

# Алгоритмизация и программирование

## Лекция 1

# О преподавателях

Чабанов Владимир Викторович, старший преподаватель Кафедры компьютерной инженерии и моделирования Физико-технического института.

**Кафедра:** 310А

**E-mail:** [chabanov.vv@cfuv.ru](mailto:chabanov.vv@cfuv.ru)

**VK:** <https://vk.com/id444710087>

# О курсе

- Год поступления студентов: 2021;
- Всего: 180 часов = 32 аудиторная работа + 148 самостоятельная работа;
- Аудиторная работа: 32 часа = 16 часов лекций + 16 часов практики;
- Самостоятельная работа: 148 часов = 112 + 36 (подготовка к экзамену);
- Аттестация: экзамен;
- Основной язык: C++;

*\* Академический час - 45 минут, т.е. пол пары.*

Дни 1 - 10

Выучить переменные, константы, массивы, строки, выражения, функции...



Дни 11 - 21

Выучить потоки, указатели, ссылки, классы, объекты, наследование, полиморфизм...



Дни 22 - 697

Много программировать для себя. Иногда взламывать что-то, но все время учиться на ошибках.



Дни 698 - 3648

Общаться с другими программистами. Работать над проектами с ними. Учиться у них.



Дни 3649 - 7781

Выучить продвинутую теоретическую физику и сформулировать теорию квантовой гравитации.



Дни 7782 - 14611

Выучить биохимию, молекулярную биологию, генетику...



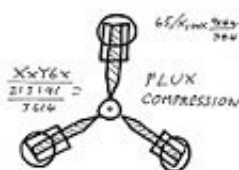
День 14611

Использовать знания по биологии для создания омолаживающего зелья.



День 14611

Использовать знания по физике для создания поточного конденсатора и вернуться в день 21.



День 21

Заменить себя-из-прошлого.



# Материалы курса

- Курс на мудле: <https://moodle.cfuv.ru/course/view.php?id=21690>;
- Материалы на GitHub: [https://github.com/VladimirChabanov/alg\\_and\\_prog\\_2year](https://github.com/VladimirChabanov/alg_and_prog_2year);

# Система оценивания

Фрагмент приказа №135 от 11.02.2020 "Об утверждении Порядка применения балльно-рейтинговой системы оценивания успеваемости обучающихся по программам ВО в ФГАОУ ВО "КФУ им. В.И. Вернадского":

## 3.13. Экзамен

3.13.1. Рейтинговые баллы промежуточной аттестации по дисциплине (модулю)  $B_{ПА}$ , изучаемой в одном семестре, определяется по сумме баллов текущего контроля успеваемости в семестре  $B_{сем}$ , и баллов, полученных при сдаче экзамена  $B_{ЭКЗ}$ , т. е.

$$B_{ПА} = B_{сем} + B_{ЭКЗ}.$$

# Система оценивания

3.13.2. Ответ на экзамене оценивается в диапазоне от 0 до 40 ( $0 \leq B_{\text{ЭКЗ}} \leq 40$ ).

Ответ на экзамене, оцененный меньше 20 баллов считается неудовлетворительным – обучающемуся за экзамен выставляется нулевой рейтинговый балл ( $B_{\text{ЭКЗ}} = 0$ ).

Шкала соответствия рейтингового балла за экзамен  
и оценки в 5-балльной системе

Оценка за экзамен в 5-балльной системе	Рейтинговый балл за экзамен ( $B_{\text{ЭКЗ}}$ )
отлично	$35 \leq B_{\text{ЭКЗ}} \leq 40$
хорошо	$28 \leq B_{\text{ЭКЗ}} < 35$
удовлетворительно	$20 \leq B_{\text{ЭКЗ}} < 28$
неудовлетворительно	$0 \leq B_{\text{ЭКЗ}} < 20$

# Система оценивания

3.13.3. Экзаменационная оценка по дисциплине (модулю) выставляется в соответствии со шкалой пересчета рейтинговых баллов в оценку по 5-балльной системе.

Шкала пересчета рейтингового балла по дисциплине (модулю)  
в экзаменационную оценку в 5-балльной системе

Рейтинговый балл по дисциплине ( $B_{ПА}$ )	Оценка по дисциплине в 5-балльной системе
$90 \leq B_{ПА} \leq 100$	отлично
$71 \leq B_{ПА} < 90$	хорошо
$53 \leq B_{ПА} < 71$	удовлетворительно
$0 \leq B_{ПА} < 53$	неудовлетворительно



# Система оценивания

Балл за **работу в семестре** определяется как сумма баллов по всем контрольным точкам (55 баллов) + бонусные баллы (5 баллов):

- Лекционные занятия:
  - Тест после пары: 4 балла;
- Практические занятия:
  - Посещение: 4 балла;
  - Тест в начале пары: 8 баллов;
  - Тест после пары: 4 балла;
  - Решение задач: 30 баллов;
  - Выполнение кейса: 5 баллов;
- Бонус: 5 баллов (дополнительно начисляются за выполнение кейса).

# Система оценивания

Экзамен проходит в 2 этапа:

- Тестирование. Проверяет полноту освоения курса (простые вопросы, но по всем темам);
- Опрос по билетам. Проверяет глубину освоения материала (несколько вопросов, но подробно).

Каждый этап оценивается отдельно, по 100 балльной шкале. Итоговая оценка за экзамен определяется по формуле:

$$B_{\text{экз}} = 40 \cdot \frac{B_1}{100} \cdot \frac{B_2}{100}$$

Если за первый этап получена оценка ниже 50 баллов, то за весь экзамен выставляется оценка *НЕудовлетворительно*, т.к.  $B_{\text{экз}}$  гарантировано будет меньше 20 баллов.

# Практика

Крайне желательно приносить с собой ноутбук.

Практические и контрольные задания размещены в системе [Яндекс.Контест](#).

Доступ к практическим заданиям: [заполните форму](#);

# Опрос

[Опрос](#)

# Где писать код?

- Локально на своей машине;
- Онлайн-компиляторы.

# Онлайн-компиляторы

## [Wandbox](#)

- доступно большое количество языков (не только C++);

## [Compiler Explorer](#)

- доступно большое количество языков (не только C++);
- для C++ доступно множество различных компиляторов в том числе экспериментальных;
- позволяет посмотреть ассемблерный код и сравнить его для разных вариантов сборки;
- есть встроенная поддержка некоторых популярных библиотек;

## [OnlineGDB](#)

- можно запустить дебагер.

# Что такое код?

Это текстовый файл с определённым расширением (для C++: .cpp .h .hpp);

Код должен быть написан в соответствии с "правилами" языка ([стандарт языка](#)).

# Минимальная программа

Минимальная программа на C++: <https://wandbox.org/permlink/0wrO9G4BX8yBQobx>

Варианты функции main: <https://wandbox.org/permlink/BkvEivNhbGzB3WKP>



# Комментарии

<https://wandbox.org/permlink/g919ArA0C3dqefZm>

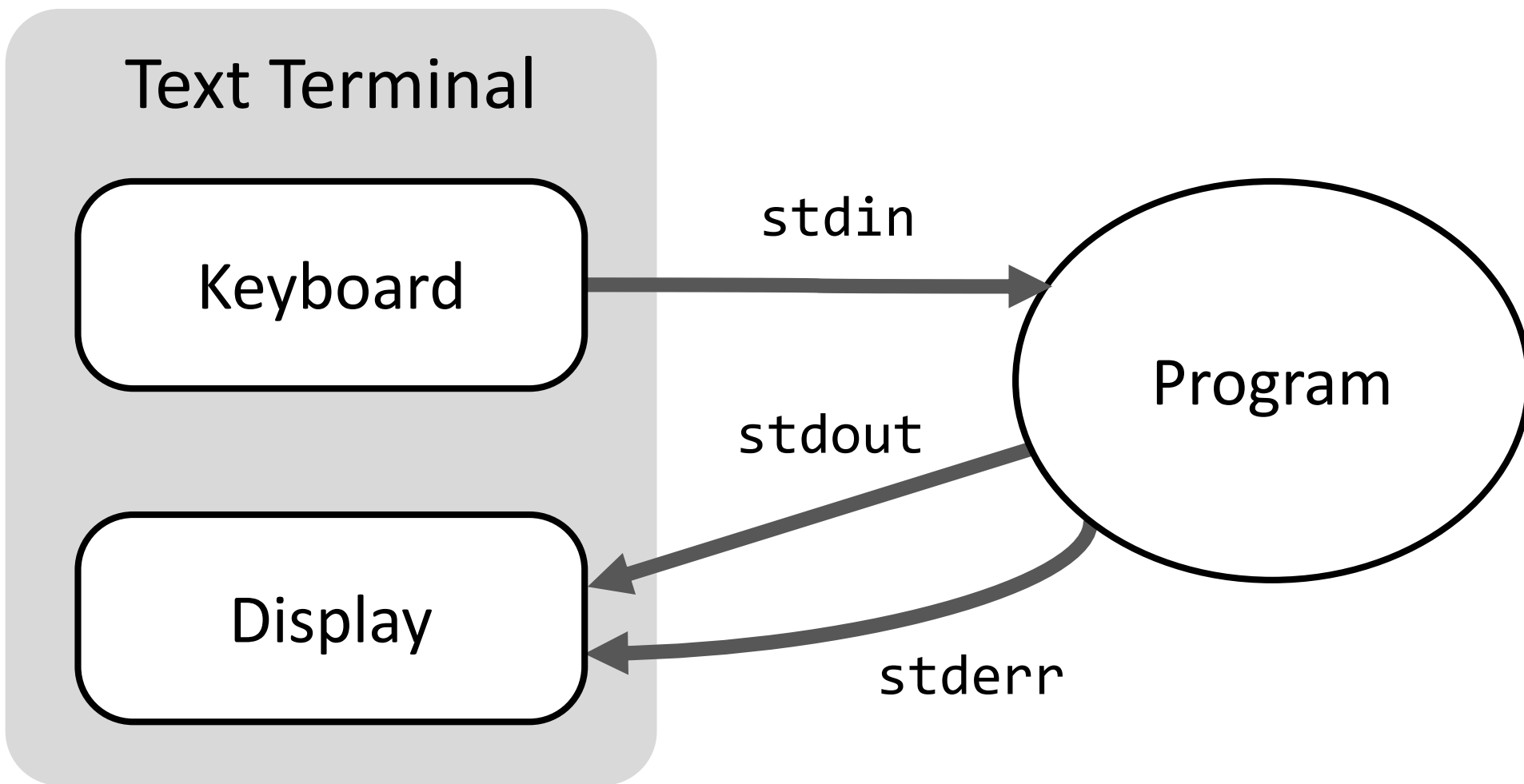
# Учебная задача 1

Выведите на экран консоли две строки: "Wake up, Neo..." и "The Matrix has you..".

Вторую вывести под первой.

<https://godbolt.org/z/13sz51z7x>

# Стандартные потоки



# Строковый и символьный литералы

Строковый литерал – это последовательность символов заключённую в **двойные кавычки**:

"Hello, World"

Сырой строковый литерал: R"(Hello "Harlan's" world)"

Символьный литерал – это один или несколько символов заключённые в **одинарные кавычки**: 'A', '\n'

Escape-последовательности – это управляющая последовательность: "Hello\nWorld"

[Подробнее ->](#)

# Вывод сообщений об ошибке

<https://wandbox.org/permlink/cVarPaK0CTP1FOk5>

# Сброс буфера потока вручную

Символ '\n' просто переводит вывод на новую строку; `std::endl` дополнительно выполняет сброс буфера

`std::flush`

`std::endl`

# Учебная задача 2

Выведите на экран строку: "Проснись, Нео...".

<https://wandbox.org/permlink/PYdzmMurBeD3cRNh>

# Крокозябры

Unix:

```
Проснись, Нео...  
Exit Code: 0
```

Windows:

```
Консоль отладки Microsoft Visual Studio  
±Ёюёэшё№, =хю...  
C:\Users\Professional\Desktop\Box\ConsoleApplication1\Release\ConsoleApplication1.exe (процесс 1196) завершил работу с кодом 0.  
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".  
Нажмите любую клавишу, чтобы закрыть это окно...
```



# setlocale

Магия которая позволяет побороть крокозябры, но работает не всегда.

Это не единственное решение, есть ещё множество вариантов.

Исходники должны быть в кодировке 1251

```
#include <windows.h>

SetConsoleCP(1251);          // установка кодовой страницы win-ср 1251 в поток ввода
SetConsoleOutputCP(1251);    // установка кодовой страницы win-ср 1251 в поток вывода

system("chcp 1251");
```

<https://wandbox.org/permlink/zEhVfp9IF76ObKvA>

# Откуда берутся крокозябры

Кодировка Windows-1251 (программа)

Кодировка 866 (консоль)

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
8.	Ђ А	Ѓ Б	Ѕ В	Ї Г	„ Д	… Е	† Ж	‡ З	€ И	‰ Й	Љ К	‹ Л	Њ М	Ќ Н	Ћ О	Ц П
9.	ђ Р	‘ С	’ Т	“ У	” Ф	• Х	— Ц	— Ч		™ Щ	љ Ъ	› Ы	њ Ь	ќ Э	ћ Ю	џ Я
A.		Ў а	ў б	Ј г	Ѡ д	Г е	Ј ж	§ з	Ё и	© й	Є к	« л	¬ м		® н	Ї о
B.	° ■	± ■	І ■	і 	г └	μ ≡	¶ 	· π	ё э	№ 	е 	» ┘	ј ┘	Ѕ ┘	ѕ ┘	ї ┘
C.	А L	Б └	В └	Г └	Д —	Е └	Ж └	З 	И L	Й └	К └	Л └	М └	Н =	О └	П └
D.	Р └	С └	Т π	У └	Ф └	Х F	Ц π	Ч 	Ш └	Щ └	Ъ Г	Ы ■	Ь ■	Э └	Ю └	Я ■
E.	а р	б с	в т	г у	д ф	е х	ж ц	з ч	и ш	й щ	к ъ	л ы	м ь	н э	о ю	п я
F.	Р Ё	С ё	Т Є	У е	Ф Ї	Х ї	Ц Ў	Ч ў	Ш °	Щ ·	Ъ ·	Ы √	Ь №	Э Ѡ	Ю ■	Я

# Учебная задача 3

Напишите программу, которая запрашивает у пользователя имя (т.е. выводит текстовое сообщение с просьбой вести имя), после чего отображает на экране текст: "Привет, {Имя пользователя}".

Имя может состоять из нескольких слов, разделённых пробелом.

<https://wandbox.org/permlink/31tYRaInrIBFtzOc>

# Пробельные символы

Символы пробел, табуляция, перевод строки, возврат каретки, новая страница, вертикальная табуляция и новая строка называются пробельными, поскольку они имеют то же самое назначение, что и пробелы между словами и строками в тексте на естественном языке. Эти символы отделяют друг от друга лексемы, например константы и идентификаторы.

# Переменная

```
std::string name = "James Bond";
```

Значение ("James Bond")

Идентификатор/имя (name)  
может быть 0 или больше

Переменная

Тип (std::string)

Адрес (0x7ffd7ca6b9a0)

# Переменная



CREATE



READ



UPDATE



DELETE

---

C

R

U

D

# Переменная

Создаём переменные:

```
std::string name = "James Bond";  
int answer = 42;  
double trash;
```

Модифицируем:

```
name = "James";  
answer = 43;  
trash = -1.0;
```

Читаем:

```
std::cout << name << ' ' << answer << ' ' << trash << std::endl;
```

<https://wandbox.org/permlink/kaZTchDsod9Tune0>

# One definition rule (ODR)

Definition – определение.

```
int answer;
```

```
int answer = 42; // ошибка
```



# Область видимости (scope)

Каждое **имя**, которое появляется в программе на C++ действительно только в пределах некоторой части исходного кода, которая называется его *областью видимости*.

В пределах области видимости поиск неполного имени можно использовать для связывания имени с его объявлением.

Области видимости могут быть вложены друг в друга.

<https://wandbox.org/permlink/5h6GTJDDUIkgN5gG>

# Учебная задача 4

Выведите на экран целое число 1337 и число  $\pi$  с точностью до 2го знака после запятой.

[Целочисленный литерал](#)

[Вещественный литерал](#)

[Манипуляторы вывода](#)

<https://wandbox.org/permlink/D43p9IH5dnVrMhat>

# Учебная задача 5

Пользователь через пробел вводит два числа. Первое целое, а второе вещественное. Нужно получить эти числа и вывести их в обратном порядке (сначала 2е, затем 1е)

[stoi](#)  
[stod](#)  
[to\\_string](#)  
[auto](#)

<https://wandbox.org/permlink/tGt5JH6N6ihdX5TI>

<https://wandbox.org/permlink/AGSfq3yuvR3gulz5>

# Учебная задача 6

Дана строка, в которой, через запятую перечислены 3 суммы в рублях. Все числа целые, без копеек. После числа, без пробела, написано слово **рубль** в соответствующем падеже. Например: "20рублей, 1рубль, 25рублей". Выведите итоговую сумму денег на экран (просто число).

[ignore](#)

[Арифметические операции](#)

<https://wandbox.org/permlink/w5orQu7iTxO83Emv>

<https://wandbox.org/permlink/tms9z0eFCpPHYDJa>

<https://wandbox.org/permlink/c6xsZrWvUbXUTm7K>

# Учебная задача 7

Существует лотерея в которой можно выиграть джекпот угадав 6 номеров из 36 доступных. Максимум в билете можно отметить 6 номеров (то есть нужно угадать все выпавшие).

Определите вероятность выиграть джекпот:  $\frac{1}{36} \cdot \frac{1}{35} \cdot \frac{1}{34} \cdot \frac{1}{33} \cdot \frac{1}{32} \cdot \frac{1}{31} = \frac{1}{1\,402\,410\,240}$

[Переполнение](#)

[Неопределённое поведение](#)

[numeric limits](#)

[sizeof](#)

<https://wandbox.org/permlink/2mk5vioEK8DXFR1J>

# Учебная задача 8

Вычислите результат выражения:  $x_0 + v_0 \cdot t + \frac{a^2 \cdot t}{2}$  и выведите его на экран, с точностью 4 значащих цифры, при условии, что  $a = 9.8$ . Остальные данные вводятся пользователем в порядке  $x_0$ ,  $v_0$ ,  $t$  каждое значение в отдельной строке.

<https://wandbox.org/permlink/c5B7bnP8RGY9NTg0>

# Учебная задача 9

Дана пара целых чисел. Нужно разделить первое на второе и вернуть результат округлённый до ближайшего целого.

[round](#)

[Явное преобразование типов](#)

<https://wandbox.org/permlink/KB3rXoIEaAlxM1XO>

# Учебная задача 10

На вход подаются два числа разделённые пробелом. Сравните их, и выведите на экран **true**, если первое число больше второго и **false**, для остальных случаев.

[Операторы сравнения](#)

<https://wandbox.org/permlink/DOn1wkQdRayd8t32>



# assert

Позволяет проверять утверждения. Эта конструкция может автоматически сигнализировать в случае, если переданное ей утверждение ложное (false), что обычно приводит к аварийному завершению программы с указанием места обнаружения некорректных данных.

[assert](#)

[static assert](#)

<https://wandbox.org/permlink/8owSOyhi66nxgwdY>

# Инструменты

Visual Studio Code: <https://code.visualstudio.com/>

Visual Studio Community: <https://visualstudio.microsoft.com/ru/vs/community/>

git: <https://git-scm.com/>