

И. В. Заика,

Таганрогский институт имени А. П. Чехова (филиал) Ростовского государственного экономического университета (РИНХ)

ПРАКТИЧЕСКИЕ РАБОТЫ ПО СОЗДАНИЮ ПРИКЛАДНЫХ ПРОГРАММ В VISUAL C++*

Аннотация

В статье представлены практические работы по созданию прикладных программ в среде Visual C++, в частности, рассматриваются примеры создания калькулятора с помощью Visual C++. Практические работы могут быть использованы при изучении объектно-ориентированного программирования в школьном курсе информатики.

Ключевые слова: программирование, Visual C++, создание прикладных программ, калькулятор.

DOI: 10.32517/2221-1993-2018-17-7-26-31

Контактная информация

Заика Ирина Викторовна, канд. тех. наук, доцент, доцент кафедры информатики Таганрогского института имени А. П. Чехова (филиала) Ростовского государственного экономического университета (РИНХ); *адрес:* 347936, Ростовская область, г. Таганрог, ул. Инициативная, д. 48; *телефон:* (8634) 60-18-07; *e-mail:* irin-zaik@yandex.ru

I. V. Zaika,

Taganrog Institute named after A. P. Chekhov (branch) of Rostov State University of Economics

PRACTICAL WORKS FOR CREATING APPLICATION PROGRAMS IN VISUAL C++

Abstract

The article presents practical works on creating application programs in the Visual C++ environment. In particular, examples of how to create a calculator using the Visual C++ environment are considered. Practical works can be used in the study of object-oriented programming in the school course of informatics.

Keywords: programming, Visual C++, creating application programs, calculator.

С помощью методов и средств объектно-ориентированного программирования можно осуществлять моделирование различных объектов и процессов [1, 3, 4], интересных для школьников. Разработка визуальных прикладных программ позволяет повысить мотивацию, а также познавательную активность учащихся.

Ниже представлены практические работы для пошагового создания приложения «Калькулятор» средствами Visual C++, которые могут быть использованы при изучении объектно-ориентированного программирования в школьном курсе информатики [2, 5]. Все программы создаются в Microsoft Visual C++ 2008 Express Edition.

Практическая работа 1 «Разработка программы простых вычислений средствами Visual C++»

Задание.

Создать программу для:

- перевода десятичного целого числа в двоичную систему счисления;
- вычисления факториала;
- вычисления значения тригонометрической функции $\sin(x)$.

При запуске программы должно появляться окно, где при нажатии на форме одной из трех кнопок реализуются соответствующие вычисления (рис. 1).

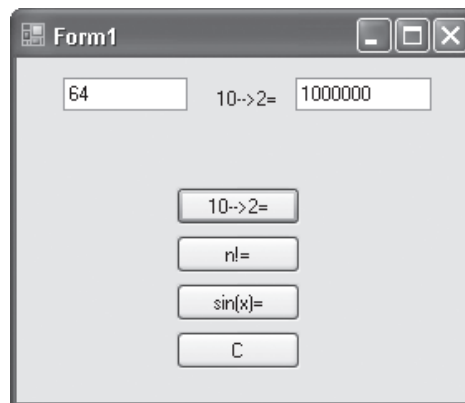


Рис. 1

Решение.

1. Запустить Microsoft Visual C++ 2008 Express Edition.
2. Создать новый проект. Для создания нового проекта необходимо:
 - выбрать в меню *File* пункт *New Projects*;
 - в открывшемся окне на закладке *Projects types* выбрать тип проекта *CLR*;
 - на закладке *Templates* выбрать *Windows Forms Application*;
 - в строке *Name* ввести имя, например *kalkulyat*;
 - щелкнуть мышью на кнопке *OK*.
3. Откроется форма *Form1*.
4. В меню *View* открыть панель инструментов *Toolbox*, во вкладке *Alle Windows Forms* выбрать элемент *TextBox* (*Текстовое поле*) и перетянуть *textBox1* на форму *Form1*, затем разместить *textBox2* таким об-

* Материалы к статье можно скачать на сайте ИНФО:
http://infojournal.ru/journals/school/school_7-2018/

разом, чтобы *textBox1* и *textBox2* находились на одной линии (см. рис. 1). Обратите внимание: название объектов (компонент), перетянутых на форму, отличается от названия элементов, указанных на панели инструментов *Toolbox*. Например, на панели инструментов *Toolbox* элемент *TextBox* начинается с прописной буквы, а после размещения на форму *Form1* объекта *textBox1* название будет начинаться со строчной буквы.

5. На панели инструментов *Toolbox* во вкладке *Alle Windows Forms* выбрать элемент *Label (Метка)* и разместить *label1* на форме.

Чтобы обращаться к элементам из *Toolbox*, каждому объекту (например, *textBox1*, *textBox2* или *label1* и т. д.) присваивается уникальное имя или номер. Каждый

объект можно охарактеризовать присущим именно ему набором свойств (*Properties*), а также объект может реагировать на события (*Events*).

6. Для *label1* нужно назначить текст «=».

Для этого необходимо в меню *View* выбрать *Other Windows, Properties Window*. Откроется окно *Properties* для объекта *label1*. В окне *Properties* с помощью меню *Properties* и *Events* можно соответственно менять свойства объектов и назначать события для объектов. Выделить объект *label1* и в окне *Properties* в меню *Properties* назначить текст «=» в свойстве *Text*.

Для реализации программы *kalkulyat* в таблице 1 приводится пошаговая инструкция по написанию кода программы.

Таблица 1

Выделенный объект	Окно <i>Properties</i>	Свойство или событие	Описание действий для выделенного объекта
Перевод десятичного целого числа в двоичную систему счисления			
<i>label1</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: =
<i>button1</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: 10->2=
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>label1->Text=(button^sender)->Text; long a,d, ostatok, razryad, dvoichnoe_chislo; a=Convert.ToInt32(textBox1->Text); d=a; ostatok=0; razryad=1; dvoichnoe_chislo=0; while(d!=0){ostatok=a%2; dvoichnoe_chislo=dvoichnoe_chislo+ostatok*razryad; razryad=razryad*10; d=a/2; a=d;} textBox2->Text=dvoichnoe_chislo.ToString();</code>
Вычисление факториала n!			
<i>button2</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: n!=
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>label1->Text=(button^sender)->Text; int k=1; for(int i=1; i<=Convert.ToInt32(textBox1->Text); i++) k=k*i; textBox2->Text=k.ToString();</code>
Вычисление значения тригонометрической функции sin(x) (для работы с математическими функциями необходимо подключение библиотеки <Math.h>)			
<i>button3</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: sin(x)=
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>label1->Text=(button^sender)->Text; float k; k=sin(Convert.ToDouble(textBox1->Text)); textBox2->Text=k.ToString();</code>
<i>button4</i> (кнопка сброса — при нажатии на кнопку <i>button4</i> будет очищен текст в <i>textBox1</i> , <i>textBox2</i> , <i>label1</i>)	<i>Properties</i>	<i>Text</i>	Ввести текст: С
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>textBox1->Text=""; textBox2->Text=""; label1->Text="";</code>

7. Сохранить программу *kalkulyat*. Для этого в меню *File* выбрать *Save All*.

8. Запустить программу *kalkulyat*. Для этого в меню *Debug* выбрать *Start Debugging*.

Ниже представлен фрагмент листинга программы *kalkulyat*.

```
private: System::Void button1_Click
(System::Object^sender, System::EventArgs^e)
{label1->Text=((button^sender)->Text;
long a,d,ostatok,razryad,dvoichnoe_chislo;
a=Convert::ToInt32(textBox1->Text); d=a;
ostatok=0;
razryad=1;
dvoichnoe_chislo=0;
while(d!=0) {ostatok=a % 2; dvoichnoe_chislo=
dvoichnoe_chislo+ostatok*razryad;
razryad=razryad*10;
d=a/2;
a=d;}
textBox2->Text=dvoichnoe_chislo.ToString();}

private: System::Void button2_Click
(System::Object^sender, System::EventArgs^e)
{label1->Text=((button^sender)->Text;
int k=1;
for(int i=1; i<=Convert::ToInt32(textBox1
->Text); i++) k=k*i;
textBox2->Text=k.ToString();}

private: System::Void button3_Click
(System::Object^sender, System::EventArgs^e)
{label1->Text=((button^sender)->Text;
float k;
k=sin(Convert::ToDouble(textBox1->Text));
textBox2->Text=k.ToString();}

private: System::Void button4_Click
(System::Object^sender, System::EventArgs^e)
{textBox1->Text="";
textBox2->Text="";
label1->Text="";}
```

Полный листинг программы приведен в *Приложении 1*, которое можно скачать на сайте ИНФО.

Далее рассматривается возможность создания кнопочного калькулятора с помощью Visual C++. При разработке калькулятора необходимо учесть все нюансы при работе с целыми и вещественными типами данных в C++ [6–8]. В частности, при реализации частного от деления двух чисел все данные должны быть вещественного типа, в противном случае результат от деления будет выводиться только как целое число. Поэтому необходимо уметь оперировать не только целыми, но и вещественными типами данных [10].

Практическая работа 2 «Создание приложения Калькулятор средствами Visual C++»

Задание.

Создать форму по образцу (рис. 2).

Решение.

1. Запустить Microsoft Visual C++ 2008 Express Edition.
2. Создать новый проект. Для создания нового проекта необходимо:

- выбрать в меню *File* пункт *New Projects*;
- в открывшемся окне на закладке *Projects types* выбрать тип проекта *CLR*;

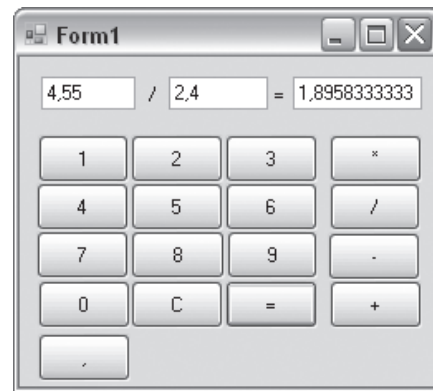


Рис. 2. Калькулятор

- на закладке *Templates* выбрать *Windows Forms Application*;
- в строке *Name* ввести имя, например *kalkulyator*;
- щелкнуть мышью на кнопке *OK*.

3. Откроется форма *Form1*.

4. В меню *View* открыть панель инструментов *Toolbox*, во вкладке *Alle Windows Forms* выбрать элемент *TextBox* (Текстовое поле) и разместить *textBox1* на форме *Form1*, затем на форму разместить *textBox2* и *textBox3* таким образом, чтобы они находились на одной линии (см. рис. 2).

5. Выбрать на панели инструментов *Toolbox* элемент *Label* (Метка) и перетащить *label1* на форму, чтобы метка была расположена между объектами *textBox1* и *textBox2*. Вторую метку *label2* расположить между *textBox2* и *textBox3* (образец представлен на рисунке 2).

В меню *View* выбрать *Other Windows, Properties Window*. Откроется окно *Properties*. Выделить объект *label2* и в окне *Properties* в меню *Properties* назначить текст «=» в свойстве *Text*. Выделить *label1* и в окне *Properties* в меню *Properties* очистить текст в свойстве *Text*.

6. Выбрать в *Toolbox* элемент *Button* (Кнопка) и перетащить *button1* на форму. Скопировать 16 раз кнопку *Button* на форму *Form1*.

7. Переименовать кнопки — соответственно на 0, 1, 2 и т. д.

Для реализации программы *kalkulyator* в таблице 2 приводится пошаговая инструкция по написанию кода программы.

8. Сохранить программу *kalkulyator*. Для этого в меню *File* выбрать *Save All*.

9. Запустить программу *kalkulyator*. Для этого в меню *Debug* выбрать *Start Debugging*.

Полный листинг программы приведен в *Приложении 2*, которое можно скачать на сайте ИНФО.

Для самостоятельной работы можно предложить дополнительное задание, которое позволит закрепить изученный материал.

Задание.

Добавить кнопку для возведения числа в степень: N^m .

Решение.

В C++ для возведения N в степень m используется функция *pow*(N , m).

Для обращения к функции возведения в степень необходимо в самом начале программы подключить библиотеку функций `<math.h>`.

1. Запустить Microsoft Visual C++ 2008 Express Edition.

Таблица 2

Выделенный объект	Окно <i>Properties</i>	Свойство или событие	Описание действий для выделенного объекта
Чтобы после ввода цифр в <i>textBox1</i> осуществлялся ввод во вторую компоненту — <i>textBox2</i> , необходимо перед областью описания класса в редакторе кода формы ввести логическую переменную <code>bool Z = false</code>			
<i>label1</i>	<i>Properties</i>	<i>Text</i>	Очистить текст
<i>label2</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: =
<i>button1</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: 1
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>if (Z==false)textBox1->Text+= ((button^).sender)->Text; else textBox2->Text+= ((button^).sender)->Text;</code>
Переименовать кнопки <i>button2</i> , <i>button3</i> , ..., <i>button11</i> соответственно в 2, 3, ..., 0, «,»			
<i>button2</i> , ... <i>button11</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: 2, ..., 0, «,»
	Для каждой кнопки <i>button2</i> , <i>button3</i> , ..., <i>button11</i> назначить событие кнопки <i>button1</i>		
	<i>Events</i>	<i>Click</i>	Чтобы назначить событие кнопки <i>button1</i> на другие кнопки, нужно в меню <i>Events</i> напротив вкладки <i>Click</i> для кнопок <i>button2</i> , <i>button3</i> , ..., <i>button11</i> выбрать из списка код: <code>button1_Click</code>
<i>button12</i> (кнопка для умножения)	<i>Properties</i>	<i>Text</i>	Ввести текст: *
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>label1->Text=((button^).sender)->Text; Z=true;</code>
<i>button13</i> , <i>button16</i> , <i>button17</i> (кнопки для сложения, вычитания, деления соответственно)	<i>Properties</i>	<i>Text</i>	Соответственно ввести текст: +, -, /
	<i>Events</i>	<i>Click</i>	Нужно назначить событие кнопки <i>button12</i> на другие кнопки с арифметическими операциями. Для этого в меню <i>Events</i> напротив вкладки <i>Click</i> для кнопок <i>button13</i> , <i>button16</i> , <i>button17</i> выбрать из списка код: <code>button12_Click</code>
<i>button14</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: =
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>double k; switch (label1->Text[0]) {case '*': k=Convert::ToDouble(textBox1->Text)* Convert::ToDouble(textBox2->Text); break; case '/': k=Convert::ToDouble(textBox1->Text)/ Convert::ToDouble(textBox2->Text); break; case '+': k=Convert::ToDouble(textBox1->Text)+ Convert::ToDouble(textBox2->Text); break; case '-': k=Convert::ToDouble(textBox1->Text)- Convert::ToDouble(textBox2->Text); break; } textBox3->Text=k.ToString();</code>
<i>button15</i> (кнопка сброса — при нажатии на кнопку <i>button15</i> будет очищен текст в <i>textBox1</i> , <i>textBox2</i> , <i>textBox3</i> , <i>label1</i>)	<i>Properties</i>	<i>Text</i>	Ввести текст: C
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>Z=false; textBox1->Text=""; textBox2->Text=""; textBox3->Text=""; label1->Text="";</code>

2. Создать новый проект. Для создания нового проекта необходимо:

- выбрать в меню *File* пункт *New Projects*;
- в открывшемся окне на закладке *Projects types* выбрать тип проекта *CLR*;
- на закладке *Templates* выбрать *Windows Forms Application*;
- в строке *Name* ввести имя, например *Pow1*;
- щелкнуть мышью на кнопке *OK*.

3. Откроется форма *Form1*.

4. В меню *View* открыть панель инструментов *Toolbox*, во вкладке *Alle Windows Forms* выбрать элемент *TextBox* (Текстовое поле) и разместить *textBox1*

на форме *Form1*, затем на форму разместить *textBox2* и *textBox3* таким образом, чтобы они находились на одной линии.

5. Выбрать на панели инструментов *Toolbox* элемент *Label* (Метка) и перетащить *label1* на форму, чтобы метка была расположена между объектами *textBox1* и *textBox2*. Вторую метку *label2* расположить между *textBox2* и *textBox3*.

Для реализации программы *Pow1* в таблице 3 приводится пошаговая инструкция по написанию кода программы.

6. Сохранить программу *Pow1*. Для этого в меню *File* выбрать *Save All*.

Таблица 3

Выделенный объект	Окно <i>Properties</i>	Свойство или событие	Описание действий для выделенного объекта
Разместить на форме компоненты <i>textBox1</i> , <i>textBox2</i> , <i>textBox3</i> . Чтобы после ввода цифр в <i>textBox1</i> осуществлялся ввод во вторую компоненту — <i>textBox2</i> , необходимо перед областью описания класса в редакторе кода формы ввести логическую переменную <code>bool Z = false</code>			
<i>label1</i>	<i>Properties</i>	<i>Text</i>	Очистить текст
<i>label2</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: =
<i>button1</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: 1
	<i>Events</i>	<i>Click</i>	Вести текст в редактор кода: <code>if (Z==false) textBox1->Text+= ((button^).sender)->Text;</code> <code>else textBox2->Text+= ((button^).sender)->Text;</code>
Переименовать кнопки <i>button2</i> , <i>button3</i> , ..., <i>button10</i> , соответственно в 2, 3, ..., 0			
<i>button2</i> , ... <i>button10</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: 2, ..., 0
	<i>Events</i>	<i>Click</i>	Для каждой кнопки <i>button2</i> , <i>button3</i> , ..., <i>button10</i> назначить событие кнопки <i>button1</i> Нужно назначить событие кнопки <i>button1</i> на другие кнопки. Для этого в меню <i>Events</i> напротив вкладки <i>Click</i> для кнопок <i>button2</i> , <i>button3</i> , ..., <i>button10</i> выбрать из списка код: <code>button1_Click</code>
<i>button11</i> (кнопка для возведения в степень)	<i>Properties</i>	<i>Text</i>	Ввести текст: P
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>label1->Text= ((button^).sender)->Text; Z=true;</code>
При нажатии на кнопку <i>button12</i> происходит возведение в степень целых чисел. Число <i>N</i> вводится в <i>textBox1</i> , число <i>m</i> — в <i>textBox2</i> , и результат вычисления выводится в <i>textBox3</i>			
<i>button12</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: =
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>double k;</code> <code>switch (label1->Text[0])</code> <code>{case 'P': k=pow(Convert.ToInt32(textBox1->Text),</code> <code>Convert.ToInt32(textBox2->Text));break;}</code> <code>textBox3->Text=k.ToString();</code>
<i>button13</i> (кнопка сброса — при нажатии на кнопку <i>button13</i> будет очищен текст в <i>textBox1</i> , <i>textBox2</i> , <i>textBox3</i> , <i>label1</i>)	<i>Properties</i>	<i>Text</i>	Ввести текст: C
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>Z=false;</code> <code>textBox1->Text="";</code> <code>textBox2->Text="";</code> <code>textBox3->Text="";</code> <code>label1->Text="";</code>

7. Запустить программу *Pow1*. Для этого в меню *Debuge* выбрать *Start Debugging*.

* * *

Создание прикладных программ в средах объектного программирования развивает у учащихся умения самостоятельно переносить знания и умения в новую ситуацию, видеть новую проблему в знакомой ситуации, устанавливать новые свойства объекта изучения. Разработка калькулятора помогает начинающему пользователю разобраться с особенностями работы с разными типами представления данных в C++, познакомиться с библиотеками математических функций и их подключением при разработке прикладных приложений [5–7, 9].

Обучение программированию в средах объектно-ориентированных языков помогает достигнуть эффективного результата в формировании основных познавательных процессов у школьников: воображения, памяти, абстрактного и логического мышления.

Список использованных источников

1. Айра П. Объектно-ориентированное программирование на C++. М.: БИНОМ; Невский Диалект, 2001.

2. Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технология разработки программного обеспечения: учебное пособие для СПО. М.: Юрайт, 2018.

3. Зыков С. В. Программирование. Объектно-ориентированный подход: учебник и практикум для академического бакалавриата. М.: Юрайт, 2018.

4. Комлев Н. Ю. Объектно Ориентированное Программирование. Хорошая книга для Хороших Людей. М.: СОЛОН-Пресс, 2014.

5. Лафоре Р. Объектно-ориентированное программирование в C++. СПб.: Питер, 2015.

6. Лесневский А. С. Объектно-ориентированное программирование для начинающих. М.: БИНОМ. Лаборатория знаний, 2010.

7. Огнева М. В., Кудрина Е. В. Программирование на языке C++: практический курс: учебное пособие для СПО. М.: Юрайт, 2018.

8. Павловская Т. А. Процедурное и объектно-ориентированное программирование. C/C++: учебник. СПб.: Питер, 2015.

9. Трофимов В. В., Павловская Т. А. Алгоритмизация и программирование: учебник для академического бакалавриата / под ред. В. В. Трофимова. М.: Юрайт, 2018.

10. Тузовский А. Ф. Объектно-ориентированное программирование: учебное пособие для прикладного бакалавриата. М.: Юрайт, 2018.

НОВОСТИ

НР создала симуляцию марсианского поселения в виртуальной реальности

НР объединила силы с Nvidia и мировым творческим сообществом, чтобы реализовать проект HP Mars Home Planet — масштабную VR-симуляцию того, как могла бы выглядеть жизнь одного миллиона поселенцев на Марсе. Виртуальная симуляция HP Mars Home Planet — кульминация годового сотрудничества с дизайнерами, инженерами и изобретателями, в результате которого была смоделирована жизнь на Красной планете.

Разработка проекта привлекла более 90 тыс. представителей различных творческих профессий, архитекторов, инженеров и студентов из более чем 150 стран. Самые инновационные идеи были реализованы в виртуальной реальности от Technicolor, которая будет представлена на SIGGRAPH, крупнейшей в мире конференции, посвященной компьютерной графике.

«Максимальный эффект погружения в этой VR-симуляции достигнут благодаря кооперации творческого сообщества и технологий, которые предоставила НР и наши партнеры, — сказала Гвен Кобл (Gwen Coble), директор направления рабочих станций, тонких клиентов, решений для розничной торговли и иммерсивных вычислений в регионе ЕМЕА, НР Inc. — Передовые технологии и богатая история инноваций НР в сочетании с творческим подходом позволяют воплотить в жизнь самые потрясающие идеи».

Виртуальная симуляция программы HP Mars Home Planet демонстрирует центр адаптации новых поселенцев Марса, в котором показаны инновации в сферах архитектуры, техники и транспорта, которые позволят людям осваивать Марс. Компания Technicolor разработала визу-

альные образы с помощью игрового движка Unreal Engine Epic Games, в результате чего была создана первая в мире VR-программа с шестью степенями свободы для игровых кресел с поддержкой движения и гарнитур HP Windows Mixed Reality. Окружающая среда и основные объекты из VR будут доступны творческому сообществу уже этой осенью. Их можно будет скачать через Unreal Engine.

«Мы живем в интересное время, когда технологический прогресс идет рука об руку с множеством фундаментальных научных и планетарных исследований. Надеемся, что это сочетание приблизит тот момент, когда человек начнет осваивать Марс, — сказала доктор Дарлин Лим (Darlene Lim), научный руководитель проекта NASA Biologic Analog Science. — В наши дни люди по всему миру мечтают об инновациях, которые сделают такое будущее возможным, а этот удивительный проект — своего рода виртуальное окно, через которое мы можем увидеть, как будет выглядеть жизнь на Марсе для миллиона поселенцев».

Университеты и школы по всему миру также подхватили инициативу изобретений, проектирования и моделирования жизни в виртуальном поселении на Марсе. При поддержке технологий НР и ведущих специалистов в сфере образования в этих учреждениях были созданы команды энтузиастов, которые вошли в образовательную программу HP Mars Home Planet. В рамках этого проекта соревновались 10 европейских и два австралийских университета, а в Японии был проведен отдельный конкурс со 110 участниками. В результате были выбраны победители, чьи работы вошли в проект HP Mars Home Planet.

(По материалам CNews)