|  |  |
| --- | --- |
| Институт (факультет) | Институт информационных технологий |
| Кафедра | Кафедра математического и программного обеспечения ЭВМ |

**ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ОТНОШЕНИЯ ПРОСТОГО НАСЛЕДОВАНИЯ**

|  |  |
| --- | --- |
| **Дисциплина:** | ООП |
| **Темы:** | Наследование |

**Среда разработки:** Microsoft Visual Studio

**Язык программирования:** C++

**Тип проекта:** Консольное приложение

**Задание на лабораторную работу №6**

**ТРЕБОВАНИЯ К РАЗРАБОТКЕ**

1. Запрещается использовать обработку исключительных ситуаций и генерировать исключения.
2. Придерживайтесь принципа DRY (Don’t repeat yourself).
3. Крайне рекомендуется придерживаться принципа DRY (Don’t repeat yourself).
4. Обязательно наличие комментариев.
5. Разрешается использовать шаблонный класс-контейнер, либо следует заменить хранимый тип объекта на указатель на класс-интерфейс.
6. Допускается заменить хранимый тип объекта на обобщенный указатель (void\*).
7. Проверьте, что ваш код соответствует принципу из П.2

**ЦЕЛЬ РАБОТЫ**

Изучить способы создания производного класса и особенности работы с ним, правила инициализации и доступа к элементам производного класса; приобрести практические навыки наследования.

**ЗАДАНИЯ**

*8 часов*

1. Ознакомьтесь с УМП по ООП часть 2 раздел VIII.
2. Создайте базовый класс-интерфейс для АТД, реализованного по заданию ЛР1, состоящий чисто из виртуальных функций, используя одиночное наследование.
3. Разработайте иерархию родственных типов используя простое наследование, количество уровней 4-5 или (5+ классов).
4. В реализации функции вывода (кроме абстрактных классов), необходимо в самом начале обеспечить вывод информации о названии класса (название предметной области), например:

«**Монитор**

Производитель: …

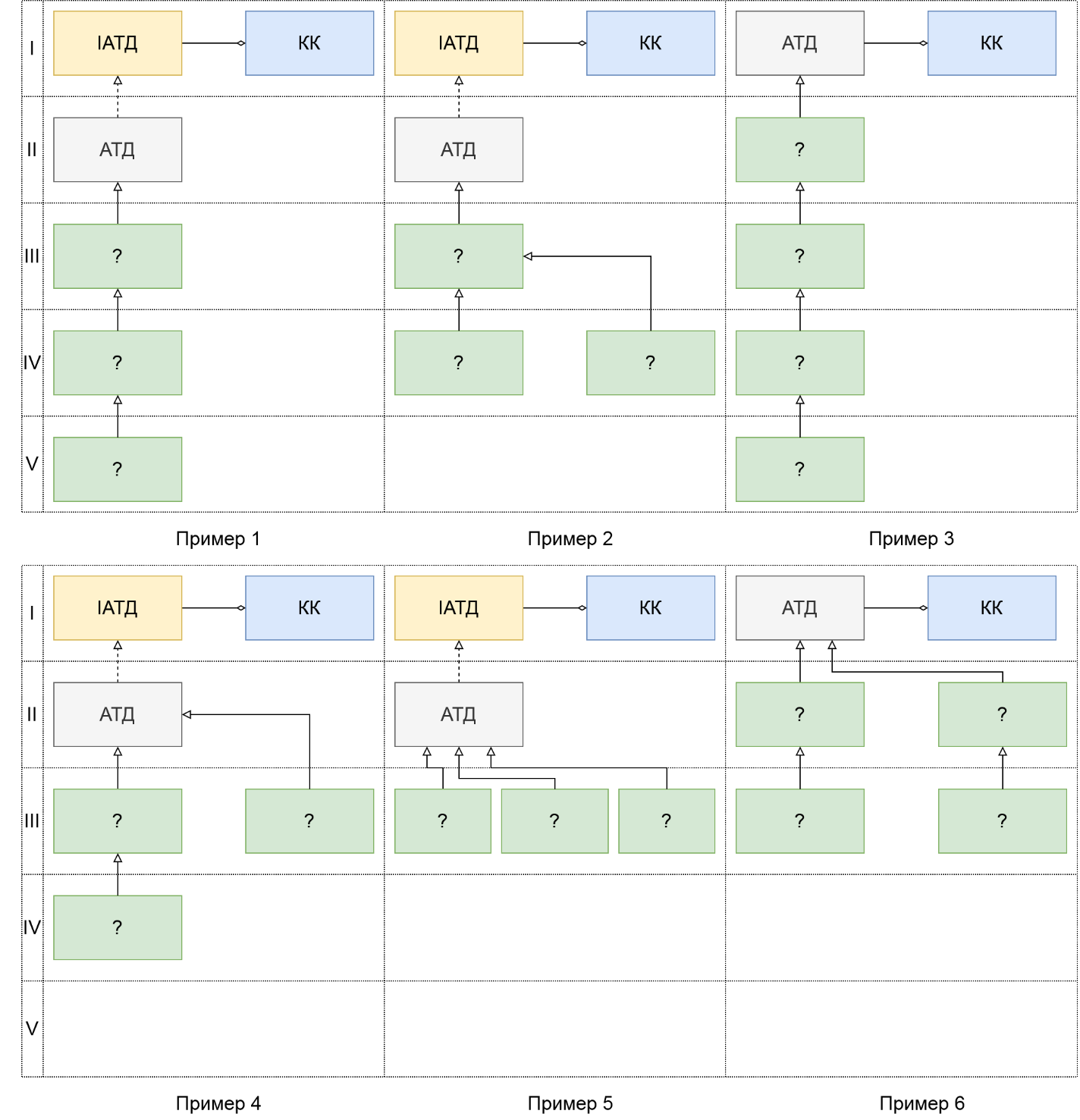
Диагональ: 32 дюйма

….

»

1. Проверьте работоспособность АТД на тестовом наборе данных.

**ПРИМЕРЫ ИЕРАРХИЙ**

****

*Примеры 1,2,4 и 5 для ПИ.*

*Примеры 3 и 6 для ИС.*

Требования к иерархии:

1. Минимум 4 новых классов, включая класс-интерфейс «IАТД» (если такой имеется), напоминаю, что «АТД» и «Класс-контейнер» были уже реализованы ранее.
2. Лимит 5 уровней иерархии.
3. У АТД тоже может быть несколько производных классов, но на данный момент не более трех (на рисунке такие примеры 5 и 6).
4. У каждого производного класса должна быть добавлена как минимум одна новая отличительная характеристика (новое поле), которую нельзя применить к базовому классу.

**КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Дайте определение наследования.
2. Какие модификаторы прав доступа к производному классу вы знаете и в чем их особенности?
3. Как выполняется конструктор при наследовании?
4. Дайте определение чисто виртуальной функции.
5. Какие операторы используются для разыменования указателя на член класса?

**Приложение 1**

**Варианты АТД:**

1. Вычислительные машины
2. Строительные материалы
3. Строительные конструкции
4. Электроинструмент
5. Железнодорожный транспорт
6. Автомобильный транспорт
7. Воздушный транспорт
8. Водный транспорт
9. Энергетические установки
10. Запоминающие устройства
11. Устройства для управления компьютером
12. Телефонные аппараты
13. Бытовая техника
14. Устройства передачи данных
15. Программное обеспечение вычислительной техники
16. Печатающие устройства
17. Офисная мебель
18. Аппаратное обеспечение вычислительной техники
19. Элементная база для сборки компьютеров
20. Аппаратное обеспечение компьютерных сетей
21. Звуковоспроизводящая аппаратура
22. Звукозаписывающая аппаратура
23. Осветительные приборы
24. Оптические приборы
25. Нагревательные приборы
26. Холодильная техника
27. Электрические машины
28. Строительные машины
29. Металлургические машины
30. Сельскохозяйственные машины
31. Элементы интерьера
32. Канцелярские товары
33. Изделия целлюлозно-бумажной промышленности
34. Металлургические технологии
35. Врачебный контроль физиологического состояния спортсмена
36. Печи
37. Камины
38. Инструментальные среды для разработки программного обеспечения
39. Видеомониторы
40. Принтеры
41. Подъемно-транспортные машины
42. Плоттеры

**Приложение 2**

**Варианты КК:**

1. Статический вектор\*

2. Динамический вектор\*

3. Статическая матрица

4. Динамическая матрица

5. Линейный односвязный список\*

6. Линейный двусвязный список\*

7. Стек\*

8. Очередь\*

9. Закольцованный список\*

10. Двоичное дерево\*

\* - обязательно с использованием структуры и указателей.