

C++

Урок 23



Разбор ДЗ

Разбор дз

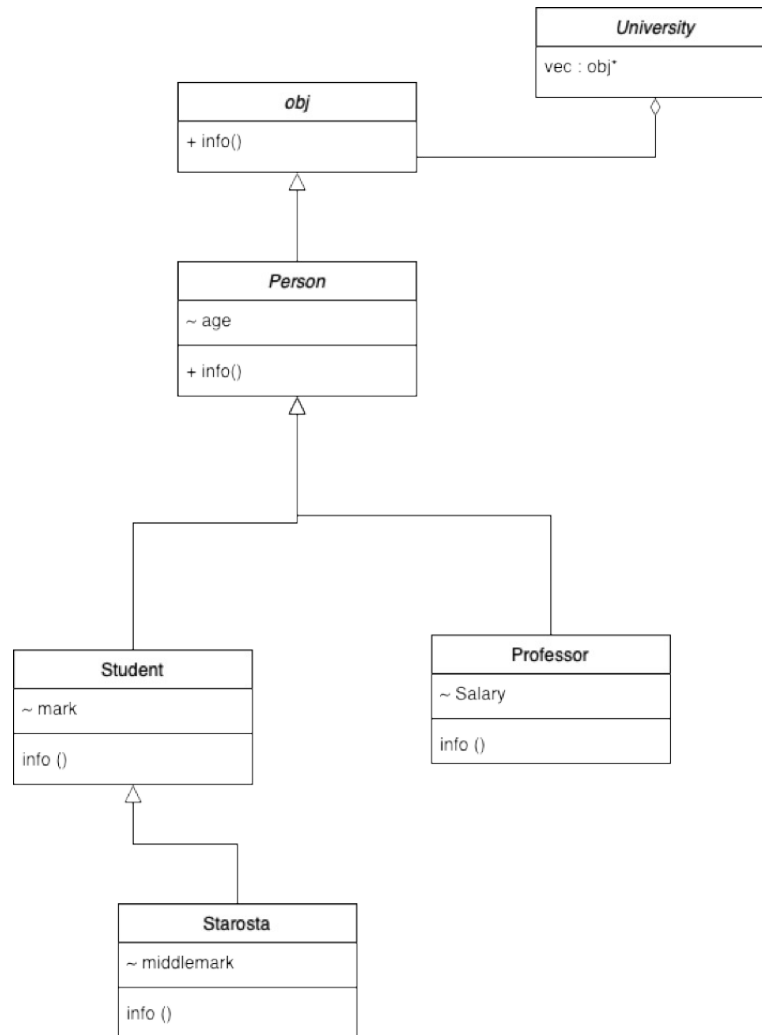


Диаграмма классов

Разбор дз

```
class obj
{
public:
    virtual void info() = 0;
    virtual ~obj() = 0;
};
obj::~~obj() {}
```

Интерфейс obj

Разбор дз

```
class Person : public obj
{
protected:
    int *age;

public:
    Person(int age = 12) : age(new int(age))
    {
        cout << "Конструктор базового" << endl;
    }

    virtual ~Person()
    {
        cout << "Деструктор персона" << endl;
        delete age;
    }
    virtual void info()
    {
        cout << *age << endl;
    }
};
```

Класс Person

Разбор ДЗ

```
class Student : public Person
{
protected:
    int *mark;

public:
    Student(int age = 21, int mark = 4) : Person(age), mark(new int(mark))
    {
        cout << "Конструктор студента" << endl;
    }

    virtual ~Student()
    {
        delete mark;
        cout << "Деструктор студента" << endl;
    }
    virtual void info()
    {
        cout << *age << " " << *mark << endl;
    }
};
```

Класс Student

Разбор ДЗ

```
class Professor : public Person
{
protected:
    int *salary;

public:
    Professor(int age = 21, int salary = 12000) : Person(age), salary(new int(salary))
    {
        cout << "Конструктор профессора" << endl;
    }

    virtual ~Professor()
    {
        delete salary;
        cout << "Деструктор профессора" << endl;
    }

    virtual void info()
    {
        cout << *age << " " << *salary << endl;
    }
};
```

Класс Professor

Разбор дз

```
class Star : public Student
{
protected:
    int *mid;

public:
    Star(int age = 18, int mark = 5, int mid = 5) : Student(age, mark), mid(new int(mid))
    {
        cout << "Конструктор старосты" << endl;
    }

    ~Star()
    {
        delete mid;
        cout << "Деструктор старосты" << endl;
    }

    virtual void info()
    {
        cout << *age << " " << *mark << " " << *mid << endl;
    }
};
```

Класс Star

Разбор дз

```
class University
{
    int size;
    void input()
    {
        obj *mem;
        string str;
        for (int i = 0; i < size; i++)
        {
            cin >> str;
            if (str == "Pers")
            {
                mem = new Person;
                vec.push_back(mem);
            }

            if (str == "Student")
            {
                mem = new Student;
                vec.push_back(mem);
            }

            if (str == "Prof")
            {
                mem = new Professor;
                vec.push_back(mem);
            }

            if (str == "Star")
            {
                mem = new Star;
                vec.push_back(mem);
            }
        }
        mem = nullptr;
        delete mem;
    }

protected:
    vector<obj> >> vec;

    int read(int &size)
    {
        cout << "Укажите количество персон: " << endl;
        cin >> size;
        return size;
    }

public:
    University(int size = 0) : size(size)
    {
        while (size <= 0)
        {
            this->size = read(size);
        }
        cout << "s = " << size << endl;
        input();
    }
    void showInf()
    {
        for (auto &it : vec)
        {
            it->info();
        }
    }
    ~University()
    {
        for (auto &it : vec)
        {
            delete it;
        }
    }
};
```

Класс University

Разбор дз

```
int main()
{
    University Madagaskar;
    Madagaskar.showInf();
}
```

Main функция

**Для чего нужны виртуальные
деструкторы?**

**Какой порядок вызова
конструкторов?**

**Какой порядок вызова
деструкторов?**

A hand-drawn blue oval frame with a slightly irregular, sketchy border. It is centered horizontally and vertically on the page.

Пары

Пары

Пара - переменная, позволяющая хранить в себе два значения.



Пары

Синтаксис:

```
pair <string, double> Izmerenia;
```

Переменная `Izmerenia` будет хранить в себе значение типа **string** и значение типа **double**.

Пары

Синтаксис:

```
pair <string, double> Izmerenia;
```

```
pair<string, double> Izmerenia;
```

```
Izmerenia.first = "Alex"; //доступ к первому значению
```

```
Izmerenia.second = 178.12; //доступ ко второму значению (полю)
```

Пары

Синтаксис:

```
pair<string, double> Izmerenia;  
Izmerenia = make_pair("Alex", 178.12); //присвоить сразу два значения
```

Задача

Задача: создайте пару translator, где будет занесено слово и его перевод на английском языке.

А что если мы хотим много пар?



**В какую структуру их можно
занести?**

Пары

```
#include <iostream>
#include <string>
#include <vector>

using namespace std;

int main()
{
    vector<pair<string, string>> vocabulary;
    string str1, str2;
    int cnt;
    cin >> cnt;
    for (int i = 0; i < cnt; i++)
    {
        cin >> str1 >> str2;
        vocabulary.push_back(make_pair(str1, str2));
    }

    for (auto &it : vocabulary)
    {
        cout << it.first << " " << it.second << endl;
    }
}
```

Задача

Задача: создайте словарь пары translator, где будут занесены слова и их переводы на английском языке.

A hand-drawn blue oval frame with a double-line border, centered on the page. The word "Словари" is written inside this frame.

Словари

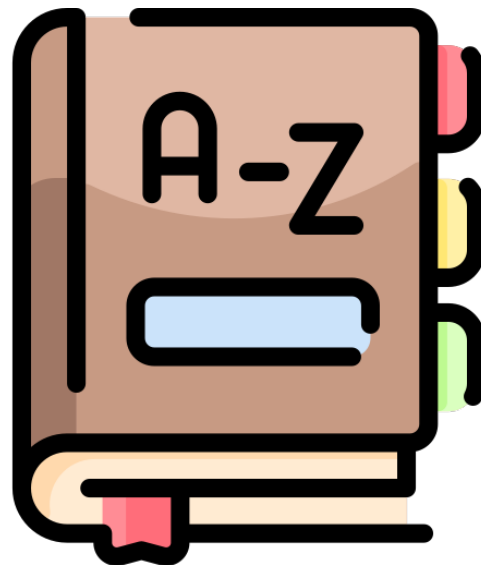


Что такое словарь?

Как вы им пользовались?

Словари

Словарь - сборник слов в алфавитном порядке, с пояснениями, толкованиями или с переводом на другой язык.



Словари

Словарь - сборник слов в алфавитном порядке, с пояснениями, толкованиями или с переводом на другой язык.

Словарь C++ (map) – это ассоциативный контейнер, который работает по принципу — [ключ — значение].

Словари

Словари часто называют также **ассоциативными массивами** или **отображениями**.

Словарь построен на основе пар значений, первое из которых представляет собой ключ для идентификации элемента, а второе — собственно элемент

Словари

Примеры ассоциативных контейнеров:

- `map` – словарь уникальных ключей,
- `multimap` – словарь ключей с дубликатами,
- `set` – множество,
- `multiset` – мультимножество,
- `bitset` – битовое множество (набор битов).

Easy Peasy



It comes from a 1970's [british TV](#) commercial for [Lemon Squeezy](#) detergent. They were with a little girl who points out dirty greasy dishes to an adult (mom or relative) and then this adult produces Lemon Squeezy and they clean the dishes quickly. At the end of the commercial the girl says "[Easy Peasy Lemon Squeezy](#)".

Today it is a silly way to state something was or will be very easy.

I will be [in an out](#), [easy peasy](#).

Can you open this [jar of pickles](#)? Sure thing easy peasy.

by [Creepy Gnome](#) August 2, 2009

**Приведите свой пример
ассоциативного массива**

Словари

```
#include <map>
```

- Подключаем библиотеку

Создание словаря

Словари

```
#include <map>
```

- Подключаем библиотеку

```
map<тип ключа, тип значения> название;
```

Создание словаря

Словари

```
#include <map>
```

- Подключаем библиотеку

```
map<тип ключа, тип значения> название;
```

```
int main()
{
    map<string, string> urbanDict;    //создали массив с ключем и значением типа string
    map<string, double> medicalMeasure;
    //создали массив с ключем типа string, а значением типа double
}
```

Создание словаря

Словари

```
map<string, double> medicalMeasure = {"Marti", 14.5},  
                                       {"Gloria", 10.15},  
                                       {"Melmon", 30.15}};
```

Инициализация словаря в момент
объявления

Словари

```
int main()
{
    map<string, double> medicalMeasure;

    medicalMeasure["Marti"] = 14.5;
    medicalMeasure["Gloria"] = 10.15;
    medicalMeasure["Melmon"] = 30.15;

    cout << medicalMeasure["Melmon"] << endl; // 30.15
}
```

Добавление элемента

Словари

```
int main()
{
    map<string, double> medicalMeasure;

    medicalMeasure["Marti"] = 14.5;
    medicalMeasure["Gloria"] = 10.15;
    medicalMeasure["Melmon"] = 30.15;
    medicalMeasure["Gloria"] = 25.15;

    cout << medicalMeasure["Gloria"] << endl;
}
```

Что будет выведено?

Словари

```
int main()
{
    map<string, double> medicalMeasure;

    medicalMeasure["Marti"] = 14.5;
    medicalMeasure["Gloria"] = 10.15;
    medicalMeasure["Melmon"] = 30.15;

    // 1 способ
    for (auto &it : medicalMeasure)
    {
        cout << it.first << " " << it.second << endl;
    }

    // 2 способ
    for (auto it = medicalMeasure.begin(); it != medicalMeasure.end(); it++)
    {
        cout << it->first << " " << it->second << endl;
    }

    // 3 способ
    for (auto it = medicalMeasure.begin(); it != medicalMeasure.end(); it++)
    {
        cout << (*it).first << " " << (*it).second << endl;
    }
}
```

Вывод эл-тов

Задача

Задача: Реализуйте словарь buildings, в котором содержится информация о самых высоких зданиях в мире.

Входные данные: ввод информации в словарь с **клавиатуры**.

Выходные данные: высота здания по его названию.

Задача

Задача: Реализуйте словарь buildings, в котором содержится информация о самых высоких зданиях в мире.

Входные данные: название строения.

Выходные данные: высота строения.

Словари

Ассоциативные массивы относятся к STL контейнерам.
Также в STL есть vector.

Словари

Ассоциативные массивы относятся к STL контейнерам.
Также в STL есть vector.

Какие методы мы использовали в vector?

Словари

Ассоциативные массивы относятся к STL контейнерам.
Также в STL есть vector.

Какие методы мы использовали в vector?

- `find(ключ)`
- `count (ключ)`
- `erase (ключ)`
- `size()`
- `clear()`
- `empty()`
- `swap(контейнер)`

Словари

Ассоциативные массивы относятся к STL контейнерам. Также в STL есть vector.

Какие методы мы использовали в vector?

- `find(ключ)` – возвращает итератор на найденный элемент. Если не нашел эл-т, то итератор на конец.
- `count (ключ)` - возвращает кол-во эл-тов по заданному ключу.
- `erase (ключ)`
- `size()`
- `clear()`
- `empty()`
- `swap(контейнер)`

[map документация](#)

Словари

Ассоциативные массивы относятся к STL контейнерам. Также в STL есть vector.

Какие методы мы использовали в vector?

- `find(ключ)` – возвращает итератор на найденный элемент. Если не нашел эл-т, то итератор на конец.
- `count (ключ)` - возвращает кол-во эл-тов по заданному ключу. В `map` вернет либо 0, либо 1.
- `erase (ключ)` – удаляет эл-т по заданному ключу.
- `size()` – кол-во эл-тов в словаре.
- `clear()` – удаление всех эл-тов из словаря.
- `empty()` – определяет пустой ли словарь.
- `swap(контейнер)` – меняет элементы между двумя сопоставлениями

Задача

Задача: Реализуйте словарь buildings, в котором содержится информация о самых высоких зданиях в мире.

Входные данные: ввод информации в словарь с **клавиатуры**.

Выходные данные: высота здания по его названию.