Программирование в среде RobotC

Введение

3-й и 4-й год обучения

Подготовка

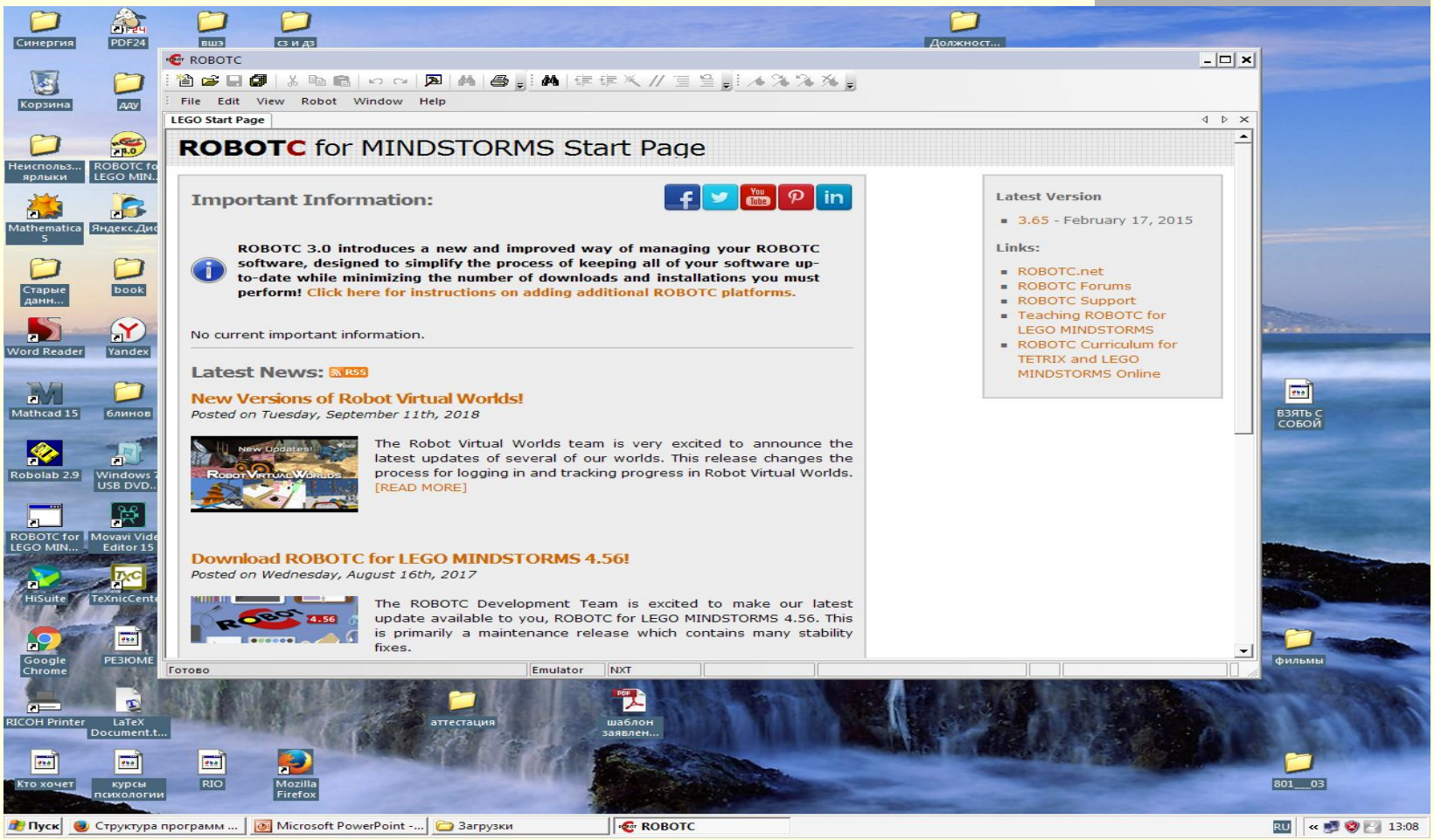
- Установка RobotC на свой компьютер или ноутбук
- Если программировать NXT, то достаточно 3 версии.
- Если программировать EV3, то нужна 4 версия.
- Если дома нет робота, то можно установить любую версию
- На занятиях мы будем ориентироваться на тех, у кого робота нет.

Запуск среды RobotC

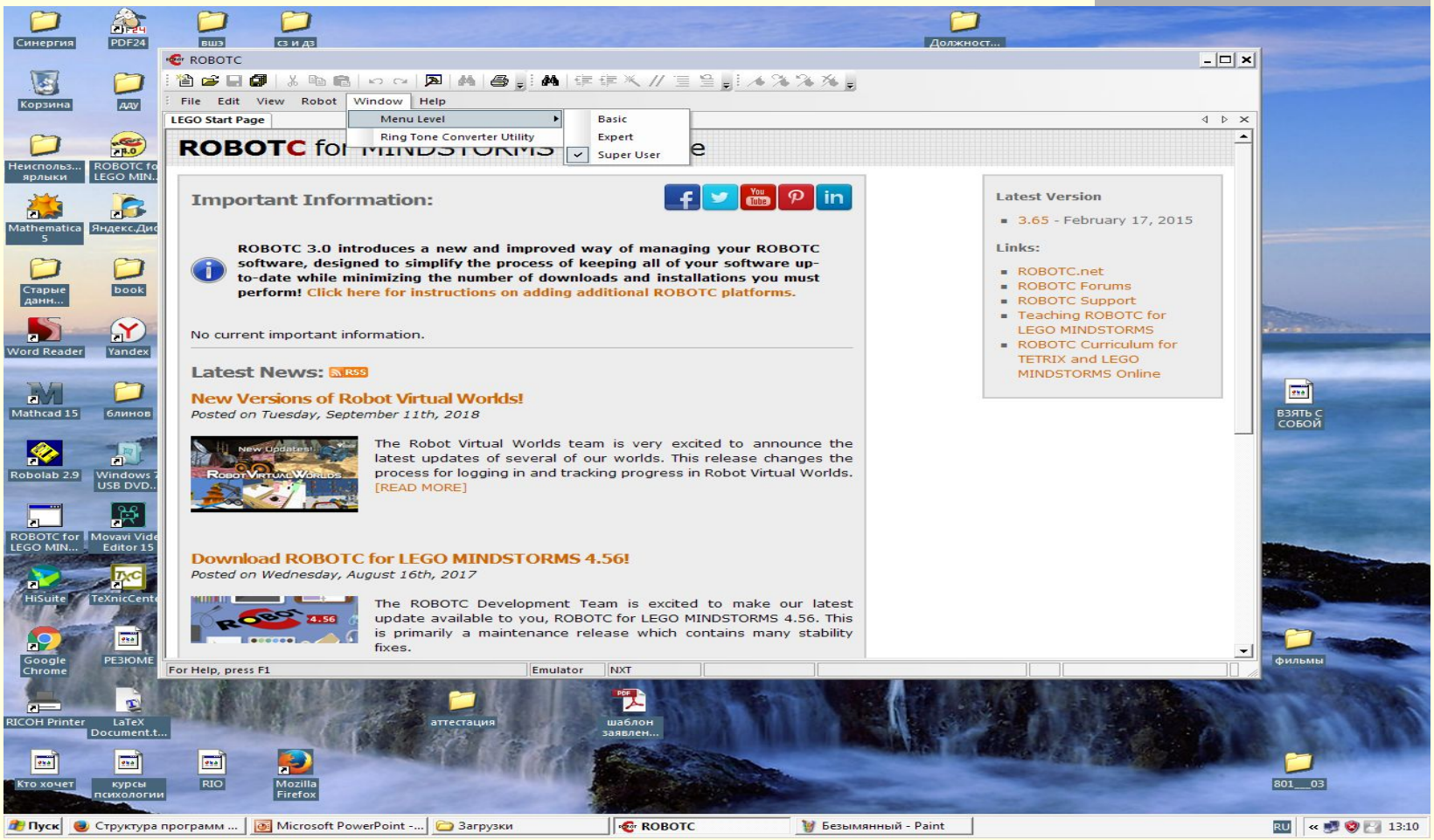
- Нажимаем на пиктограмму RobotC
- Открывается стартовая страница с меню
- Выбираем в меню Window – Menu Level – Super User
- Выбираем в меню Robot – Compiler Target – PC-based Emulator
- Выбираем в меню File – New – New File

На следующих слайдах продемонстрированы все указанные действия и результат

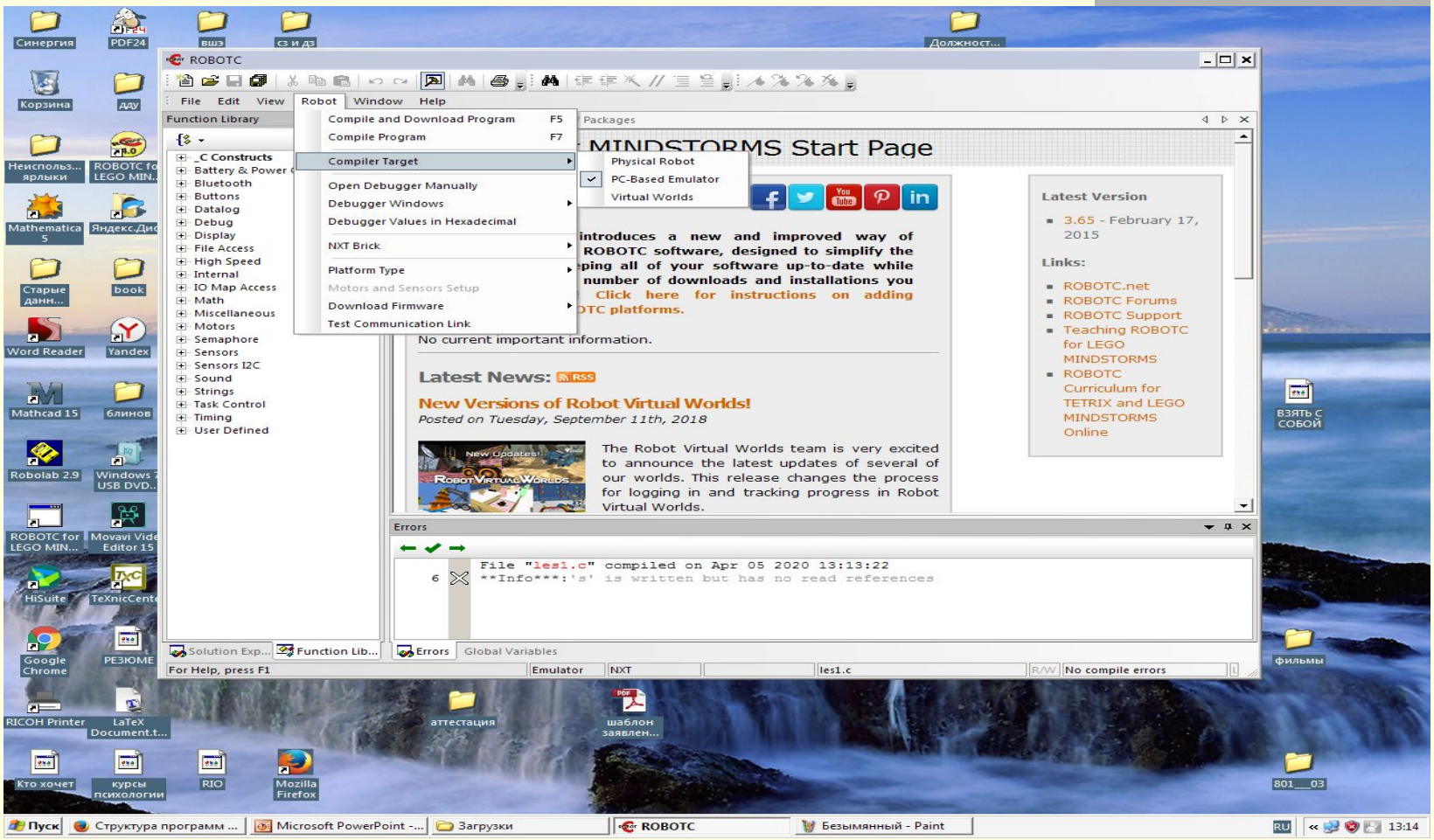
Запуск среды RobotC



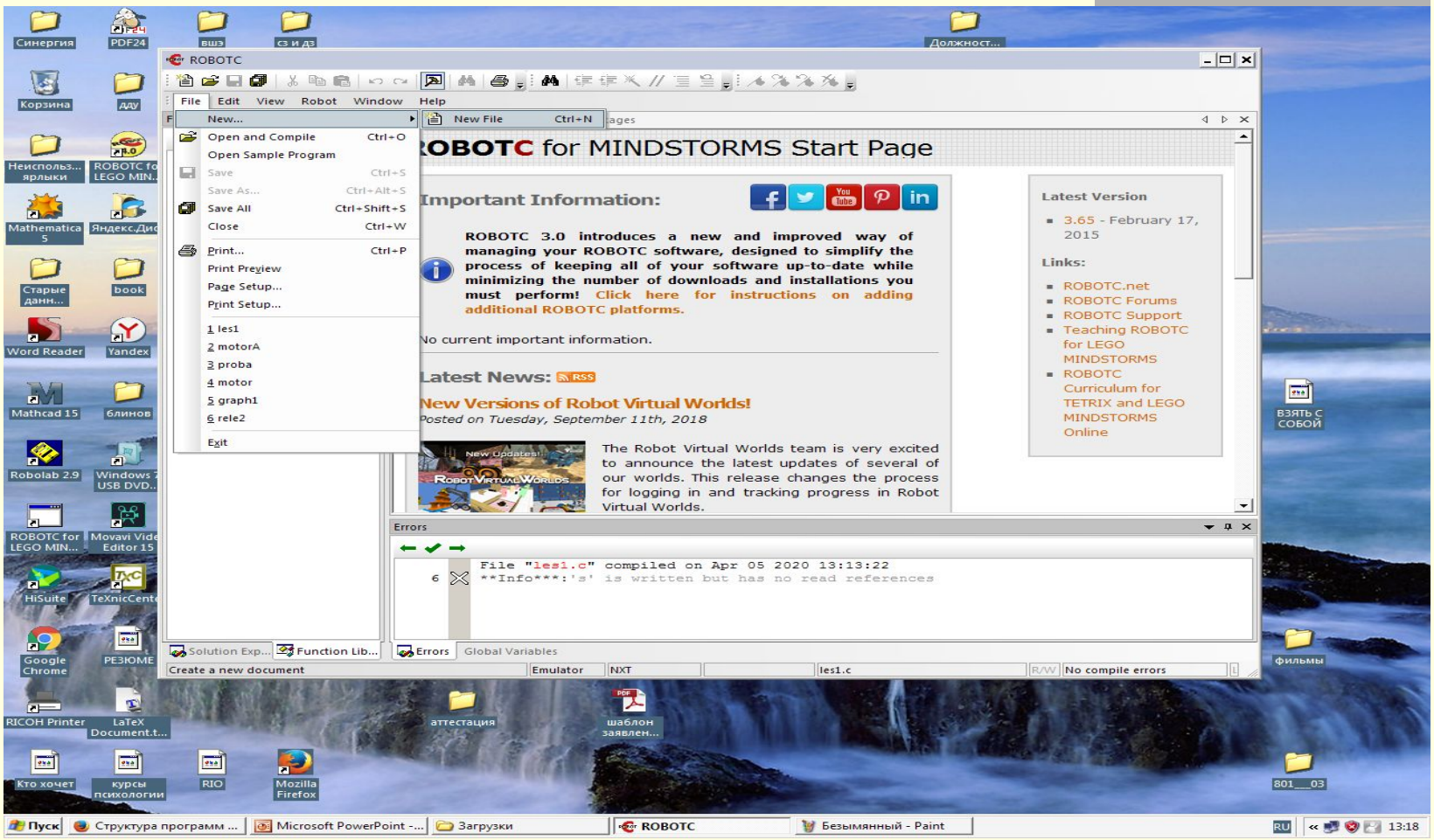
Запуск среды RobotC



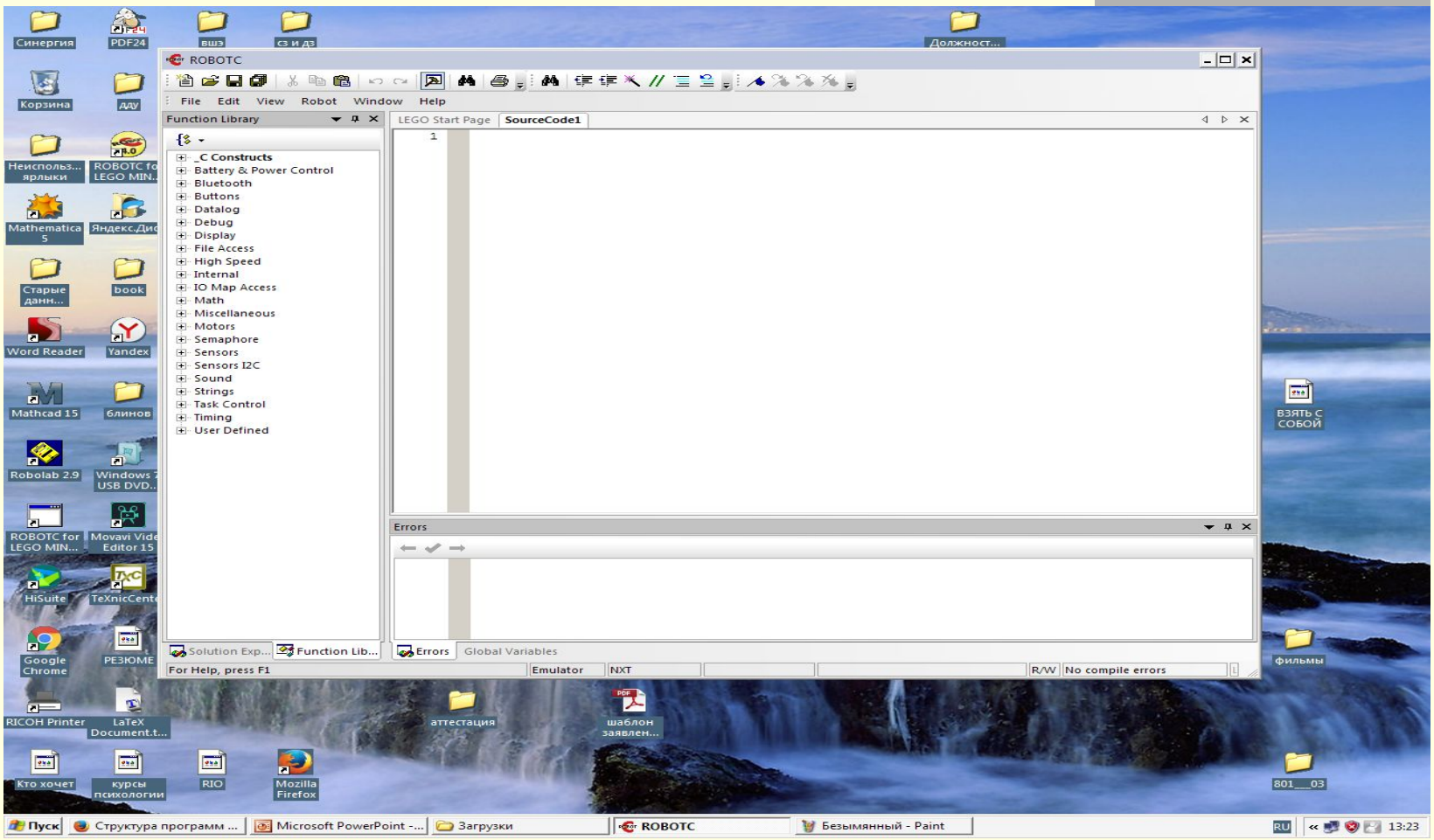
Запуск среды RobotC



Запуск среды RobotC



Запуск среды RobotC



Программирование

Программирование – это процесс написания компьютер-ных программ. Примерно вот так компьютерные программы выглядят изначально:

C:\Users\KaDeaT\Desktop\Work\firefox\firefox-31.2.0esr.source\mozilla-esr31\security\nss\lib\libpkix\pkix_pl_nss\system\pkix_pl_pri...

Файл Правка Поиск Вид Кодировки Синтаксисы Опции Макросы Запуск Плагины Окна ?

pkix_pl_primhash.c

```
273         void **pValue,
274         void *plContext)
275 {
276         pkix_pl_HT_Elem *element = NULL;
277         pkix_pl_HT_Elem *prior = NULL;
278         PKIX_Boolean compResult;
279
280         PKIX_ENTER(HASHTABLE, "pkix_pl_PrimHashTable_Remove");
281         PKIX_NULLCHECK_FOUR(ht, key, pKey, pValue);
282
283         *pKey = NULL;
284         *pValue = NULL;
285
286         for (element = ht->buckets[hashCode%ht->size], prior = element;
287             (element != NULL);
288             prior = element, element = element->next) {
289
290                 if (element->hashCode != hashCode){
291                         /* no possibility of a match */
```

Программирование

Если вы обратите внимание на картинку, то заметите, что программа - это просто некоторый текст. Возможно, сейчас он кажется вам непонятной абракадаброй, но на самом деле в нем есть свои четкие правила, которые никто не может нарушить. Эти правила и составляют основу любого языка программирования. В принципе, записать текст программы можно в любом текстовом редакторе, даже в стандартном блокноте ОС Windows, но есть одно значительное но.

Как вы, наверное, знаете, компьютер не понимает ничего, кроме нулей и единиц. Уж тем более он не понимает никаких слов, поэтому написанный нами текст программы для него, примерно как и для вас сейчас – набор непонятных символов. Чтобы компьютер понял то, что мы от него хотим, нам нужна специальная программа, которая, если так можно выразиться, переведет программу, записанную на языке программирования, на язык, понятный компьютеру. Такая программа называется компилятором, а сам процесс перевода программы на понятный компьютеру язык - компиляцией.

Программирование

Итак, процесс написания программ состоит в самом простом виде из следующих шагов:

- *написать код программы в любом текстовом редакторе*
- *скомпилировать исходный код*

Но неудобно писать код в блокноте, потом компилировать его специальной программой, особенно если ваша программа состоит из большого количества файлов с исходным кодом. В связи с этим программисты создали для себя такие программы, в которых можно сразу писать код, проверять его на ошибки, компилировать и запускать готовую программу. В общем, все сразу и в одном флаконе. Такие программы называются интегрированными средами разработки (IDE). Таким образом, любая IDE сочетает в себе по крайне мере текстовый редактор и компилятор. Здесь мы подходим к нашей среде RobotC.

Программирование

Прежде, чем написать первую программу мы поговорим о структуре самой программы.

Все программы на RobotC (и на C) выглядят одинаковым образом.

Во всех программах присутствует приведенная на следующем слайде конструкция.

Структура программы

task main ()	*Обязательный заголовок*
{	*Начало программы*
	Тело программы
}	*Конец программы*

Структура программы

Что же это за конструкция? Это объявление функции main. Такая функция обязательно есть в каждой программе, которая написана на языке Си. Большая программа или маленькая, компьютерная игра или программа "Hello, World!", написана вами или Биллом Гейтсом -- если программа написана на языке Си -- в ней есть функция main. Это так сказать главная функция нашей программы. Когда мы запускаем программу, то можно думать, что запускаем функцию main этой программы.

Остановимся на секундочку. Мы, кажется, уже кое-что выяснили о структуре программ на языке Си. Любая программа на языке Си должна содержать функцию main.

Функция main

Давайте немного поговорим о функции main и о функциях вообще.

После имени функции записаны скобки. Вообще, в скобках обычно записывают аргументы функции, но в нашем случае, когда в скобках ничего нет, это означает, что аргументов у функции нет. Другими словами, чтобы функция main начала работу ей не нужны никакие дополнительные данные извне.

Внутри фигурных скобок идёт описание функции main, т.е. непосредственно то, что эта функция должна делать.

То что записано в фигурных скобках обычно называют "тело функции" (или описание функции), а первую часть, та что перед фигурными скобками называется заголовок функции.

Функция main

Перед закрывающей фигурной скобкой иногда ставят команду return. Именно эта команда отвечает за то, чтобы вернуть значение из функции. Т.е. смотрит, если программа дошла до этого места, то значит всё было хорошо и никаких ошибок не возникло, а значит можно вернуть значение нуль.

Вы можете спросить, а почему именно нуль? Просто так обычно делают. Можно, в принципе, возвращать какое-нибудь другое целое число, например 100, или -236. Вот мы и разобрались с функцией main.

Завершение вводной части

Теперь у нас всё готово, чтобы написать первую программу на RobotC. Поскольку дома робот может оказаться не у всех, мы будем создавать программы, которые не будут загружаться на котроллер. Но программные конструкции, которые мы будем рассматривать, нам будут необходимы не только для обучения программированию в целом, но и для отладки программ наших творческих проектов в дальнейшем.