# DR. BOB DAVIDOV

# Знакомство с компилятором GNAT языка Ада для программирования систем управления автоматизированными комплексами

*Цель работы:* Получить представление об интегрированной среде разработки программ GNAT Programming Studio для специализированных платформ.

*Задача работы:* На примере готовой программы на языке Ада познакомиться со средствами разработки программ GNAT Programming Studio.

*Приборы и принадлежности:*.Персональный компьютер, компилятор GNAT.

# ОБЩИЕ СВЕДЕНИЯ

Язык программирования Ада (Ada) был создан для обеспечения надежности программного кода систем управления автоматизированными комплексами, прежде всего, бортовыми системами управления военными объектами (кораблями, самолётами, танками, ракетами, снарядами и т.п.) работающими в реальном времени. По утверждению сторонников этого языка программирования, разработка программного обеспечения на Аде в целом обходится на 60 % дешевле, а разработанная программа имеет в 9 раз меньше дефектов, чем при использовании языка Си.

GNAT Programming Studio (GPS) – открытая (бесплатная) многоязычная интегрированная среда разработки программ включающая редактор, компилятор, отладчик и профилировщик. Кроме языка Ада GPS поддерживает языки C, JavaScript, Pascal и Python. GPS использует компиляторы из коллекции компиляторов GNU.

#### НАСТРОЙКА GNAT НА КОМПИЛЯТОР ТРЕБУЕМОГО ТИПА

Изначально среда GNAT с компилятором GCC настроена на создание исполняемого кода для x86 контроллера. Для подключения к среде GNAT компилятора GCC, настроенного на создание кода для другого контроллера (например, хранимого в папке C:\ARM Toolchain), необходимо отредактировать системную переменную окружения Path, как показано на Puc. 2.

★c:\GNAT\2012\*.*	
Name	Ext
<b>€</b> []	
🗀 [bin]	
🗀 [DLLs]	
🗀 [etc]	
🗀 [i686-pc-mingw32]	
🗀 (include)	
🗀 [lib]	
🗀 [libexec]	
🗀 [share]	
👼 uninstall-gnatgpl-2012	exe

Рис. 1. Структура каталогов компилятора GCC.

1	öystem variables		
	Variable	Value	~
	NUMBER_OF_P	2	11
	OS	Windows_NT	
	Path	(c:\ARM Toolchain\bin;) C:\GNAT	_
	PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;	_
	PROCESSOR_A	x86	<b>*</b>
		New Edit Delete	

**Рис. 2.** Редактирование переменой окружения Path. My Computer > Properties > Advanced > Environment Variables > System variables > Path > Edit >

General	Languages			Apply changes to:
anguages	Language	No complier		Project
VCS				Tharness Deco
ource dirs				
ource files	C++			
Objects	Asm			
Anin Files	Asm2			
main riles	I I Asm Cod			
Library	Toolchains	Location Version		
Make	x86-windows (	(pative) C:)GNAT\2012) GPL 2012 (20120509)	Scan	
Naming	U XOO HIII GOHD		[add]	
5witches				
				Scenario
				Deenano
	Details			
	This section allows you To select an alternativ	i to modity individual tools for the selected toolchain. e toolchain, use the 'Add' button above		
	Tools:	•		
	GNAT Driver:	anat		
	GNAT List:	apath	French	
	Debugger:	yiidus	reset	
		gdb	reset	
				OK Cancel

						Apply shapped to:	
General	Languages	<b></b>				Show as hierarchy	
[Languages]						Project	•
VCS						Tharn	
Source dirs							
Source files	Asm						
Objects	Autocopf						
Main files					~		
Library	Toolchains						
Make	Name	Location	Version	I	Scan		
Maming	💿 arm-none-ea	bi C:\ARM Toolchain\ 4.6	5.1				
Cuitabaa					Add		
Switches						L	
						Scenario	
	Details						
	This section allows yo To select an alternati	ou to modify individual tools ive toolchain, use the 'Add'	r for the selected toolchain. ' button above		ñ		
	Tools				∃		
	GNAT Driver:	arm-none-eabi-goat					
	GNAT List:	aminone-eabi-gnat			rocot I		
	Debugger:	arm-none-eabi-ghads			reset		
		arm-none-eabi-gdb			reset 🗸		
(A. D.							
🔮 Proper	ties for Tharn						X
© Proper General	ties for Tharn					Apply changes to:	
Proper     General     Languages	ties for Tharn Make makefile: Makefile				Browse	Apply changes to:	
General Languages VCS	ties for Tharn Make makefile: Makefile				Browse	Apply changes to: Show as hierarchy Project	
Proper     General     Languages     VCS     Source dirs	ties for Tharn Make makefile: Makefile				Browse	Apply changes to: Show as hierarchy Project Tharn	-
General Languages VCS Source dirs Source files	ties for Tharn Make makefile: Makefile				Browse	Apply changes to: Show as hierarchy Project Tharn	
© Proper General Languages VCS Source dirs Source files Objects	ties for Tharn Make makefile: Makefile				Browse	Apply changes to: Show as hierarchy Project Tharn	
General Languages VCS Source dirs Source files Objects Main files	ties for Tharn Make makefile: Makefile				Browse	Apply changes to: Show as hierarchy Project Tharn	•
General Languages VCS Source dirs Source files Objects Main files	ties for Tharn Make makefile: Makefile				Browse	Apply changes to: Show as hierarchy Project	
© Proper General Languages VCS Source dirs Source files Objects Main files Library	ties for Tharn Make makefile: Makefile				Browse	Apply changes to: Show as hierarchy Project Tharn	
© Proper General Languages VCS Source dirs Source files Objects Main files Library Make	ties for Tharn				Browse	Apply changes to: Show as hierarchy Project Tharn	
© Proper General Languages VCS Source dirs Source files Objects Main files Library Make Naming	ties for Tharn Make makefile: Makefile				Browse	Apply changes to: Show as hierarchy Project Tharn	
General Languages VCS Source dirs Source files Objects Main files Library Make Naming Switches	ties for Tharn Make makefile: Makefile				Browse	Apply changes to: Show as hierarchy Project Tharn	
Proper     General     Languages     VCS     Source dirs     Source files     Objects     Main files     Library     Make     Naming     Switches	ties for Tharn Make makefile: Makefile				Browse	Apply changes to: Show as hierarchy Project Tharn	
Proper     General     Languages     VCS     Source dirs     Source files     Objects     Main files     Library     Make     Naming     Switches	ties for Tharn				Browse	Apply changes to: Show as hierarchy Project Tharn	
Proper     General     Languages     VCS     Source dirs     Source files     Objects     Main files     Library     Make     Naming     Switches	ties for Tharn Make makefile: Makefile Details				Browse	Apply changes to: Show as hierarchy Project Tharn	
Proper     General     Languages     VC5     Source dirs     Source files     Objects     Main files     Library     Make     Naming     Switches	ties for Tharn Make makefile: Makefile Makefile: Makefile Makefile: Makefile Details	ou to modify individual tool	ls for the selected toolchain, f button above		Browse	Apply changes to: Show as hierarchy Project Tharn	
Proper     General     Languages     VCS     Source dirs     Source files     Objects     Main files     Library     Make     Naming     Switches	ties for Tharn Make makefile: Makefile Makefile: Makefile Details This section allows y To select an alternat	ou to modify individual tool ive toolchain, use the 'Add	ls for the selected toolchain. If button above		Browse	Apply changes to: Show as hierarchy Project Tharn	
Proper     General     Languages     VCS     Source dirs     Source files     Objects     Main files     Library     Make     Naming     Switches	ties for Tharn Make makefile: Makefile Makefile: Makefile Details Details This section allows y To select an alternat Tools: GNAT Driver:	ou to modify individual tool ive toolchain, use the 'Add	k for the selected toolchain.		Browse	Apply changes to: Show as hierarchy Project Tharn	
Proper General Languages VCS Source dirs Source files Objects Main files Library Make Naming Switches	ties for Tharn Make makefile: Makefile Makefile: Makefile Details This section allows y To select an alternat Tools: GNAT Driver: GNAT List:	ou to modify individual tool ive toolchain, use the 'Add arm-none-eabi-gnat arm-none-eabi-gnat	ls for the selected toolchain.		Browse	Apply changes to: Show as hierarchy Project Tharn	
Proper     General     Languages     VC5     Source dirs     Source files     Objects     Main files     Library     Make     Naming     Switches	ties for Tharn Make makefile: Makefile Makefile: Makefile Details <i>This section allows yf</i> <i>To select an alternat</i> <b>Tools:</b> GNAT Driver: GNAT List: Debugger:	ou to modify individual tool ive tookhain, use the 'Ada arm-none-eabi-gnat arm-none-eabi-gnatls	ls for the selected tookhain.		Browse)	Apply changes to: Show as hierarchy Project Tharn	

General	Ada C Asm			Apply changes to:	
anguages	-Details	T.		Project	
VCS	Header files:	ш		🗹 Tharn	
ource dirs	Implementation:	 ,C			
ource files	Exceptions				
Objects	Filename		<u> </u>		
Objects	Filename		<u> </u>		1
Objects Propert	ties for Tharn		<u> </u>		
Objects Propert General	Filename ties for Tharn		<u> </u>	Apply changes to:	
Objects Propert General anguages	Filename ties for Tharn Ada C Asm Details		<u> </u>	Apply changes to:	
Objects Propert General anguages VCS	Filename ties for Tharn Ada C Asm Details Spec. Extension:		<u> </u>	Apply changes to: Show as hierarchy Project	
Objects Propert General anguages VCS	Filename         Ada       C         Ada       C         Ada       C         Asm         Details         Spec.       Extension:         Impl.       Extension:			Apply changes to: Show as hierarchy Project Tharn	
Objects Propert General anguages VCS ource dirs	Filename  ties for Tharn  Ada C Asm  Details  Spec. Extension: Impl. Extension: Exceptions			Apply changes to: Show as hierarchy Project Tharn	

**Рис. 3.** Подключение нового компилятора и языков программирования в среде GNAT: меню > Project > Edit Project Properties.

Примечание: Настройка среды GNAT может выполняться и через Makefile.

# НАСТРОЙКА КОМПИЛЯТОРА ПРОГРАММЫ

В соответствии с настройками компилятор помогает найти ошибки разного типа.

Настройка компилятора может выполняться через главное меню: Main Menu > Project > Edit Project Properties или с помощью gpr файла.

Для начальной настройки компилятора в меню среды GNAT предлагаются следующие установки.

anguages Cod VCS No ource dirs objects Library Make	le generation o optimization ] Inlining ] Unroll loops ] Position independent code ] Code coverage ] Instrument arcs	Run-time checks  Overflow checking  Suppress all checks  Stack checking  Dynamic elaboration	Show as hierarchy     Y Check attribute casing     Project     Y Use of array index nu     Y Deco     Y Blanks not allowed at:     Y Check boolean operat     Y Check no DOS line ten     Y Check	) mbers in array attributes statement end ors ninators present
Naming	] Always generate ALI file ] Separate function sections ] Separate data sections		We have been been as a constraint of the second secon	cal tabs rds
witches Mess	ssages ] Full errors /arnings: alldity checking mode: tyle checks: ck usage ] Generate stack usage information	Debugging  Debug Information  Enable assertions  Syntax  Ada 83 mode Ada 95 mode  Ada 2005 mode	Check layout     Scenario     Scenario     Set maximum nesting level     Check casing of entitie     Set maximum line length     Check order of subpro     Check that overriding     Check pragma casing     Check sparate specs     Check no statements.     Check to references     Check     Check to references     Check to references	12 6 s in Standard 120 g gram bodies subprograms are explicitly marked as such after THEN/ELSE ank lines

Рис. 4. Параметры настройки компилятора

Компилятор выдает список ошибок при их наличии. Для проверки нахождения ошибок в коде программы целесообразно использовать компиляцию отдельного модуля которая вызывается командой меню > Build > Compile file ().



Рис. 5. Пример вывода сообщения об ошибке в окне Location.

```
project Tharn is
for Object_Dir use "obj";
for Exec_Dir use ".";
for Main use ("Tharn.adb");
for Source_Dirs use (".", "..\..\Alg");
package Builder is
for Default_Switches ("ada") use ("-g");
end Builder;
package Compiler is
for Default_Switches ("ada") use ("-g", "-gnat05");
end Compiler;
```

package Linker is for Default\_Switches ("ada") use ("-g"); end Linker;

end Tharn;

**Рис. 6.** Пример настройки компилятора и редактора связей (Compile, Linker, Builder) в файле gpr.

# КОМПИЛЯЦИЯ КОДА В ЕХЕ ФАЙЛ

Компиляция программного кода в исполняемый ехе файл вызывается командой меню > Build

> Project > Build All ( ).
Примечание: Чтобы удалить файлы предыдущей компиляции используйте команду меню >

Build > Clean > "Clean All" ( (

#### ОТЛАДЧИК ПРОГРАММЫ

Загрузка отладчика Debugger <u>выполняется после создания ехе файла</u> по команде меню > Debug > Initialize > ехе файл

Чтобы установить / удалить точку остановки выполнения программы (breakpoint) щелкните по точке 🔹 второй колонки соответствующей линии, показанной на Рис. 7. Красная точка © отображает место, где остановится запущенная на выполнение программа.

Примечание: Для работы с точками останова необходимо установить следующие параметры отладчика.

- Meню > Project > Edit Project Properties > Switches > Gnatmake > Compilation > Debug information
- Meню > Project > Edit Project Properties > Switches > Ada > Debugging > Debug Information
- Meню > Project > Edit Project Properties > Switches > Ada Linker > Debug information

Чтобы увидеть значение программной переменной в точке останова укажите на нее мышью, нажмите правую кнопку и выберите Debug > Display. Имя и значение вычисленной переменной появятся в окне отладчика "Debugger Data" (см. Рис. 7).

Для копирования значения переменной укажите на нее в программе мышью, нажмите правую кнопку, выберите Debug > Print, выделите отображаемое значение и скопируйте его в буфер Paste (Ctrl+C).



**Рис. 7.** Редактирование переменой окружения Path. My Computer > Properties > Advanced > Environment

Для выполнения команд программы по отладчиком **Debugger** используйте следующие GPS команды

- Меню > Debug > Run/Continue или <F2>/<F2> или 
   началь
  - Меню > Debug > Step или <F5> или 🛅

•

начальный / повторный запуск

- пошаговое выполнение
- Меню > Debug > Next или <F6> или 🖾 пошаговое выполнение

• 🖾 выполнение до следующей breakpoint или выхода из модуля Для выхода из отладчика выполните меню > Debug > Terminate

# ЗАПУСК И ОСТАНОВ ЕХЕ ФАЙЛА В GNAT

Для запуска исполняемого файла в среде GNAT выполните команду меню > Build > Run > "имя файла"



или нажмите выделенную на рисунке кнопку:

Ввод и вывод данных исполняемого файла и результаты отображается в следующем окне Run среды GNAT



Рис. 8. Окно Run ввода-вывода исполняемого файла в среде GNAT

Остановку работающего исполняемого (exe) файла в GNAT можно выполнить командой

```
меню > Tools > Interrupt или Shift + Ctrl + C
```

или при помощи менеджера программ операционной среды Windows, выполнив последовательность:

Ctrl + Alt + Del > Windows Task Manager > Processes > выбор ехе файла в списке "Image Name" > End Process

#### ЗАПУСК ФАЙЛА ЧЕРЕЗ КОМАНДНОЕ ОКНО

Если программа, скомпилированная в среде GNAT, не содержит команд ввода (например, Get), то при запуске ехе файла в среде Windows, зачастую, не удается увидеть результаты программы поскольку окно с результатами открывается на мгновение время выполнения программы.

В случае, если исполняемый файл необходимо загружать за пределами среды GNAT и при этом необходимо сохранять на дисплее окно с выводимыми результатами выполните запуск файла через командную строку Windows:

Start **Start** > Run > Open > cmd

В открывшемся окне, используя команду cd (Change Directory - смена каталога), настройте путь на директорию с файлом, введите имя файла и нажмите клавишу Enter.

Примечание: Можно использовать копирование имени через буфер paste и ввод имени в командную строку командой paste, которая вызывается нажатием на правую клавиши мыши в командной строке.

#### ТЕСТИРОВАНИЕ ПРОГРАММ В GNAT

Модуль **gcov**, запускаемый из среды GNAT, анализирует выполнение программы на прохождение выбранных тестов. Модуль помогает выявить те части программы, которые не участвовали в тестировании. Информация о таких частях программы помогает оптимизировать тест программы и вектор входных параметров.

🔵 Propert	ies for THarn		X
General	Gnatmake Ada Binder Ada Linker Pretty Printer GnatC	heck	Apply changes to:
Languages	Code generation	Run-time checks	Project (
VCS	No optimization	Overflow checking	Tharn
Source dirs	Inlining	Suppress all checks	
Source files	🔲 Unroll loops	Stack checking	
Objects	Position independent code	Dynamic elaboration	
Main files	Code coverage		
Library	✓ Instrument arcs		
Make	📃 Always generate ALI file		
Naming	Separate function sections		
Switches	Separate data sections		
Dwitches	Messages	Debugging	
	E Full errors	Debug Information	
	Warnings:	Enable assertions	
	Validity checking mode:	Syntax	
	Style checks:	📃 Ada 83 mode	
		📃 Ada 95 mode	
		📃 Ada 2005 mode	
	Stack usage		
	Generate stack usage information		
	-g -ftestcoverage fprofile-arcs		



Рис. 9. Настойка среды GNAT на тестирование полноты выполнения кода (Test Coverage).

Tools Window He	lp		
⊻iews	•		
Browsers	•		
Coding <u>S</u> tandard	•		
C <u>o</u> mpare	•		
<u>C</u> onsoles	•		
Coverage	•	<u>G</u> cov	Compute coverage files
<u>D</u> ocumentation	•	Show report	Remove coverage files
M <u>a</u> cro	•		
<u>M</u> etrics	•	Load data for <u>a</u> ll projects	
🔕 Interrupt	Shift+Ctrl+C	Load data for current project	
<u>Plug-ins</u>		Clear coverage from memory	

Рис. 10. Команды выполнения теста Coverage.

Coverage Report		
Entities	Coverage	Percentage
😑 🚍 THarn	385 lines (107 not covered)	72 %
🗉 📄 ARM_Emulator.adb	25 lines (0 not covered)	100 %
🗉 📄 BC.adb	253 lines (95 not covered)	62 %
BC.ads	2 lines (2 not covered)	0%
🗉 📄 Physical_Units.adb	6 lines (6 not covered)	0%
🗉 📄 THarn_BC.adb	99 lines (4 not covered)	95 %
📄 Math.adb	undetermined	n/a
📄 Math.ads	undetermined	n/a

**Рис. 11**. Пример отчета выполнения теста Coverage. Видно, что модуль ARM\_Emulator.adb тестировался полностью, BC.adb – частично, а модуль Math.adb – не участвовал в тестировании.

# ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Задание 1. Компиляция программы Ада в среде GNAT

- Установите на компьютере среду GNAT Programming Studio(например, <u>http://libre.adacore.com/download/configurations</u> > GNAT 2012 > gnat-gpl-2012-i686-pcmingw32-bin.exe).
- 2. Создайте рабочий каталог с подкаталогами src и obj
- 3. В основной каталог переместите файл проекта traffic.gpr

★ f:\spark_code\*.*	
Name	+ Ext
<b>@</b> []	
🗀 [obj]	
🗀 [sɪc]	
📑 traffic	gpr

traffic.gpr project Traffic is

> for Object\_Dir use "obj"; for Exec\_Dir use "."; for Main use ("Traffic.adb"); for Source\_Dirs use (".", "src");

end Traffic;

```
4. В подкаталог src скопируйте файл Traffic.adb:
```

```
Traffic.adb
```

with Ada.Text\_IO; use Ada.Text\_IO;

procedure Traffic is

type Airplane ID is range 1..10; -- 10 airplanes (= tasks) task type Airplane (ID: Airplane ID); -- task type representing airplanes type Airplane Access is access Airplane; -- access type (reference) to Airplane protected type Runway is -- a protected object - the shared runway entry Assign Aircraft(ID: Airplane ID); entry Cleared\_Runway (ID : Airplane\_ID); entry Wait\_For\_Clear; private Clear: Boolean := True; -- protected private data - generally more than just a flag... end Runway; type Runway\_Access is access all Runway; -- the air traffic controller takes requests for takeoff and landing task type Controller(My\_Runway: Runway\_Access) is entry Request Takeoff (ID: in Airplane ID: Takeoff: out Runway Access); entry Request\_Approach(ID: in Airplane\_ID; Approach: out Runway\_Access); end Controller: Runway1 : aliased Runway; -- instantiate a runway Controller1: Controller(Runway1'Access); -- and a controller to manage it ----- the implementations of the above types ----protected body Runway is entry Assign Aircraft (ID : Airplane ID) when Clear is -- the entry guard - tasks are blocked until this is true begin Clear := False; Put\_Line (Airplane\_ID'Image (ID) & " on runway "); end: entry Cleared\_Runway (ID : Airplane\_ID) when not Clear is begin

Clear := True; Put\_Line (Airplane\_ID'Image (ID) & " cleared runway "); end;

```
entry Wait_For_Clear when Clear is begin
null;
end;
end Runway;
```

```
task body Controller is
 begin
   loop
     My Runway.Wait For Clear; -- wait until runway is available
                        -- wait for two types of requests
     select
       when Request_Approach'count = 0 \Rightarrow -- landings have priority
       accept Request Takeoff (ID : in Airplane ID; Takeoff : out Runway Access) do
         My_Runway.Assign_Aircraft (ID); -- reserve runway
        Takeoff := My_Runway; -- tell airplane which runway
       end Request_Takeoff; -- end of the synchronised part
     or
       accept Request_Approach (ID : in Airplane_ID; Approach : out Runway_Access) do
         My Runway.Assign Aircraft (ID);
        Approach := My_Runway;
       end Request_Approach;
                       -- terminate if nobody left who could call
     or
       terminate:
     end select;
   end loop;
 end:
 task body Airplane is
   Rwy: Runway_Access;
 begin
   Controller1.Request_Takeoff (ID, Rwy); -- wait to be cleared for takeoff
   Put_Line (Airplane_ID'Image (ID) & " taking off..."); delay 2.0;
   Rwy.Cleared Runway (ID);
   delay 5.0; -- fly around a bit...
   loop
     select -- try to request a runway
       Controller1.Request_Approach (ID, Rwy); -- this is a blocking call
       exit; -- if call returned we're clear for landing - proceed ...
     or delay 3.0; -- timeout - if no answer in 3 seconds, do something else
       Put Line (Airplane ID'Image (ID) & " in holding pattern");
     end select:
   end loop;
   delay 4.0; -- do landing approach...
   Put_Line (Airplane_ID'Image (ID) &"
                                               touched down!");
   Rwy.Cleared Runway(ID); -- notify runway that we're done here.
 end;
 New Airplane: Airplane Access;
begin
 for I in Airplane_ID'Range loop -- create a few airplane tasks
   New Airplane := new Airplane(I); delay 3.0;
 end loop;
```

end Traffic;

- 5. Загрузите среду **GNAT** и откройте проект **traffic.gpr**. Загрузить среду можно через загрузку файла проекта **gpr**.
- 6. Используя правую клавишу мыши откройте в среде GNAT gpr файл как показано на



Открытый файл можно редактировать.

- 7. Откройте программу на языке Ада: Traffic.adb.
- 8. Просмотрите код программы, найдите отличительные особенности языка Ада и редактора среды GNAT
- 9. Внесите в код Traffic.adb несколько ошибок разного рода.
- 10. Настройте стили компилятора как показано на Рис.4.
- 11. При помощи компилятора найдете и исправьте ошибки в программе.
- 12. Скомпилируйте исполняемый (ехе) файл.
- 13. Выполните ехе файл запуская его в среде GNAT, в Windows и в командной строке Windows. Найдите отличия в выполнении.
- 14. Настройте GNAT на режим отладки программы (см. соответствующий раздел выше)
- 15. Запустите отладчик.
- 16. Установите несколько точек останова в программе (см. Рис. 7).
- 17. Выведите на экран поля значений нескольких переменных.
- 18. Выполните программу под отладчиком изменяя его режимы: выполнение до точки останова (breakpoint), выполнение в пошаговом режиме, и др. Постарайтесь определить назначение программы.
- 19. Выйдете из режима отладки.

### контрольные вопросы

- 1. Для чего предназначена и как работает программа Traffic.adb.
- 2. Какие библиотечные функции использует программа.
- 3. Назовите основные модули интегрированной среды GNAT.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1. Гавва А.Е. "Адское" программирование. Ada 95. Компилятор GNAT, V-0.4, май 2004. 431с., <u>http://www.ada-ru.org/</u>
- 2. Язык программирования SPARK ADA <u>http://www.adahome.com/Tutorials/Lovelace/lovelace.htm</u>
- 3. Компилятор GNAT. (Компилятор GCC для Ada интегрирован в GNAT) <u>http://www.gnu.org/s/gnat/</u>
- 4. Компилятор GCC для Ada <u>http://gcc.gnu.org/</u>