Last updated: 2019-05-11
vadimov@i.ua

Практическое занятие 13 (семестр2).

Задача 13.1. В Unit 7, Example 7.1 был создан класс coord для хранения целочисленных координат. Создайте родовую версию этого класса, чтобы можно было хранить координаты любого типа. Продемонстрируйте программу решения этой задачи.

Задача 13.2. Переделайте функцию **main()** из Example 13.6 так, чтобы для запрещения вывода на экран объектов типа **NullShape** использовался не оператор **typeid**, а оператор **dynamic_cast**.

Задача 13.3. Будет ли работоспособен следующий фрагмент программы в иерархии классов с базовым классом **Num** из Example 13.9?

```
Num <int> *ptrb;
Square <double> *ptrd;
// ...
ptrd = dynamic_cast <Num <int>> (ptrb);
```

Задача 13.4. В следующей программе имеется ошибка. Исправьте ее с помощью оператора **const_cast**.

```
#include <iostream >
using namespace std;
void f(const double &i) {
    i = 100;
}
int main() {
    double x = 98.6;
    cout << x << endl;
    f(x);
    cout << x << endl;
    return 0;
}
```

Объясните на двух примерах, почему оператор **const_cast** следует использовать только в самых крайних случаях.

Задача 13.5. Допишите следующую программу, чтобы на экран выводилась информация о выбранном пользователем типе объекта.

```
#include <iostream>
#include <typeinfo>
using namespace std;
class A {
    virtual void f() {}
};
class B: public A {};
int main() {
   A *p, obja;
   B objb;
   C objc;
```

```
    int i;
    cout << "Enter 0 for A objects, ";
    cout << "1 for B objects or ";
    cout << "2 for C objects.\n";
    cin >> i;
    if (i==1)
        p = &Objb;
    else if (i==2)
        p = &objc;
    else
        p = &obja;
    // report type of object selected by user
    return 0;
}
```

Задача 13.6. а) Переделайте программу Example 13.3a, чтобы возможную ошибку выделения памяти внутри функции **generator()** отслеживать с помощью механизма обработки исключительных ситуаций.
б) Измените функцию **generator()** из вопроса а), чтобы в ней использовать версию оператора **new** с ключевым словом **nothrow**.

Задача 13.7. Создайте иерархию классов с абстрактным классом **DataStruct** на ее вершине (См. материал Unin 11). В основании иерархии создайте два производных класса. В одном должен быть реализован стек, в другом - очередь. Создайте также функцию **DataStructFactory()** со следующим прототипом:

```
DataStruct *DataStructFactory(char what);
```

Функция **DataStructFactory()** должна создавать стек, если параметр **what** равен **s**, и очередь, если параметр **what** равен **q**. Возвращаемым значением функции должен быть указатель на созданный объект. Продемонстрируйте работоспособность вашей программы.

Совет. Попробуйте поэкспериментировать с RTTI. Хотя польза от динамической идентификации типа в контексте приведенных в Unit 13 простых примеров может показаться не слишком очевидной, тем не менее это мощный инструмент управления объектами во время работы программы.