

C++ course

*name: Виктория Александровна
e-mail: kewtree1408@gmail.com*

*source: <https://bitbucket.org/Kewtree/coursecpp/src>
login: coursecpp
password: 12345*

General Plans

1. Inception
- 2-3. Statements
4. Functions
5. Pointers
6. Arrays
7. Structures
8. Files, OOP: encapsulation
9. OOP: inheritance
10. OOP: polymorph
11. Test
12. Game over

Plan. 7 lesson

0. File stream (begin)

1. Quick Sort

2. Strings

3. OOP: begin

4. Struct

5. Union

File stream

Файловый поток

Работа как с обычными `cin`, `cout`. Операции записи в файл и извлечения из файла -- аналогичные `>>`, `<<`.

Реализуется:

Включениями

```
#include <ofstream> - output file stream
```

```
#include <ifstream> - input file stream
```

```
ofstream output("name.txt"); ifstream input("name.txt");  
output << "Hello";           input >> s;
```

File stream

Example: `iofile.cpp`

Работа с файлами на пример ввода-вывода чисел

Sort

1. Bubble sort - пузырьковая
2. Shaker sort - шейкерная
3. Changed sort - обменная
4. Quick sort - быстрая
5. Shell sort - сортировка Шелла
6. Heap sort - пирамидальная
7. Radix sort - поразрядная
- ...

Introspection sort in C++

Quick Sort

Алгоритм быстрой сортировки:

Краткое описание алгоритма

1. выбрать элемент, называемый опорным.
2. сравнить все остальные элементы с опорным, на основании сравнения разбить множество на три — «меньшие опорного», «равные» и «большие», расположить их в порядке меньше-равные-большие.
3. повторить рекурсивно для «меньших» и «больших».

Быстрая сортировка использует стратегию «разделяй и властвуй».

Quick Sort

Example: `quick_sort.cpp`

Strings

Строка - массив символов.

2 способа:

1. С-строка

2. Класс String из STL-библиотеки

Пример:

```
char dog[6] = {'c', 'h', 'a', 'p', 'p', 'i'}; //НЕ  
строка  
char cat[5] = {'f', 'i', 's', 'h', '\0'};  
//строка
```

Strings

Example: `init_cstring.cpp`

Работа со строками

Strings

Example `strip.cpp`

Написать функцию, которая удаляет начальные и конечные пробелы из введенной с клавиатуры строки.

(в Python эта функция уже готова и называется `strip()`)

Strings

Какие проблемы возникают с С-строками?

1. массив
2. работа с динамической памятью
3. сложность операций
4. специальные функции из `<string.h>`,
`<ctype.h>`

Strings

Таблица 9.1. Некоторые функции из библиотеки обработки символов **ctype.h**

<i>Прототип функции</i>	<i>Описание функции</i>
int isdigit(int c)	Возвращает значение true , если c является цифрой, и false в других случаях
int isalpha(int c)	Возвращает значение true , если c является буквой, и false в других случаях
int isalnum(int c)	Возвращает значение true , если c является цифрой или буквой, и false в других случаях
int islower(int c)	Возвращает значение true , если c является буквой нижнего регистра, и false в других случаях
int isupper(int c)	Возвращает значение true , если c является буквой верхнего регистра, и false в других случаях
int tolower(int c)	Если c является буквой верхнего регистра, то результат – буква нижнего регистра, в других случаях возвращается аргумент без изменений
int toupper(int c)	Если c является буквой нижнего регистра, то результат – буква верхнего регистра, в других случаях возвращается аргумент без изменений

Таблица 9.2. Некоторые функции преобразования строк из библиотеки **stdlib.h**

<i>Прототип функции</i>	<i>Описание функции</i>
double atof(const char *s)	Преобразует строку s в тип double
int atoi(const char *s)	Преобразует строку s в тип int
long atol(const char *s)	Преобразует строку s в тип long int

Таблица 9.3. Некоторые функции из библиотеки **string.h**

<i>Прототип функции</i>	<i>Описание функции</i>
size_t strlen(const char *s)	Вычисляет длину строки s в байтах
char *strcat(char *s1, const char *s2)	Присоединяет строку s1 в конец строки s2
char *strcpy(char *s1, const char *s2)	Копирует строку s1 в место памяти, на которое указывает s2
char *strncat(char *s1, const char *s2, size_t maxlen)	Присоединяет строку maxlen символов строки s2 в конец строки s1
char *strncpy(char *s1, const char *s2, size_t maxlen)	Копирует maxlen символов строки s2 в место памяти, на которое указывает s1
char * strstr(char *s1, char *s2)	отыскивает позицию первого вхождения строки s2 в строку s1
int strcmp(const char *s1, const char *s2)	сравнивает две строки в лексикографическом порядке с учетом различия прописных и

OOP

1. Class. Класс
2. Object. Объект
3. Instance. Экземпляры
4. Our types. Пользовательские типы

Struct

```
struct Part{  
    int modelnumber;  
    int partnumber;  
    float cost;  
};
```

```
Part p; // объявили переменную собственного типа  
p.modelnumber = 1;  
p.partnumber = 2;  
p.cost = 12.34;
```

```
sizeof(p); // 4+4+4 = 12bytes
```

Struct

Example:

1. employee.cpp
2. complex.cpp

Union

```
union Distance{  
    int km;  
    double sm;  
};
```

Предполагается, что в любой момент времени будет использоваться только одна переменная
`sizeof(Distance) == 8 bytes (not 12)`

Union

Example: union.cpp

Homework

0) Пусть у нас есть декартова система координат.

0.1) Ввести со стандартного ввода одну точку и вывести ее: "first point: <x;y>", где вместо x и y соответствующие координаты.

0.2) Автоматически сгенерировать 10 точек, вывести их на стандартный вывод и найти сумму соответствующих координат x и y.

1) Создать структуру Рюкзак. У Рюкзака есть 3 параметра: тип, стоимость и объем.

Из консоли мы считаем информацию о 3-х рюкзаках. А затем спросим у пользователя: какое кол-во одинаковых по форме предметов и какого объема он захочет положить в этот рюкзак?

Как только ввели нужное кол-во данных (6 чисел для кол-ва предметов и их объема соответственно), программа должна выдать результат для каждого рюкзака: вместится ли туда указанное кол-во предметов заданного объема или нет.

Пример выполнения:

На вход:

small 1000 56.8

middle 2800 108.6

big 1004.7

Homework

Требуется ввести кол-во данных:

5 12

no

1 89

yes

4 25

yes

2) Требуется, чтобы программа по введенным данным считывала и обрабатывала всю информацию о школьниках и проводила выборку. Нужно отобрать школьников с баллом > 4 и вывести их и на стандартный вывод.

Доступная информация о школьниках: Имя, Фамилия, номер школы, класс, средний балл. Количество школьников задается пользователем (со стандартного ввода)

3*) Создать структуру с информацией о пассажирах самолета: фамилия, инициалы, количество вещей, общий вес вещей, время вылета, пункт назначения, наличие пересадок, сведения о детях. Заполнить структуру данными. По этим данным найти пассажиров, вес багажа которых отличается от максимального веса менее чем на X кг. X -- вводится с клавиатуры.

4*) Разобрать пример со стеком и задать вопросы.