

# ИСПОЛЬЗОВАНИЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В УЧЕБНОМ КУРСЕ “ОСНОВЫ ПРОГРАММИРОВАНИЯ”

Г. Злобин, П. Рыковский, А. Чмыхало

Львовский национальный университет имени  
Ивана Франко

*zlobin@electronics.lnu.edu.ua*

# Предистория

В ноябре 2012г. ректор ЛНУ имени Ивана Франко получил от Microsoft Ukraine вежливое письмо с предложением составить план приобретения лицензий на продукты Microsoft в 2012-2013 гг. Поскольку в бюджете факультета электроники не было средств на приобретение лицензий проприетарного ПО, то после обсуждения ситуации было принято решение о немедленном переводе всех учебных лабораторий на СПО. Однако, после заявления лектора, который читал курсы “Алгоритмизация и программирование”, “Объектно-ориентированное программирование” с использованием исключительно проприетарного ПО в ОС Microsoft Windows, о “гибели дела его жизни” Ученый Совет факультета выделил 1000 у.е. на приобретение подписки Dream Spark на факультет. Но из-за ошибок в аппликационных формах факультет так и не получил этой подписки.

# Предистория

Впоследствии удалось оформить бесплатную подписку Dream Spark на кафедру радиофизики и КТ. Благодаря этому в трех учебных лабораториях кафедры радиофизики и КТ появились лицензионные версии Microsoft Windows 7, а во всех других лабораториях факультета была установлена ОС Linux. Это привело к тому, что студенты не могли использовать в лабораториях с ОС Linux тех средств разработки, которые они использовали при изучении курсов "Алгоритмизация и программирование" и "Объектно-ориентированное программирование".

В 2017/18 учебном году на факультете открыли новую специальность "Информационные технологии". Для студентов этой специальности в первом семестре читается лекционный курс "Основы программирования"

# Курс “Основы программирования”

Первый семестр

Лекции - 32 часа

Лабораторные работы - 32 часа

Самостоятельная работа студента - 56 часов  
экзамен

Курсы, в которых будут использоваться  
полученные знания: Дискретная математика,  
объектно-ориентированное программирование,  
Численные методы, Микропроцессорная техника

# Лекции. Структурное программирование на языке Си

1. Введение. Алфавит языка Си. Идентификаторы. Ключевые (зарезервированные) слова.
2. Типы данных языка Си. Константы и переменные. Данные целого типа. Действительные типы данных. Символьный тип. «Пустой» тип void. совместимость типов
3. Структура простой программы на языке Си. Директивы препроцессора Си. Комментарии. Операции в языке Си. Выражения. Приоритет операций.
4. Функции ввода-вывода в языке Си. Функция форматного вывода printf (). Функция форматного ввода scanf ().
5. Операторы языка Си. Оператор присваивания. Условные операторы. Операторы цикла в языке Си
6. Работа с файлами в Си. Поточковый ввод-вывод. Открытие и закрытие потока. Стандартные файлы и функции для работы с ними. Работа с файлами на диске.

# Лекции. Структурное программирование на языке Си

7. Структуры и объединения в Си. Описание шаблонов структур. Описание структур-переменных. Доступ к компонентам структуры. Анонимный описание структуры и оператор задания типа typedef. Объединения.
8. Функции пользователя в Си. Функция пользователя и его прототип. Функции с переменным количеством аргументов. Командная строка. Параметры функции main ()
9. Массивы и указатели в языке Си. Одномерные массивы. Многомерные массивы. Указатели. Строки символов.
10. История развития программирования. Программирование в машинных кодах. Программирование на языке ассемблера. Языки программирования высокого уровня. Интерпретаторы и компиляторы языков программирования высокого уровня.

# Лекции. Основы алгоритмизации.

11. Методы записи алгоритмов. Словесное описание алгоритма, псевдокод, блок-схемы, структурные диаграммы Насси-Шнейдермана
12. Алгоритм табуляции функции
13. Решение нелинейного уравнения методом половинного деления
14. Решение нелинейного уравнения методом Ньютона
15. Алгоритм подсчета конечной сумм
16. Метод прямоугольников для вычисления значения определенного интеграла.
17. Алгоритм определения минимального элемента массива и его индекса.
18. Алгоритм определения максимального элемента массива и его индекса.
19. Алгоритмы сортировки массивов. Сортировка методом пузырька. Шейкерная сортировка. Сортировка методом выбора. Сортировка вставками. Быстрая сортировка.
20. Использование `gnuplot` для построения графика функции по ее аналитическому заданию
21. Использование `gnuplot` для построения графика функции по текстовому файлу `u` с результатами табуляции функции

# Выбор оболочки для программирования на языке Си

Для изучения начинающими программирования на языке Си стоит выбрать простую оболочку, работу с которой очень легко освоить. Рассмотрим следующую таблицу:

Turbo C	DOS	устарело
Borland C/C++	DOS, Windows	устарело
Kuzya	OS X, Windows, Linux	развивается
Geany	OS X, Windows, Linux	развивается
Anjuta	Linux	развивается
Atom	OS X, Windows, Linux	развивается
Code::Blocks	OS X, Windows, Linux	развивается
Eclipse	OS X, Windows, Linux	развивается
Visual Studio Code	OS X, Windows, Linux	развивается
QtCreator	OS X, Windows, Linux	развивается



# Выбор оболочки для программирования на языке Си

В первых двух строках таблицы упомянуты Turbo C и Borland C/C++, которые хорошо описаны в учебной литературе, но они рассчитаны на работу только в MS DOS и Windows. Остановим наш выбор на оболочках Kuzya, Geany и Code :: Blocks

# Kuzya

Kuzya — это максимально простая оболочка для обучения. Именно на простоту делался основной акцент. Например, в данной оболочке отсутствует менеджер проектов и вообще понятие «проект». Работа проходит с одним файлом, который содержит текст программы. Сразу после запуска IDE Kuzya, студент, не задумываясь над порядком работы с оболочкой, может начать вводить код программы. После сохранения он может скомпилировать и запустить его. Для удобного и быстрого набора кода в главном меню находятся шаблоны синтаксических конструкций языков C/C++ и Pascal. Нажимая левой клавишей мыши на выбранном шаблоне можно вставить соответствующий текст в место, где расположен курсор. Изменение языка перевода кода автоматически приводит к переводу шаблонов. Также поддерживаются подсветка текста программы, которая позволяет легко ориентироваться в нём. Минимум возможностей данной оболочки (оставлены только полезные или необходимые из них, с точки зрения обучения) позволяют по максимуму сконцентрироваться на изучении языков программирования. Также был создан графический движок, который позволяет отображать графические примитивы, используя библиотеку в стиле Borland C++.

# Kuzya

tab2.cpp - /home/alexchmykhalo/tab2.cpp- кузя

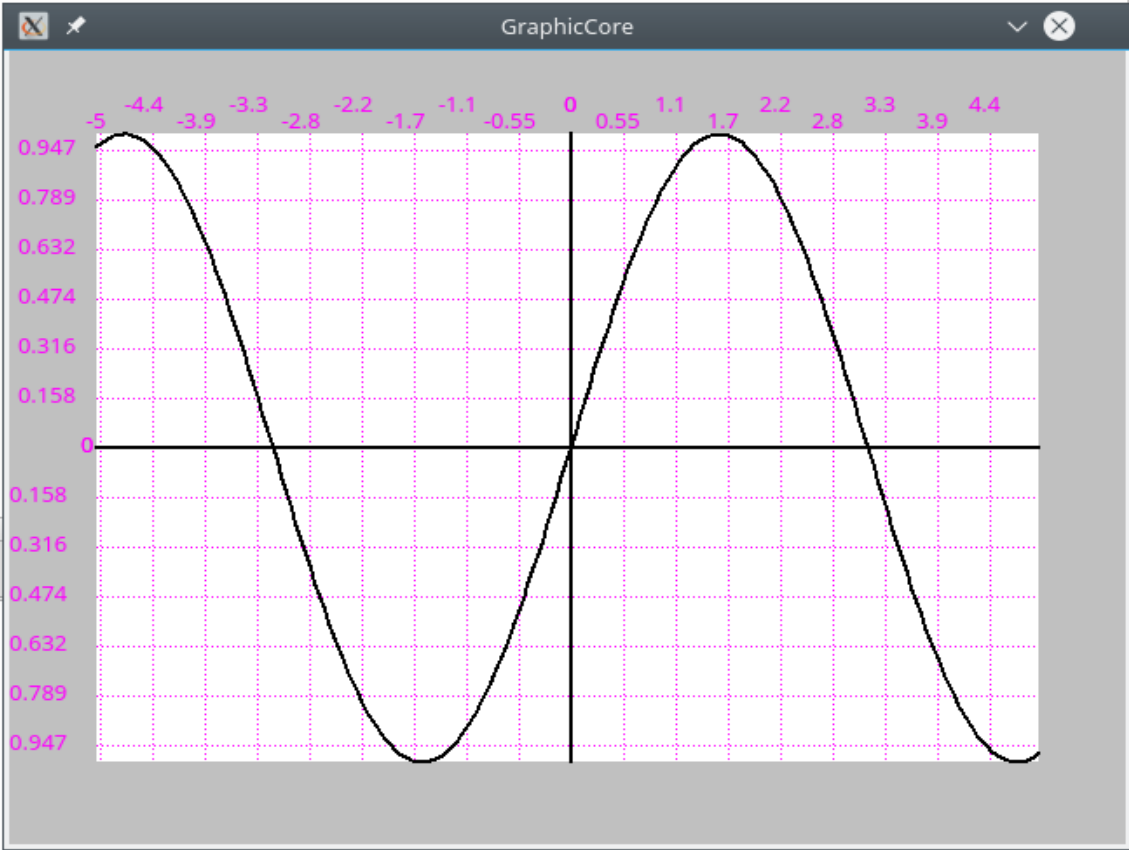
Файл Правка Вигляд Шаблони Налаштування Будувати Debug Допомога

```
#include <graphics.h>
#include <math.h>

int main()
- {
    initgraph();
    const int n = 200;
    double xs[n];
    double ys[n];
    float x = -5.0;
    for (int i = 0; i < n; ++i)
    - {
        xs[i] = x;
        ys[i] = sin(x);
        x += 0.05;
    }
    drawfunc(n, xs, ys);
    closegraph();
    return 0;
}
```

Output Debug

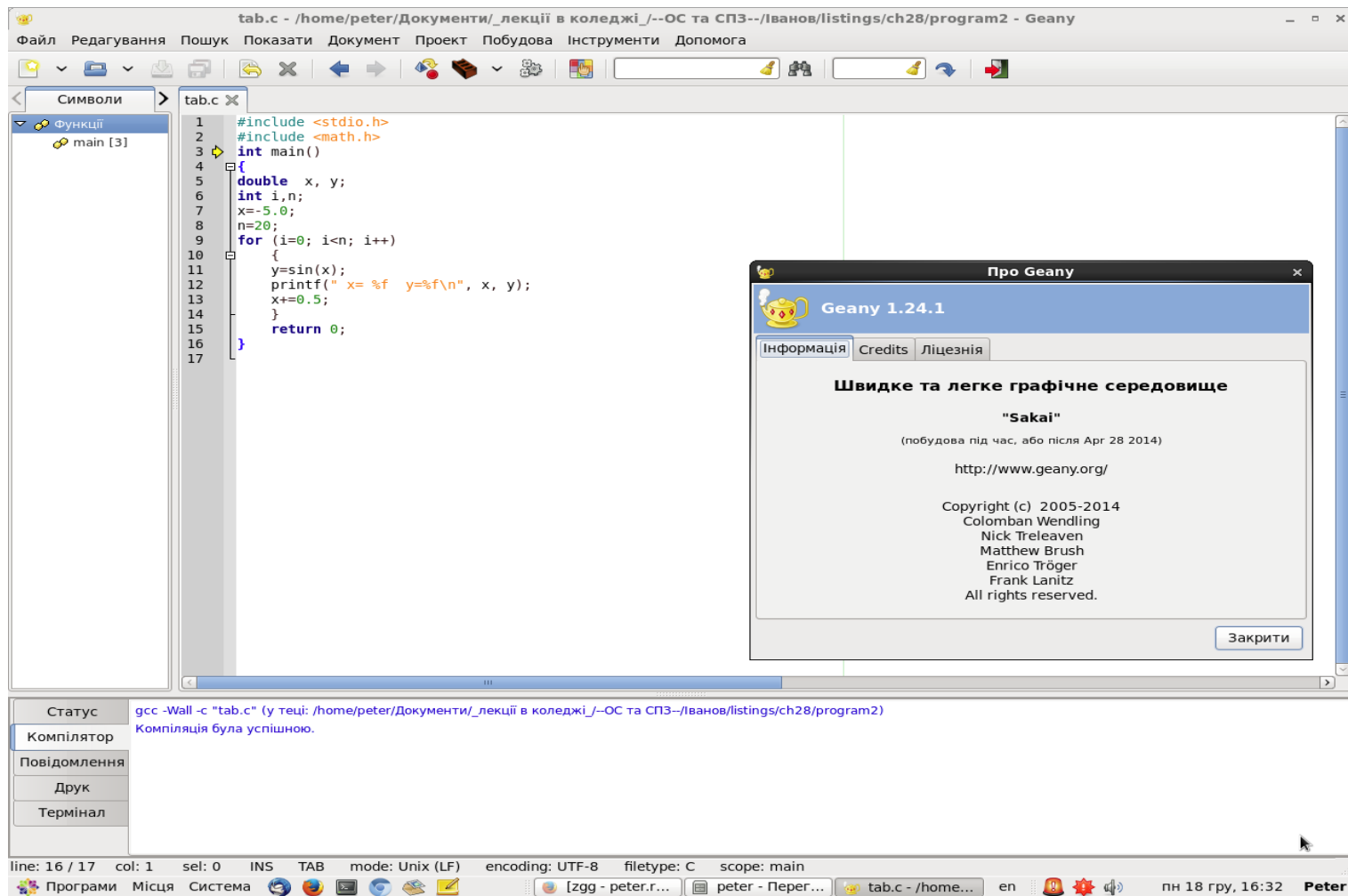
Успішно скомпільовано!



Рядок: 6 Позиція: 16

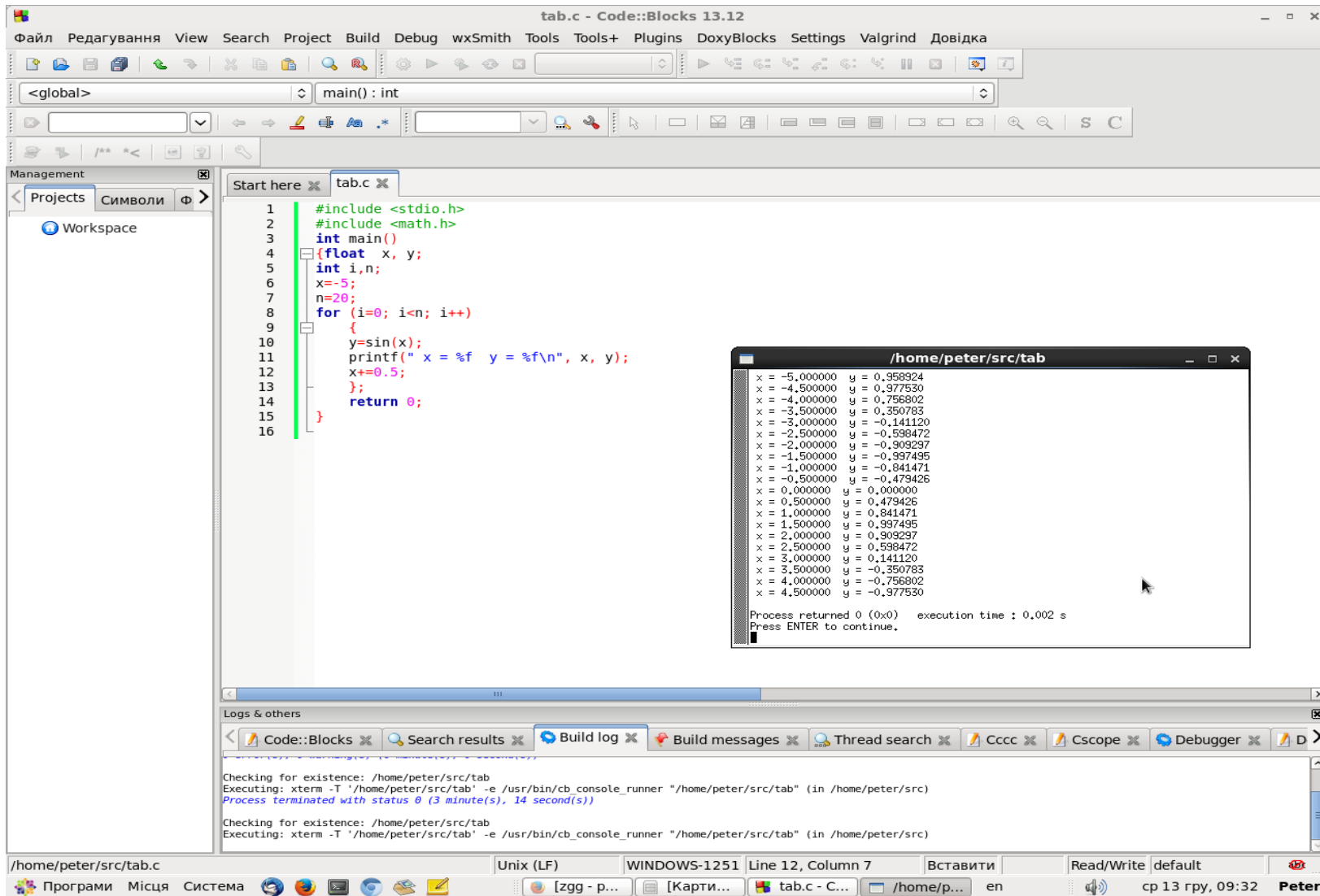
# Geany

Geany — среда разработки программного обеспечения, написанная с использованием библиотеки GTK+. Доступна для следующих операционных систем: BSD, Linux, Mac OS X, Solaris и Windows. Geany распространяется согласно GNU General Public License. Geany не включает в свой состав компилятор. Для создания исполняемого кода используется GNU Compiler Collection или, при необходимости, любой другой компилятор.



# Code::Blocks

Code::Blocks — свободная кроссплатформенная среда разработки. Code::Blocks написана на C++ и использует библиотеку wxWidgets. Имея открытую архитектуру, может масштабироваться за счёт подключаемых модулей. Поддерживает языки программирования C, C++, D (с ограничениями), Fortran. Code::Blocks разрабатывается для Windows, Linux и Mac OS X.



# Выводы

1. Все три оболочки являются кроссплатформенными, что позволяет студентам работать в той ОС, которая установлена на его рабочем месте (стационарной или переносной ЭВМ). Благодаря этому студенты без дополнительных затрат времени могут познакомиться с кроссплатформенным программированием на уровне компиляции.
2. Оболочки Kuzuя и Geany допускают выбор пользователем языка интерфейса (русский, украинский, английский), что уменьшает умственные усилия начинающего при работе с оболочкой.
3. Оболочка Kuzuя имеет графический движок, который позволяет студентам строить графики исследуемых функций. Для построения графиков функций, протабулированных в Geany или Code::Blocks придется использовать внешние средства, например gnuplot.