**Задание 6. Облако слов.**

**Оригинал.**

Developing software for modern GUI-based operating systems differs in many ways from the earlier text-based systems. Until the mid 1980s attention focused on the quality of the programming language itself. Languages such as FORTRAN, Pascal or C might be for reasons of elegance, flexibility or efficiency. Less attention was paid to the input/output functions of how data was entered and displayed by the program. With GUIs the situation is the exact opposite, the key issues in a programming language being how well it supports the user interface. The creation and manipulation of multiple windows on the display screen requires the tracking of a large amount of information within the program, with the support provided for these tasks determining how quickly and easily a program can be written. The programming language is only one component of a broader software development system which includes tools for creating the user interface and testing the programs.

In producing laboratory software, the choice of software development system is crucial. The first decision is between using a general purpose system, designed to produce any kind of program, and specialist systems designed for particular types of application, data acquisition and analysis in our case. There are three main general purpose software development systems for Microsoft Windows in common use:

•Microsoft Visual Basic

•Borland Delphi

•Microsoft Visual C++

Of the systems designed for specialist data acquisition and analysis applications, the two most notable are:

•National Instruments LabVIEW

•Mathworks Matlab

Each of these systems has its own strengths and weaknesses, but from our point of view – developing software for physiological data acquisition and analysis – three issues are of prime importance:

•Ease of use

•Signal analysis/display support

•Data acquisition hardware support

Software development within a laboratory is usually done in response to a need that has arisen in the experimental work. A new approach to signal analysis may be required or new types of data acquisition hardware may need to be integrated into the experimental apparatus. Usually, there is a desire to develop the software quickly in order to aid the progress of the research. At the same time, the software development skills of the typical researcher, at least initially, may be fairly limited. The ease with which a development system can be learned is thus significant. Some systems have much steeper learning curves than others, with months of effort required before a useful output is obtained. Equally, the extent to which the system supports the required tasks must be considered. Typically, there is a need for the analysis and graphical display of physiological signals or images in a variety of forms. Specialist packages are likely to have built-in support for this, which is lacking from the general purpose systems. Finally, if the software is to acquire experimental signals or images, the system must be able to make use of the data acquisition hardware. These criteria should be borne in mind during the detailed discussion of each development system that follows.

*Незнакомые слова*:

data acquisition – сбор данных;

elegancy - ясность,

broad - широкий ,

arise - возникать,

significant -важный,

steep – обрыв, крутизна

acquire - обзаводиться,

crucial - ключевой,

strength - сила,

weakness - слабость,

obtain - получать,

effort – попытка, усилие

**Перевод Google.**

Разработка программного обеспечения для современных операционных систем с графическим интерфейсом пользователя во многом отличается от более ранних текстовых систем. До середины 1980-х внимание было сосредоточено на качестве самого языка программирования. Такие языки, как FORTRAN, Pascal или C, могут быть предпочтительнее по причинам элегантности, гибкости или эффективности. Меньше внимания уделялось функциям ввода / вывода того, как данные вводятся и отображаются программой. С графическим интерфейсом пользователя ситуация прямо противоположная, ключевой проблемой языка программирования является то, насколько хорошо он поддерживает пользовательский интерфейс. Создание и управление несколькими окнами на экране дисплея требует отслеживания большого количества информации в программе, при этом поддержка, предоставляемая для этих задач, определяет, насколько быстро и легко программа может быть написана. Язык программирования - это только один компонент более широкой системы разработки программного обеспечения, которая включает инструменты для создания пользовательского интерфейса и тестирования программ.

При производстве лабораторного программного обеспечения решающее значение имеет выбор системы разработки программного обеспечения. Первое решение - между использованием системы общего назначения, предназначенной для создания любого вида программ, и специализированных систем, разработанных для конкретных типов приложений, сбора и анализа данных в нашем случае. Обычно используются три основных системы разработки программного обеспечения общего назначения для Microsoft Windows:

• Microsoft Visual Basic

• Borland Delphi

• Microsoft Visual C ++

Из систем, разработанных для специализированных приложений сбора и анализа данных, наиболее заметными являются следующие:

• National Instruments LabVIEW

• Mathworks Matlab

Каждая из этих систем имеет свои сильные и слабые стороны, но с нашей точки зрения - разработка программного обеспечения для сбора и анализа физиологических данных - три вопроса имеют первостепенное значение:

• Простота использования.

• Анализ сигналов / поддержка отображения

• Аппаратная поддержка сбора данных

Разработка программного обеспечения в лаборатории обычно выполняется в ответ на потребность, возникшую в ходе экспериментальной работы. Может потребоваться новый подход к анализу сигналов или могут потребоваться новые типы оборудования для сбора данных, которые необходимо будет интегрировать в экспериментальную установку. Обычно есть желание быстро разработать программное обеспечение, чтобы способствовать прогрессу исследования. В то же время навыки разработки программного обеспечения типичного исследователя, по крайней мере на начальном этапе, могут быть довольно ограниченными. Таким образом, очень важна легкость, с которой можно изучить систему разработки. Некоторые системы имеют гораздо более крутые кривые обучения, чем другие, и для получения полезного результата требуются месяцы усилий. Точно так же необходимо учитывать степень, в которой система поддерживает требуемые задачи. Обычно существует потребность в анализе и графическом отображении физиологических сигналов или изображений в различных формах. Специализированные пакеты, вероятно, будут иметь встроенную поддержку для этого, чего нет в системах общего назначения. Наконец, если программное обеспечение должно получать экспериментальные сигналы или изображения, система должна иметь возможность использовать аппаратное обеспечение для сбора данных. Эти критерии следует учитывать при подробном обсуждении каждой последующей системы разработки.

**Перевод Яндекс.**

Разработка программного обеспечения для современных операционных систем с графическим интерфейсом во многом отличается от более ранних текстовых систем. До середины 1980-х годов внимание было сосредоточено на качестве самого языка программирования. Такие языки, как FORTRAN, Pascal или C, могут быть предпочтительны по соображениям элегантности, гибкости или эффективности. Меньше внимания уделялось функциям ввода/вывода того, как данные вводились и отображались программой. С графическими интерфейсами ситуация прямо противоположная, ключевые проблемы языка программирования заключаются в том, насколько хорошо он поддерживает пользовательский интерфейс. Создание и управление несколькими окнами на экране дисплея требует отслеживания большого объема информации в программе, при этом поддержка, предоставляемая для этих задач, определяет, насколько быстро и легко может быть написана программа. Язык программирования является лишь одним компонентом более широкой системы разработки программного обеспечения, которая включает инструменты для создания пользовательского интерфейса и тестирования программ.

При создании лабораторного программного обеспечения решающее значение имеет выбор системы разработки программного обеспечения. Первое решение заключается в использовании системы общего назначения, предназначенной для создания любых программ, и специализированных систем, предназначенных для конкретных типов приложений, сбора и анализа данных в нашем случае. В общем использовании существует три основные системы разработки программного обеспечения общего назначения для Microsoft Windows:

• Microsoft Visual Basic (Microsoft Visual Basic)

• Борланд Дельфи

• Microsoft Visual C++ (Microsoft Visual C++)

Из систем, предназначенных для специализированных приложений сбора и анализа данных, наиболее заметными являются две:

• Лабораторный обзор National Instruments

• Mathworks Matlab (Математические работы)

Каждая из этих систем имеет свои сильные и слабые стороны, но с нашей точки зрения – разработка программного обеспечения для сбора и анализа физиологических данных – три вопроса имеют первостепенное значение:

• Простота использования

• Поддержка анализа сигналов/отображения

• Аппаратная поддержка сбора данных

Разработка программного обеспечения в лаборатории обычно осуществляется в ответ на потребность, возникшую в ходе экспериментальной работы. Может потребоваться новый подход к анализу сигналов или может потребоваться интеграция в экспериментальную аппаратуру новых типов оборудования для сбора данных. Обычно возникает желание быстро разработать программное обеспечение, чтобы способствовать прогрессу исследований. В то же время навыки разработки программного обеспечения типичного исследователя, по крайней мере на начальном этапе, могут быть довольно ограниченными. Таким образом, легкость, с которой можно изучить систему разработки, имеет большое значение. Некоторые системы имеют гораздо более крутые кривые обучения, чем другие, и для получения полезного результата требуются месяцы усилий. В равной степени необходимо учитывать степень, в которой система поддерживает требуемые задачи. Как правило, возникает необходимость в анализе и графическом отображении физиологических сигналов или изображений в различных формах. Специализированные пакеты, вероятно, будут иметь встроенную поддержку для этого, которой не хватает в системах общего назначения. Наконец, если программное обеспечение должно получать экспериментальные сигналы или изображения, система должна иметь возможность использовать аппаратное обеспечение для сбора данных. Эти критерии следует учитывать при подробном обсуждении каждой последующей системы разработки.