доме	int **EtagCount**;	
4.	Число жителей в доме	int **MenCount** ;	
5.	Тип дома	HomeType **TypeHome** ;	
6.	Требуется ли ремонт дома	BOOL **HomeRemont** ;	
7.	Число квартир в доме	int **NumbApartament**;	

Методы класса **Home**

№ п/п	Содержание метода	Тип метода	Прототип метода	Примечание
1.	Создание дома без параметров	конструктор (конст.)	**Home**();	Построение пустого дома
2.	Создание дома по типу другого	конст.	**Home**(Home & H) ;	Построение
3.	Создание дома по типу другого	конст.	**Home**(Home * pH);	Построение
4.	Создание дома с симв. номером	конст.	**Home**(const char *HomName, const char *Number);	Построение
5.	Создание дома с симв. И числовым-номерами	конст.	**Home**(const char *HomName, const char *Number, int Numb) ;	Построение
6.	Создание дома со всеми параметрами	конст.	**Home**(const char *HomName, const char *Number, int Numb, int Etag, int Men=0,HomeType Type = fast,int Apart=0)	Построение
7.	Удаление дома	Деструктор	~**Home**()	Снос дома

8.	Изменить имя дома	Метод	void **setName**(const char *HomName , const char *Number=NULL);	
9.	Получить имя дома	Метод	const char ***getName**();	
10.	Получить номер дома	Метод	const char ***getNumb**();	
11.	Получить числовой номер дома	Метод	int **getNo**();	
12.	Получить параметры дома	Метод	void **getParam**(int & iH, int & Etag ,int & Men ,HomeType & Type, int & Apart);	
13.	Установить параметры дома	Метод	void **setParam**(int iH, int Etag ,int Men ,HomeType Type, int Apart);	
14.	Установить все параметры дома	Метод	void **setAllParam**(const char *HomName , const char *Number, int iH, int Etag , int Men ,HomeType Type, int Apart , BOOL rem = false);	
15.	Оператор присваивания	Оператор =	Home **operator** =(Home & H);	Оператор должен быть перегружен, так как в объекте есть данные из динамической памяти
16.	Оператор вставки в поток вывода	Оператор <<	ostream &**operator** << (ostream & out);	
Виртуальные методы класса **Home**				
17.	Получить тип класса	Виртуальный метод	virtual int **classType**();	
18.	Получить имя класса	Виртуальный метод	virtual char ***className**();	

Для улиц можем получить следующую таблицу свойств и методов (расширенный вариант):

Класс улиц - Street

Смысловое описание класса улиц: класс улиц используется для создания объектов типа улица, включающая в себя дома. Дома на улице упорядочены. Учитываются свойства: название улицы, признаки ремонта, тип улицы, число домов на улице, список соседних улиц.

Базовые классы класса **Street: List**

Свойства класса **Street**

№ п/п	Содержание свойства	Тип данных и название	Примечание
1.	Символьное название улицы	char *Name_Street	
2.	Число домов на улице	int Homes_num	
3.	Номер улицы	int NumberStreet	
4.	Признак ремонта домов на улице	BOOL Remont	
5.	Признак ремонта улицы	BOOL RemontStreet	
6.	Тип улицы	StreetType StrType	
7.	Список соседних улиц	Street * ListOfNear	Зарезервировано для развития
8.	Список домов на улице	Home * pList;	

Методы класса **Street**

№ п/п	Содержание метода	Тип метода	Прототип метода	Примечание
1.	Создать улицу без параметров	конструктор (конст.)	**Street**()	Проложить улицу
2.	Создать улицу с названием	конст.	**Street**(const char *sName)	Проложить улицу
3.	Создать улицу с названием и ключом для поиска	конст.	**Street**(const char *sNumbSearch, const char *sName)	Проложить улицу
4.	Создать улицу с номером	конст.	**Street**(int Num)	Проложить улицу
5.	Создать улицу с названием и номером	конст.	**Street**(const char *sName , int Num)	Проложить улицу
6.	Создать улицу на основе другой	конст	**Street**(Street & S)	Проложить улицу
7.	Удалить улицу	Деструктор	~**Street**()	Снести улицу
8.	Добавить дом на улицу	Метод	void **add**(Home *pH, TypeAddDel = tail , int Numb = 1 , TypeAddDel = createObj)	
9.	Удалить дом с улицы	Метод	void **del**(Home *pH , TypeAddDel = tail , int Numb = 1 , TypeAddDel = nodelete-	

			Obj)	
10.	Получить число домов	Метод	int **GetNumberHome**()	
11.	Получить число жителей	Метод	int **GetNumberMens**()	
12.	Получить число квартир	Метод	int **GetNumberApart**()	
13.	Получить название улицы	Метод	char * **GetNameStreet**()	
14.	Получить имя улицы для поиска	Метод	char * **GetKeyNameStreet**()	
15.	Получить номер улицы	Метод	int **GetNumbStreet**()	
16.	Получить номер улицы для поиска	Метод	int **GetKeyNumbStreet**()	
17.	Установить название улицы	Метод	void **SetNameStreet**(const char * NameStr)	
18.	Установить название ключ для поиска улицы	Метод	void **SetKeyNameStreet**(const char * sName)	
19.	Установить номер улицы	Метод	void **SetNumbStreet**(int n)	
20.	Установить номер ключ для поиска улицы	Метод	void **SetKeyNumbStreet**(int k)	
21.	Получить признак ремонта домов	Метод	BOOL **GetRemont**()	
22.	Получить признак ремонта улицы	Метод	BOOL **GetRemontStr**()	
23.	Установить признак ремонта улицы	Метод	void **SetRemontStr**(BOOL rS)	
24.	Получить тип улицы	Метод	StreetType **GetStreetType**()	
25.	Установить тип улицы	Метод	void **SetStreetType**(StreetType t)	
26.	Оператор присваивания	Оператор =	Street **operator** =(Street & S);	Оператор должен быть перегружен, так как в объекте есть данные из динамической памяти
27.	Добавить дом к улице	Оператор +	friend Street & **operator** +(Street & X , Home & Y);	Дружественная внешняя функция
28.	Оператор вставки в поток вывода	Оператор <<	ostream &**operator** << (ostream & out);	

Виртуальные методы класса **Street**				
29.	Получить тип класса	Виртуальный метод	virtual int **classType**();	
30.	Получить имя класса	Виртуальный метод	virtual char ***className**();	

Использование контейнерных классов

В каждом задании на ДЗ требуется использовать минимум один контейнерный класс. Контейнерные классы – это такие классы, на основе которых создаются объекты, включающие в себя другие объекты (отношение накопления). Контейнерные классы могут быть последовательными (вектор, список, дек и др.), упорядоченными (map, set и др.), неупорядоченными (unordered_set, unordered_map и др.) Для реализации контейнерных классов студенты могут выбрать один из подходов:

- Использование стандартных (библиотечных) классов.
- Создание своего класса.

Некоторые особенности выполнения ДЗ

При выполнении задания должны быть учтены следующие требования.

1. Для классов объектов **явно** определяются и разрабатываются: **конструкторы** (не менее двух для каждого класса) и **деструкторы** (для каждого класса);

Пример. Конструкторы могут с первоначальным заданием параметров или нет, например, с заданием названия улицы и т.д.

2. В каждом классе **помимо** конструкторов и деструкторов должны быть определены не менее **четырех** методов (функций членов класса) и не менее **четырех** свойств (данных класса);

Пример. Методы для улиц могут быть такими: добавление дома, удаление дома, распечатки домов улицы, нумерация домов, сложение улиц, деление улиц и т.д. Например, деление улицы на две может интерпретироваться так: первые N домов переносятся на "улицу Горького", а остальные на "Тверскую". Т.е., другими словами, из одного объекта мы получаем два других объекта с определенными свойствами. Разрабатывая такую операцию нужно продумать вопросы: где создаются новые объекты, как они заполняться, уничтожается ли исходный объект и т.д.

3. В одном классе по **выбору,** в зависимости от варианта, определяется как минимум **одна** перегружаемая операция (обязательная операция, обозначенная знаком "+").

Пример. Добавление дома к улице – **Street1+Home1;**.

4. Описания классов должны быть вынесены в отдельные файлы (*.h). При этом необходимо исключить повторное включение описаний в исходный текст.

Пример. Например, с помощью переменных этапа компиляции (define __STREET_H):

```
#if !defined( __STREET_H )
…… Основной текст файла
#endif  //
```

5. Обязательным является перегрузка операции вывода (<<) в стандартный поток cout для предметного класса.

ВАРИАНТЫ ДЗ

Тема задания	Контейнер		
	Список list	Неупорядоченное множество unordered_set	Упорядоченное множество set
	Номера вариантов		
Класс домов и класс улиц для учета поступлений квартплаты.	1	8	15
Класс студентов и класс учебных групп для учета успеваемости по итогам одного семестра, предусмотреть расчет среднего балла для группы по всем дисциплинам и по отдельной дисциплине, печать отличников и задолженников.	2	9	16
Класс сотрудников и класс структурных подразделений (отделов и др.) для отдела кадров организации.	3	10	17
Класс сотрудников и класс структурных подразделений (отделов и др.) для учета начислений зарплаты в бухгалтерии.	4	11	18
Класс книга и класс библиотека для учета книг в библиотеке.	5	12	18
Класс файлов и класс каталогов файлов, предусмотреть поиск по имени файла. Предусмотреть операции перемещения файлов, их добавления и удаления, поиска, переименования, сравнения и объединения каталогов и т.д.	6	13	20
Класс домов и класс улиц для получения списка избирателей для участия в выборах.	7	14	21

Задание

Разработать программу, работающую с объектами классов. Используя интерфейс командной строки, реализовать следующие режимы работы: «Ввод нового объекта и добавление его в контейнер», «Поиск объекта в контейнере по значениям полей с печатью данных о найденных объектах», «Редактирование объекта», «Удаление объекта из контейнера», «Сохранение данных всех объектов в файле», «Чтение данных объектов из файла», «Сортировка объектов контейнера по выбранному полю для list», «Печать списка объектов». Предусмотреть обработку исключений, возможные исключения определить самостоятельно. В функции main должен быть главный поток, который создает консольное меню для выбора режима, режимы, требующие взаимодействия с пользователем (ввод нового объекта, редактирование объекта, поиск и печатью, печать списка объектов и т.п.) выполняются в этом главном потоке. Режимы, не требующие взаимодействия с пользователем (удаление, сохранение в файле и чтение из файла), выполняются в отдельном потоке, созданном в главном, при этом обеспечить синхронизацию при доступе к данным объектов.