

# Введение

## Си, разбор задач



Санкт-Петербургский  
государственный университет

# Критерии оценивания

- Шкала оценивания ECTS, оценки от A до F
- Надо набирать баллы:
  - За домашки (их будет много!)
  - За две контрольные
  - За зачёт, который по сути большая контрольная
- Итоговый балл за домашки:  $\max(0, (n/N - 0.6)) \times 2.5 \times 100$ 
  - Если сделано меньше 60% — это 0, если 80% — 50 баллов
  - Зачёт — строго больше 50 баллов, так что 80% — минимум
- Есть дедлайны (минус балл к максимуму за каждую неделю, но не больше половины баллов)
- Итоговый балл за контрольные:  $n/N \times 100$ , их можно переписывать
- Балл за зачёт считается так же, но переписывать можно только трижды
- В качестве итогового берётся **минимум** из этих баллов

# Шкала оценивания ECTS

Балл	Оценка ECTS
90–100	A
80–89	B
70–79	C
61–69	D
50–60	E
0–50	на пересдачу

# Что будет в I семестре

- Ликвидация безграмотности по программированию на Си
- Отладка и тестирование
- Инструменты разработчика
- Внутреннее представление данных
- Работа с указателями, стеки, очереди, списки и т.п.

# Советы по организации работы

- Windows хорош наличием Visual Studio со встроенным компилятором
- Установите себе Linux
  - EndeavourOS: <https://endeavouros.com> (помни про <https://wiki.archlinux.org>)
  - Ubuntu: <https://ubuntu.com> (Debian тоже хорош!)
  - WSL2 (не совсем Linux)
- Под Linux вполне хватит компилятора gcc или clang и Visual Studio Code с расширением для Си (Microsoft или clangd)
- Используйте командную оболочку
  - Linux: bash, zsh, fish
  - Windows: cmd.exe, PowerShell, MSYS2
- И утилиты (Linux-only): ls/zoxide, nano/vim, cat/bat, less, grep/ripgrep, find/fd, mc/yazi

# Небольшое введение в Си

```
#include <stdio.h>

void display_abs(int input_value) {
    if (input_value < 0) {
        int tmp_value;
        tmp_value = -input_value;
        printf("displaying %d\n", tmp_value);
    } else { printf("displaying %d\n", input_value); }
}

int main() {
    display_abs(-3);
    display_abs(8);
    return 0;
}
```

# Как запустить

```
$ gcc 01_basic_c.c -Wall -o 01_basic_c
```

```
$ ls -l 01_basic_c
```

```
-rwxr-xr-x 1 wowaster wowaster 15456 сен  3 12:38 01_basic_c
```

```
$ ./01_basic_c
```

```
displaying 3
```

```
displaying 8
```

# Указатели

```
#include <stdio.h>
```

```
int main() {  
    int x = 1;  
    int *y = &x;  
    *y = 2;  
  
    printf("%d\n", x);  
  
    return 0;  
}
```

Что выведет данная программа?



# Указатели для передачи параметров

```
#include <stdio.h>
```

```
void value_incrementer(int in_value) { in_value += 1; }
```

```
void pointer_incrementer(int *in_value) { *in_value += 1; }
```

```
int main(int argc, char **argv) {  
    int val = 34;  
    int *pointer_to_val = &val;  
  
    value_incrementer(val);  
    printf("val %d\n", val);  
  
    pointer_incrementer(pointer_to_val);  
    printf("val %d\n", val);  
  
    return 0;  
}
```

# Домашнее задание

## Задача № 1

Написать «Hello, world!» на Си

Примерный план решения:

- 1 Зарегистрироваться на HwProj
- 2 Установить компилятор
- 3 Настроить среду разработки
- 4 Написать код
- 5 Скомпилировать программу
- 6 Запустить программу
- 7 Сделать скриншот работы
- 8 Сдать на HwProj текст программы и скриншот

- Спецификация Си с примерами
  - <https://en.cppreference.com/w/c.html>
- ВикиКнига «C Programming»
  - [https://en.wikibooks.org/wiki/C\\_Programming](https://en.wikibooks.org/wiki/C_Programming)
  - Раздел «Intro exercise» — подробная инструкция по выполнению домашнего задания
- Тьюриал по Си от команды среды рабочего стола для Linux Enlightenment
  - <https://www.enlightenment.org/docs/c/start>

## Условия задач с теста (если останется время)

- 1 Написать алгоритм нахождения неполного частного от деления  $a$  на  $b$  (целые числа), используя только операции сложения, вычитания и умножения.
- 2 Подсчитать число «счастливых билетов» (билет считается «счастливым», если сумма первых трёх цифр его номера равна сумме трёх последних).
- 3 Написать алгоритм проверки баланса скобок в исходной строке (т.е. число открывающих скобок равно числу закрывающих и выполняется правило вложенности скобок).
- 4 Какое наименьшее количество операций умножения достаточно для вычисления значения формулы  $x^4 + x^3 + x^2 + x + 1$ ?