

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГТУ»)

Факультет экономики, менеджмента и информационных технологий
(факультет)

Кафедра Систем управления и информационных технологий в строительстве

КУРСОВОЙ ПРОЕКТ

по дисциплине: «Архитектура и организация ЭВМ»

Тема: «Использование табличного редактора для выполнения операций с
числами в различных системах счисления. Анализ современных систем
программирования »

Расчетно-пояснительная записка

Разработал студент

27.12.2021 Д.В. Тюленев
Подпись, дата Инициалы, фамилия

Руководитель

27.12.2021 А.В. Норман
Подпись, дата Инициалы, фамилия

Члены комиссии

Подпись, дата Инициалы, фамилия

Подпись, дата Инициалы, фамилия

Нормоконтроллер

Подпись, дата Инициалы, фамилия

Защищена 24.12.2021
дата

Оценка 5. Отм.

ВОРОНЕЖ

2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГТУ»)

Кафедра Систем управления и информационных технологий в строительстве

ЗАДАНИЕ
на курсовой проект

по дисциплине: «Архитектура и организация ЭВМ»

Тема: «Использование табличного редактора для выполнения операций с числами в различных системах счисления. Анализ современных систем программирования»

Студент группы БИСТ-214 Тюленев Данил Вячеславович

Фамилия, имя, отчество

Вариант: № 23

Технические условия процессор Intel® Core™ i3-8145U CPU @ 2,10 ГГц,
операционная система Windows 10, ОЗУ 8192 МБ

Содержание и объем проекта (графические работы, расчёты и прочее):

52 стр, 51 рисунков, 1 таб

Сроки выполнения этапов: 01.09.2021 г. – 01.12.2021 г.

Срок защиты курсового проекта: 15.12.2021 г.

Руководитель



27.12.2021.

Подпись, дата

А.В. Норман

Инициалы, фамилия

Задание принял студент



27.12.2021

Подпись, дата

Д.В. Тюленев

Инициалы, фамилия

Содержание

Содержание.....	3
Введение.....	4
1. Использование табличного редактора для выполнения операций с числами в различных системах счисления.....	5
1.1. Задание № 1 (Перевод чисел в различные СС).....	5
1.2. Задание № 2: (Действия над числами в двоичной, десятичной, восьмеричной и шестнадцатеричной СС).....	13
2. Анализ современные системы программирования.....	24
2.1. Понятие системы программирования, основные компоненты.....	24
2.2. Современные системы/среды программирования.....	29
2.3. Понятие интегрированной среды разработки ПО.....	29
2.4. Microsoft Visual Studio.....	30
2.5. IntelliJ IDEA.....	36
2.6. Eclipse.....	41
2.7. Xcode.....	45
2.8. NetBeans.....	47
Заключение.....	50
Библиографический список.....	51

Введение

Вместе с появлением электронно-вычислительных машин, начинают формироваться и развиваться системы программирования, появляются новые языки программирования и среды разработки программ. В настоящее время наблюдается стремительное развитие научной дисциплины, называемой программирование. К сегодняшнему дню насчитывают несколько поколений систем программирования.

Выбранная для исследования тема является крайне актуальной в области разработки ПО различного назначения, поскольку системы программирования являются универсальными средствами работы с информацией. В настоящее время системы программирования применяются в самых различных областях человеческой деятельности: научные вычисления, системное программирование, обработка информации, искусственный интеллект и т. д.

Мы можем наблюдать тенденцию развития инструментов разработки программного обеспечения. Как и другие традиционные инженерные дисциплины, разработка программного обеспечения имеет дело с проблемами качества, стоимости и надёжности. Некоторые программы содержат миллионы строк исходного кода, которые, как ожидается, должны правильно исполняться в изменяющихся условиях. Современные системы программирования для персональных компьютеров предоставляют пользователю весьма мощные и удобные средства для разработки программ. Они призваны полностью или частично автоматизировать, ускорить и сократить процесс жизненного цикла программы: от создания архитектуры ПО до внедрения его в продуктивную среду, более того, современные интегрированные средства разработки позволяют настроить окружение под конкретные задачи программиста и использовать всевозможные “расширения” для индивидуализации этих процессов.

Целью данной работы является исследование современных систем программирования, выявление сильных и слабых сторон каждой из них.

1. Использование табличного редактора для выполнения операций с числами в различных системах счисления

Цель работы:

1. Перевести число из одной (изначальной) системы счисления в указанную заданием.
2. Выполнить математические операции над числами указанных систем счисления.

Перевод чисел в различные СС	Действия над числами в двоичной, десятичной, восьмеричной и шестнадцатеричной СС (указаны в десят-ой)
$1618_{10} \rightarrow A_2$ $0,101_2 \rightarrow A_8$ $44,B1_{16} \rightarrow A_2$ $3,662_8 \rightarrow A_{10}$	$586,3+96,42$ $55,12-9,88$ $64,22*833,5$ $355,71 / 4$

Рисунок 1 — Таблица с примерами СС

1.1. Задание № 1 (Перевод чисел в различные СС)

В соответствии с поставленной задачей, необходимо произвести перевод исходных чисел из начальной системы счисления в указанную.

1. $1618_{10} \rightarrow A_2$:

Перевод целых десятичных чисел в любую другую системы счисления осуществляется делением числа на основание новой системы счисления до тех пор, пока в остатке не останется число меньше основания новой СС. Новое число записывается в виде остатков деления, начиная с последнего.

Для того чтобы совершить перевод числа из десятичной СС в двоичную, необходимо последовательно производить операцию деления исходного числа на соответствующее основание конечной СС (делить на 2) до момента пока число не обратится в единицу, при этом необходимо сохранять значение

остатка от деления. После нужно записать множество остатков в обратном направлении (справа налево), данная запись будет являться итоговым двоичным числом. Ответ: $1618_{10} = 11001010010_2$.

Перевод целой части числа										Вход: 1618
1618	2									
-1618	809	2								
0	-808	404	2							
	1	-404	202	2						
		0	-202	101	2					
			0	-100	50	2				
				1	-50	25	2			
					0	-24	12	2		
						1	-12	6	2	
							0	-6	3	2
								0	-2	1
									1	
←										
Ответ: 11001010010										

Рисунок 2 — Перевод из десятичной системы в двоичную

Иногда, чтобы ускорить процесс перевода целой части числа в определённую систему счисления, используется специальная таблица, демонстрирующая соотношение чисел в разных СС.

Табличный перевод СС			
Основание системы счисления			
10	2	8	16
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Рисунок 3 — Таблица соответствий чисел в разных СС

Для перевода числа в табличном редакторе необходимо реализовать: процесс сохранения остатка от деления для каждой итерации и нового значения после деления на два. Для сохранения остатка применяется формула «ОСТАТ(X;2)», где X — адрес клетки с исходным числом для данной итерации цикла последовательного деления. Для определения нового значения (для следующего повторения) используется формула «ОКРВНИЗ(X/2;1)», X — адрес исходного числа данного блока. После необходимо растянуть сформированную формулу вдоль столбцов для получения результатов для разрядов нового двоичного числа. После применяется функции «=ЦЕЛОЕ(СЦЕПИТЬ(Y_n;Y_{n-1};...))» для получения итогового двоичного числа, где Y — адрес клетки с остатком.

Перевод 10 -> 2 СС				Вход: 1618	
1618	:	2	=	809	0
809	:	2	=	404	1
404	:	2	=	202	0
202	:	2	=	101	0
101	:	2	=	50	1
50	:	2	=	25	0
25	:	2	=	12	1
12	:	2	=	6	0
6	:	2	=	3	0
3	:	2	=	1	1
1	:	2	=	0	1
0	:	2	=	0	0
0	:	2	=	0	0
Ответ: 11001010010					

Рисунок 4 — Перевод из десятичной системы в двоичную, используя табличный редактор

2. $0,101_2 \rightarrow A_8$:

Чтобы произвести перевод числа из двоичной СС в восьмеричную СС необходимо произвести промежуточный перевод числа в десятичную СС. Для перевода в десятичную систему счисления необходимо найти сумму произведений основания текущей системы счисления на соответствующую степень разряда: $A = x_{k-1} \cdot n^{k-1} + x_{k-2} \cdot n^{k-2} + \dots + x_0 \cdot n^0 + x_{-1} \cdot n^{-1} + \dots + x_{-j} \cdot n^{-j}$,

где k — количество разрядов целой части, j — количество разрядов дробной части, n — основание системы счисления.

$0.101_2 = 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 0 + 0.5 + 0 + 0.125 = 0.625_{10}$ — перевод из двоичной СС в десятичную.

Для перевода полученного дробного числа в восьмеричную необходимо производить последовательное умножение его на основание новой СС.

Как итог, получаем ответ: $0.101_2 = 0.5_8$.

	0.625 (10)	
	0	.625
		* 8
	5	.0
		* 8
	0	.0
	0.50 (8)	

Рисунок 5 — Перевод дробного числа из десятичной системы в восьмеричную

Чтобы осуществить перевод в десятичную систему, используя редактор, нужно произвести разделение разрядов исходного числа в отдельные ячейки. Для этого нужно последовательно производить умножение числа на 10, сохраняя результат в соседних ячейках строки, и определять остаток от деления целой части на 10 с помощью «ОСТАТ(ЦЕЛОЕ(X);10)», где X — адрес клетки с сохраненным значением. После необходимо выставить последовательно в ячейках одной строки значения степени для соответствующего разряда. После необходимо вычислить элементы для конечной суммы: « $A \cdot \text{СТЕПЕНЬ}(2;B)$ », где A — соответствующая ячейка с разрядом исходного числа, B — степень для разряда. После получаем готовое число в десятичной СС путём применения функции «СУММ», в качестве аргумента выставлен диапазон ячеек с ранее вычисленными элементами.

Для дальнейшего перевода полученного дробного числа из десятичной СС в необходимо находить целую часть от произведения дробной части исходного десятичного числа (или дробной части после предыдущей итерации) на 8, используя формулу «ОКРВНИЗ($X \cdot 8; 1$)», где X - дробная часть исходного числа, или прошлого повтора. Прямая последовательность найденных целых частей будут формировать готовое восьмеричное число. Параллельно этим действиям, необходимо сохранять остаточную дробную часть: « $X \cdot 8 - \text{ОКРВНИЗ}(X \cdot 8; 1)$ ».

Перевод дробной части из (Двоичной) в X (Восьмеричную)								Обозначения																																
Входное значение: 0,101																																								
Перевод из двоичной системы в десятичную						Вход: 0,101																																		
0,101	1,01	10,1	101	1010	10100	101000	1010000	Нахождение разрядов																																
0	1	0	1	0	0	0	0																																	
0	-1	-2	-3	-4	-5	-6	-7	степень системы																																
0	0,5	0	0,125	0	0	0	0	элементы суммы																																
Ответ: 0,625								Результат																																
<table><tr><th colspan="4">Перевод из десятичной в восьмеричную</th></tr><tr><td>0,625</td><td>* 8 =</td><td>5</td><td>0</td></tr><tr><td>0</td><td>* 8 =</td><td>0</td><td>0</td></tr><tr><td>0</td><td>* 8 =</td><td>0</td><td>0</td></tr><tr><td>0</td><td>* 8 =</td><td>0</td><td>0</td></tr><tr><td>0</td><td>* 8 =</td><td>0</td><td>0</td></tr><tr><td>0</td><td>* 8 =</td><td>0</td><td>0</td></tr><tr><td colspan="2">Ответ:</td><td colspan="2">0,500000</td></tr></table>								Перевод из десятичной в восьмеричную				0,625	* 8 =	5	0	0	* 8 =	0	0	0	* 8 =	0	0	0	* 8 =	0	0	0	* 8 =	0	0	0	* 8 =	0	0	Ответ:		0,500000		
Перевод из десятичной в восьмеричную																																								
0,625	* 8 =	5	0																																					
0	* 8 =	0	0																																					
0	* 8 =	0	0																																					
0	* 8 =	0	0																																					
0	* 8 =	0	0																																					
0	* 8 =	0	0																																					
Ответ:		0,500000																																						
Итог перевода: 0,5								Результат																																

Рисунок 6 — Перевод числа из двоичной системы в восьмеричную с использованием табличного редактора

3. $44, B1_{16} \rightarrow A_2$:

Для перевода из 16-ой СС в 2-ую СС необходимо произвести промежуточную трансляцию в 10-ую СС.

$44, B1_{16} = 4 \cdot 16^1 + 4 \cdot 16^0 + 11 \cdot 16^{-1} + 1 \cdot 16^{-2} = 64 + 4 + 0.6875 + 0.00390625 = 68.69140625_{10}$ — перевод в десятичную систему счисления.

После чего полученное число переводится в 2-ую СС. Для этого необходимо разделить число на две части: целую и дробную, и после чего по отдельности производить трансляцию частей в нужную СС.

Перевод целой части числа					Вход:	68
68	2					
-68	34	2				
0	-34	17	2			
	0	-16	8	2		
		1	-8	4	2	
			0	-4	2	2
				0	-2	1
					0	
←						
Ответ:					1000100	

Рисунок 7 — Перевод целой части десятичного числа в двоичную СС

0.69140625 (10)	
0	.625
	* 2
1	.38281
	* 2
0	.76563
	* 2
1	.53125
	* 2
1	.0625
	* 2
0	.125
	* 2
0	.25
	* 2
0	.5
	* 2
1	.0
0.10110001 (2)	

Рисунок 8 — Перевод дробной части десятичного числа в двоичную СС

Далее необходимо сложить полученные значения для получения готового числа в 2-ой СС: $1000100_2 + 0.10110001_2 = 1000100.10110001_2$. В результате получаем: $44.B_{16} = 1000100.10110001_2$.

Для перевода из 16-ой в 10-ую СС через табличный редактор необходимо разбить число на две части: целую — «ЦЕЛОЕ(ПСТР(\$F;1;НАЙТИ(", " ;\$F)-1))»;

и дробную — «ПСТР(\$F;НАЙТИ(";", \$F)+1;10)» (\$F — адрес клетки исходного числа). Далее необходимо произвести разделение разрядов частей числа на отдельные клетки: «ПСТР(\$F;ABS(A);1)», где А — адрес ячейки с индексом разряда. После чего необходимо произвести промежуточный перевод цифр из шестнадцатеричной СС в десятичную СС: «SWITCH (E; "B"; 11; "C"; 12; "D"; 13; "E"; 14; "F"; 15; ""; 0)». Формируем элементы суммы: «А*СТЕПЕНЬ(16;В)», где А — промежуточный перевод цифр, В — индекс разряда. Далее применяем функцию «СУММ» для каждой из частей числа (целая и дробная) и производим сложение результатов — получаем итоговое число в десятичной СС.

Перевод 16 -> 10				Вход:	44,B1	Обозначения
Перевод целой части				Вход:	44	
0	1	2	3	4	5	Степень основания
1	10	100	1000	10000	100000	
4	4	0	0	0	0	разряд исходного
4	64	0	0	0	0	элементы суммы
Ответ:				68		
Перевод дробной части				Вход:	B1	
-1	-2	-3	-4	-5	-6	Степень основания
8	1					разряд исходного
11	1	0	0	0	0	десятичное представление
0,6875	0,00390625	0	0	0	0	элементы суммы
Ответ:				0,69140625		
Итог:				68,69140625		

Рисунок 9 — Перевод из шестнадцатеричной СС в десятичную СС с использованием табличного редактора

Для перевода целой готового числа в 2-ую СС будем циклично сохранять целую часть результата от деления исходного числа на 2, с помощью «ЦЕЛОЕ($X/2$)», и остаток от деления, с помощью «ОСТАТ($X;2$)», где где X — исходное значение. После соединить остатки в одну ячейку, используя ЦЕЛОЕ(СЦЕПИТЬ($A, B, C...$))), где A, B, C — адреса ячеек с значениями остатков. Для перевода дробной части необходимо циклично сохранять целую и дробную часть от произведения начального значения на основание системы для следующей итерации. После соединить целые значения и получить готовый перевод. Под конец необходимо произвести сложение перевода дробной и целой части для получения итогового результата.

Перевод 10 -> 2 СС				Вход: 68,69140625	
Перевод целой части				Вход: 68	
68	:	2	=	34	0 ↑
34	:	2	=	17	0
17	:	2	=	8	1
8	:	2	=	4	0
4	:	2	=	2	0
2	:	2	=	1	0
1	:	2	=	0	1
0	:	2	=	0	0
0	:	2	=	0	0
0	:	2	=	0	0
0	:	2	=	0	0
0	:	2	=	0	0
0	:	2	=	0	0
0	:	2	=	0	0
Ответ:				1000100	
Перевод дробной части				Вход: 0,69140625	
исходное	система	целая	остаток		
0,69140625	*2=	1	0,3828125		
0,3828125	*2=	0	0,765625		
0,765625	*2=	1	0,53125		
0,53125	*2=	1	0,0625		
0,0625	*2=	0	0,125		
0,125	*2=	0	0,25		
0,25	*2=	0	0,5		
0,5	*2=	1	0		
Ответ:				0,10110001	
Итог:				1000100,10110001	

Рисунок 10 — Перевод из 10-ой СС в 2-ую СС с использование табличного редактора

4. $3,662_8 \rightarrow A_{10}$:

$3,662_8 = 3 \cdot 8^0 + 6 \cdot 8^{-1} + 6 \cdot 8^{-2} + 2 \cdot 8^{-3} = 3 + 0,75 + 0,09375 + 0,00390625 = 3,84765625_{10}$ — перевод в десятичную систему счисления.

Чтобы реализовать данный перевод в электронной таблице, нужно произвести деление числа на 2 части: дробную и целую. Необходимо произвести перевод отдельных разрядов: «X*СТЕПЕНЬ(8;Y)», где X - ячейка с разрядом, Y - ячейка с соответствующей степенью. После чего нужно произвести сложение всех переведённых разрядов через функцию «СУММ». Далее, как итог, сложить переведённые части исходного числа.

Перевод 8 -> 10					Вход: 3,662	Обозначения
Перевод целой части					Вход: 3	
0	1	2	3	4	Степень основания	
1	10	100	1000	10000	разряд исходного	
3	0	0	0	0	элементы суммы	
3	0	0	0	0		
Ответ: 3						
Перевод дробной части					Вход: 0,662	
0	-1	-2	-3	-4	Степень системы	
0	6	6	2	0	разряд исходного	
0	0,75	0,09375	0,00390625	0	элементы суммы	
Ответ: 0,84765625						
Итог: 3,84765625						

Рисунок 11 — Перевод из восьмеричной СС в десятичную СС с использованием табличного редактора

1.2. Задание № 2 (Действия над числами в двоичной, десятичной, восьмеричной и шестнадцатеричной СС)

Алгоритм решения: В начале необходимо перевести исходные числа из 10-ой системы счисления в представленных выражениях в необходимую СС.

Далее выполнить соответствующую заданию математическую операцию над переведёнными числами в СС.

1. $586,3_{10} + 96,42_{10}$:

Десятичная СС	96,42
Двоичная СС	1100000,0110
Восьмеричная СС	140,3270
Шестнадцатеричная СС	60,6B85

Рисунок 12 — Соответствие числа 96,42 в разных СС

Десятичная СС	586,3
Двоичная СС	1001001010,0100
Восьмеричная СС	1112,2314
Шестнадцатеричная СС	24A,4CCC

Рисунок 13 — Соответствие числа 586,3 в разных СС

10-ая СС -> 2-ая СС		10-ая СС -> 8-ая СС		10-ая СС -> 16-ая СС	
Исходная часть	Остаток от деления	Исходная часть	Остаток от деления	Исходная часть	Остаток от деления
Целая часть		Целая часть		Целая часть	
586	0	586	2	586	A
293	1	73	1	36	4
146	0	9	1	2	2
73	1	1	1	Результат:	24A
36	0	Результат:	1112	Дробная часть	
18	0	Дробная часть		0,3	4
9	1	0,3	2	0,8	C
4	0	0,4	3	0,8	C
2	0	0,2	1	0,8	C
1	1	0,6	4	Результат:	0,4CCC
Результат:	1001001010	Результат:	0,2314	Итог:	24A,4CCC
Дробная часть		Итог:	1112,2314		
0,3	0				
0,6	1				
0,2	0				
0,4	0				
Результат:	0,0100				
Итог:	1001001010,0100				

Рисунок 14 — Процесс перевода числа 586,3 в разные СС

10-ая СС -> 2-ая СС		10-ая СС -> 8-ая СС		10-ая СС -> 16-ая СС	
Исходная часть	Остаток от деления	Исходная часть	Остаток от деления	Исходная часть	Остаток от деления
Целая часть		Целая часть		Целая часть	
96	0 ↑	96	0 ↑	96	0 ↑
48	0 ↑	12	4 ↑	6	6 ↑
24	0 ↑	1	1 ↑	Результат:	60
12	0 ↑	Результат:	140	Дробная часть	
6	0 ↑	Дробная часть		0,42	6 ↓
3	1 ↑	0,42	3 ↓	0,72	8 ↓
1	1 ↑	0,36	2 ↓	0,52	8 ↓
Результат:	1100000	0,88	7 ↓	0,32	5 ↓
Дробная часть		0,04	0 ↓	Результат:	0,6885
0,42	0 ↓	Результат:	0,3270	Итог:	60,6885
0,84	1 ↓	Итог:	140,3270		
0,68	1 ↓				
0,36	0 ↓				
Результат:	0,0110				
Итог:	1100000,0110				

Рисунок 15 — Процесс перевода числа 96,42 в разные СС

Сложение в 2-ой СС -> 586,3+96,42													Обозначения	
Целая часть									Дробная часть					
1	0	0	1	0	0	1	0	1	0	0	1	0	0	1-ое слагаемое
			1	1	0	0	0	0	0	0	1	1	0	2-ое слагаемое
0	0	1	0	0	0	0	0	0	0	1	0	0	0	В новый разряд
1	0	1	0	1	0	1	0	1	0	1	0	1	0	Результат
Итог:				1010101010,1010										

Рисунок 16 — Операция сложения чисел 586,3 и 96,42 в 2-ой СС

Сложение в 8-ой СС -> 586,3+96,42														Обозначения	
Целая часть										Дробная часть					
						1	1	1	2	2	3	1	4	1-ое слагаемое	
							1	4	0	3	2	7	0	2-ое слагаемое	
						0	0	0	0	0	1	0	0	В новый разряд	
						1	2	5	2	5	6	0	4	Результат	
Итог:					1252,5604										

Рисунок 17 — Операция сложения чисел 586,3 и 96,42 в 8-ой СС

Сложение в 10-ой СС -> 586,3+96,42													Обозначения	
Целая часть									Дробная часть					
							5	8	6	3	0	0	0	1-ое слагаемое
								9	6	4	2	0	0	2-ое слагаемое
							1	1	0	0	0	0	0	В новый разряд
							6	8	2	7	2	0	0	Результат
Итого:				682,7200										

Рисунок 18 — Операция сложения чисел 586,3 и 96,42 в 10-ой СС

Сложение в 16-ой СС -> 586,3+96,42													Обозначения	
Целая часть									Дробная часть					
							2	4	A	4	C	C	C	1-ое слагаемое
								6	0	6	B	8	5	2-ое слагаемое
							0	0	0	1	1	1	0	В новый разряд
							2	10	10	11	8	5	0	Результат
							2	A	A	B	8	5	0	
Итого:				2AA,B850										

Рисунок 19 — Операция сложения чисел 586,3 и 96,42 в 16-ой СС

2. $55,12_{10} - 9,88_{10}$:

Десятичная СС	9,88
Двоичная СС	1001,1110
Восьмеричная СС	11,7024
Шестнадцатеричная СС	9,E147

Рисунок 20 — Соответствие числа 9,88 в разных СС

Десятичная СС	55,12
Двоичная СС	110111,0001
Восьмеричная СС	67,0753
Шестнадцатеричная СС	37,1EB8

Рисунок 21 — Соответствие числа 55,12 в разных СС

Исходная	Остаток от	Исходная	Остаток от	Исходная	Остаток от
Целая часть		Целая часть		Целая часть	
55	1 ↑	55	7 ↑	55	7 ↑
27	1	6	6	3	3
13	1	Результат:	67	Результат:	37
6	0	Дробная часть		Дробная часть	
3	1	0,12	0	0,12	1
1	1	0,96	7	0,92	E
Результат:	110111	0,68	5	0,72	B
Дробная часть		0,44	3	0,52	B
0,12	0	Результат:	0,0753	Результат:	0,1EB8
0,24	0	Итог:	67,0753	Итог:	37,1EB8
0,48	0				
0,96	1				
Результат:	0,0001				
Итог:	110111,0001				

Рисунок 22 — Процесс перевода числа 55,12 в разные СС

Исходная	Остаток от	Исходная	Остаток от	Исходная	Остаток от
Целая часть		Целая часть		Целая часть	
9	1 ↑	9	1 ↑	9	9
4	0	1	1	Результат:	9
2	0	Результат:	11	Дробная часть	
1	1	Дробная часть		0,88	E
Результат:	1001	0,88	7	0,08	1
Дробная часть		0,04	0	0,28	4
0,88	1	0,32	2	0,48	7
0,76	1	0,56	4	Результат:	0,E147
0,52	1	Результат:	0,7024	Итог:	9,E147
0,04	0	Итог:	11,7024		
Результат:	0,1110				
Итог:	1001,1110				

Рисунок 23 — Процесс перевода числа 9,88 в разные СС

Вычитание в 2-ой СС -> 55,12 - 9,88													Обозначения	
Целая часть									Дробная часть					
				1	1	0	1	1	1	0	0	0	1	Уменьшаемое
						1	0	0	1	1	1	1	0	Вычитаемое
					*			*						Занять
				1	0	1	1	0	1	0	0	1	1	Результат
Итог:				101101,0011										

Рисунок 24 — Операция вычитания с числами 55,12 и 9,88 в 2-ой СС

Вычитание в 8-ой СС -> 55,12 - 9,88												Обозначения	
Целая часть								Дробная часть					
							6	7	0	7	5	3	Уменьшаемое
							1	1	7	0	2	4	Вычитаемое
								*			*		Занять
							5	5	1	7	2	7	Результат
Итог:				55,1727									

Рисунок 25 — Операция вычитания с числами 55,12 и 9,88 в 8-ой СС

Вычитание в 10-ой СС -> 55,12 - 9,88												Обозначения		
Целая часть								Дробная часть						
							5	5	1	2	0	0	Уменьшаемое	
									9	8	8	0	0	Вычитаемое
								*						Занять
							4	5	2	4	0	0	Результат	
Итог:				45,2400										

Рисунок 26 — Операция вычитания с числами 55,12 и 9,88 в 10-ой СС

Вычитание в 16-ой СС -> 55,12 - 9,88												Обозначения	
Целая часть								Дробная часть					
							3	7	1	E	B	8	Уменьшаемое
								9	E	1	4	7	Вычитаемое
							*						Занять
							2	13	3	13	7	1	Результат
							2	D	3	D	7	1	
Итог:				2D,3D71									

Рисунок 27 — Операция вычитания с числами 55,12 и 9,88 в 16-ой СС

3. $64,22_{10} * 833,5_{10}$:

Десятичная СС	833,5
Двоичная СС	1101000001,1000
Восьмеричная СС	1501,4000
Шестнадцатеричная СС	341,8000

Рисунок 28 — Соответствие числа 833,5 в разных СС

Десятичная СС	64,22
Двоичная СС	1000000,0011
Восьмеричная СС	100,1605
Шестнадцатеричная СС	40,3851

Рисунок 29 — Соответствие числа 64,22 в разных СС

Исходная часть	Остаток от деления	Исходная часть	Остаток от деления	Исходная часть	Остаток от деления
Целая часть		Целая часть		Целая часть	
64	0 ↑	64	0 ↑	64	0 ↑
32	0	8	0	4	4 ↑
16	0	1	1	Результат:	40
8	0	Результат:	100	Дробная часть	
4	0	Дробная часть		0,22	3
2	0	0,22	1 ↓	0,52	8 ↓
1	1	0,76	6 ↓	0,32	5 ↓
Результат:	1000000	0,08	0	0,12	1 ↓
Дробная часть		0,64	5 ↓	Результат:	0,3851
0,22	0 ↓	Результат:	0,1605	Итог:	40,3851
0,44	0	Итог:	100,1605		
0,88	1				
0,76	1 ↓				
Результат:	0,0011				
Итог:	1000000,0011				

Рисунок 30 — Процесс перевода числа 64,22 в разные СС

Исходная часть	Остаток от деления	Исходная часть	Остаток от деления	Исходная часть	Остаток от деления
Целая часть		Целая часть		Целая часть	
833	1 ↑	833	1 ↑	833	1 ↑
416	0	104	0	52	4 ↑
208	0	13	5	3	3
104	0	1	1	Результат:	341
52	0	Результат:	1501	Дробная часть	
26	0	Дробная часть		0,5	8
13	1	0,5	4 ↓	0	0
6	0	0	0	0	0
3	1	0	0 ↓	0	0 ↓
1	1	0	0	Результат:	0,8000
Результат:	1101000001	Результат:	0,4000	Итог:	341,8000
Дробная часть		Итог:	1501,4000		
0,5	1 ↓				
0	0				
0	0				
0	0 ↓				
Результат:	0,1000				
Итог:	1101000001,1000				

Рисунок 31 — Процесс перевода числа 833,5 в разные СС

Умножение в 2-ой СС -> 833,5*64,22																				Обозначения
Целая часть															Дробная часть					Множимое
															1	0	0	0	0	
															1	0	0	0	0	Множитель
															1	0	0	0	0	Процесс перемножения
															1	0	0	0	0	
															0	0	0	0	0	
															0	0	0	0	0	
															0	0	0	0	0	
															0	0	0	0	0	
															0	0	0	0	0	
															0	0	0	0	0	
															0	0	0	0	0	
															0	0	0	0	0	
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	Результат
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
Итого:		1101000011111100,01001000																		

Рисунок 32 — Операция умножения с числами 64,22 и 833,5 в 2-ой СС

Умножение в 8-ой СС -> 833,5*64,22																				Обозначения
Целая часть															Дробная часть					Множимое
															1	5	0	1	4	
															1	0	0	1	6	Множитель
															5	31	0	5	24	Процесс перемножения
															0	0	0	0	0	
															6	36	0	6	30	
															1	5	0	1	4	
															0	0	0	0	0	
															0	0	0	0	0	
															1	5	0	1	4	
															0	0	0	0	0	
															0	0	0	0	0	
															1	5	0	1	4	
															0	0	0	0	0	Результат
															0	0	0	0	0	
Итого:		150427,26074000																		

Рисунок 33 — Операция умножения с числами 64,22 и 833,5 в 8-ой СС

Умножение в 10-ой СС -> 833,5*64,22																				Обозначения
Целая часть															Дробная часть					Множимое
															8	3	3	5	0	
															6	4	2	2	0	Множитель
															0	0	0	0	0	Процесс перемножения
															0	0	0	0	0	
															0	0	0	0	0	
															16	6	6	10	0	
															16	6	6	10	0	
															32	12	12	20	0	
															48	18	18	30	0	
															0	0	0	0	0	
															0	0	0	0	0	
															0	0	0	0	0	
															0	0	0	0	0	Результат
															0	0	0	0	0	
Итого:		153527,37000000																		

Рисунок 34 — Операция умножения с числами 64,22 и 833,5 в 10-ой СС

Умножение в 16-ой СС -> 833,5*64,22															Обозначения	
Целая часть										Дробная часть					Множимое	
										3	4	1	8	0	0	0
											4	0	3	8	5	1
										3	4	1	8	0	0	0
										F	14	5	28	0	0	0
										18	20	8	40	0	0	0
										9	C	3	18	0	0	0
										0	0	0	0	0	0	0
										C	10	4	20	0	0	0
Целая часть										Дробная часть					Процесс перемножения	
										D	1	1	7	5	B	B
Итого:										D117,5B898000					Результат	

Рисунок 35 — Операция умножения с числами 64,22 и 833,5 в 16-ой СС

4. $355,71_{10} / 4_{10}$:

Десятичная СС	355,71
Двоичная СС	101100011,1011
Восьмеричная СС	543,5534
Шестнадцатеричная СС	163,85C2

Рисунок 36 — Соответствие числа 355,71 в разных СС

Десятичная СС	4
Двоичная СС	100,0000
Восьмеричная СС	4,0000
Шестнадцатеричная СС	4,0000

Рисунок 37 — Соответствие числа 4 в разных СС

Исходная часть	Остаток от деления	Исходная часть	Остаток от деления	Исходная часть	Остаток от деления
Целая часть		Целая часть		Целая часть	
355	1 ↑	355	3 ↑	355	3 ↑
177	1 ↑	44	4 ↑	22	6 ↑
88	0	5	5 ↑	1	1
44	0	Результат:	543	Результат:	163
22	0	Дробная часть		Дробная часть	
11	1 ↓	0,71	5 ↓	0,71	B ↓
5	1 ↓	0,68	5 ↓	0,36	5 ↓
2	0	0,44	3 ↓	0,76	C ↓
1	1 ↓	0,52	4 ↓	0,16	2 ↓
Результат:	101100011	Результат:	0,5534	Результат:	0,85C2
Дробная часть		Итого:	543,5534	Итого:	163,85C2
0,71	1 ↓				
0,42	0				
0,84	1 ↓				
0,68	1 ↓				
Результат:	0,1011				
Итого:	101100011,1011				

Рисунок 38 — Процесс перевода числа 355,71 в разные СС

Деление в 10-ой СС -> 355,81 / 4																Обозначения	
Целая часть								Дробная часть									
							3	5	5	8	1	0	0				Делимое
									4	0	0	0	0				Делитель
							3	5	5	1							Процесс деления
							3	2	0	0							
								3	8	1	0						
								3	6	0	0						
									2	1	0	0					
									2	0	0	0					
										1	0	0	0				
											8	0	0				
											2	0	0	0			
											2	0	0	0			
														0			
														0			
														0			
								8	8	9	5	2	5				Результат
Итог:				88,9525													

Рисунок 42 — Операция деления с числами 355,71 и 4 в 10-ой СС

Деление в 16-ой СС -> 355,81 / 4																Обозначения	
Целая часть								Дробная часть								Делимое	
							1	6	3	В	5	С	2			Делитель	
									4	0	0	0	0				
							1	6	3	В	5	С				Процесс деления	
							1	4	0	0	0	0					
								2	3	В	5	С	2				
								2	0	0	0	0	0				
									3	В	5	С	2	0			
									3	8	0	0	0	0			
										3	5	С	2	0	0		
										3	4	0	0	0	0		
											1	С	2	0	0		0
											1	С	0	0	0		0
													2	0	0	0	0
								5	8	Е	Д	7	0			Результат	
Итог:				58,ED70													

Рисунок 43 — Операция деления с числами 355,71 и 4 в 16-ой СС

2. Анализ современные системы программирования

2.1. Понятие системы программирования, основные компоненты

В рамках проведения анализа в первую очередь следует дать определение выбранному объекту для исследования и рассмотреть его основные компоненты.

Система программирования (СП) — это комплекс программных средств для разработки новых программ на конкретном языке программирования; программные инструменты, которые сопровождают весь технологический цикл создания программного продукта. Системы программирования позволяют объединить различные аспекты написания компьютерной программы, автоматизируют, облегчают и ускоряют разработку компьютерных программ, обеспечивая программистов мощными и удобными средствами разработки программ: редактор текста, язык программирования, транслятор, компоновщик, отладчик [1].



Рисунок 44 — Основные компоненты системы программирования

По набору входных языков различают системы программирования одноязыковые и многоязыковые. Отличительная черта многоязыковых систем состоит в том, что отдельные части программы можно составлять на разных

языках и с помощью специальных обрабатывающих программ объединять их в готовую для исполнения на ЭВМ программу.

Следует отметить, что любая система программирования способна работать только в определённой операционной системе (ОС), под которую она была разработана. При этом среда позволяет разрабатывать программное обеспечение под другие ОС [2].

Ядром любой системы программирования выступает язык программирования (ЯП) — формальный язык, предназначенный для записи компьютерных программ, которые представляют собой набор правил, позволяющих компьютеру выполнить тот или иной вычислительный процесс, организовать управление различными объектами. Он определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (персональный компьютер) под её управлением [3, 4].

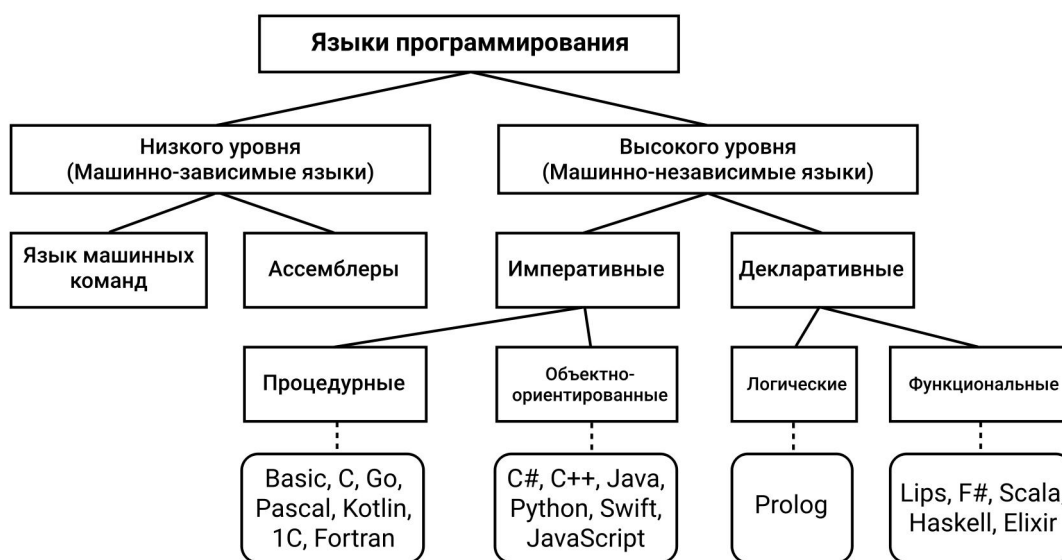


Рисунок 45 — Классификация языков программирования

Низкоуровневые языки программирования — это язык программирования, который обеспечивает небольшую абстракцию или не обеспечивает никакой абстракции от архитектуры набора команд компьютера - команд или функций в

языковой карте, которые структурно похожи на инструкции процессора. Обычно это относится либо к машинному коду (состоящий из инструкций машинного языка, который используется для управления центральным процессором (CPU) компьютера), либо к языку ассемблера (систему обозначений, используемую для представления в удобно читаемой форме программ, записанных в машинном коде; синтаксис языка определяется главным образом системой инструкций конкретного процессора и системой директив конкретного транслятора).

Высокоуровневый язык программирования — язык программирования, разработанный для быстроты и удобства использования программистом, основной чертой которых является абстракция — введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на машинном коде (или другом низкоуровневом языке программирования) очень длинны и сложны для понимания. Языки высокого уровня позволяют программистам писать инструкции на языке, который легче понять, чем языки низкого уровня, были разработаны для платформенной независимости сути алгоритмов, в них не учитываются особенности конкретных компьютерных архитектур. Программы, составляемые на таких языках, представляют собой последовательность структурированных операторов, описывающие действия, которые система должна выполнять после трансляции. Примеры высокоуровневых ЯП: C++, C#, Java, JavaScript, PHP, Python, Pascal.

После того, как был написан исходный код программы на ЯП высокого уровня необходимо осуществлять перевод программы с выбранного языка программирования в выходную (объектную) программу на языке машинных команд, и для этого используют специальную программу-переводчик — транслятор.

Транслятор — технический инструмент, производит анализ транслируемую программу — определяет, содержит ли она синтаксические ошибки и после выполняет преобразование программы, представленной на

одном из языков программирования, в программу на другом языке. Программа, подготовленная на каком-либо языке программирования, называется исходным модулем. В качестве входных данных трансляторы применяют исходные модули и формируют в результате своей работы объектные модули, являющиеся входной информацией для компоновщика. Объектный модуль содержит текст программы на машинном языке и дополнительную информацию, обеспечивающую настройку модуля по месту его загрузки и объединение этого модуля с другими независимо оттранслированными модулями в единую программу.

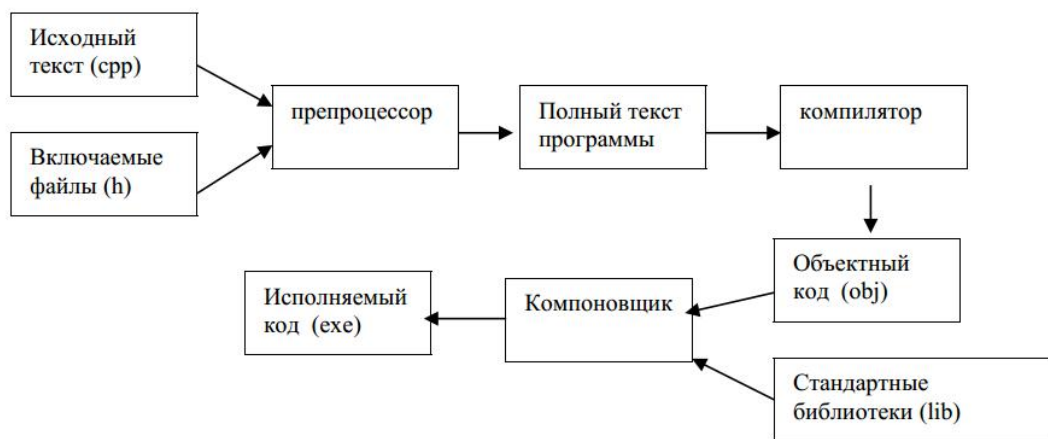


Рисунок 46 — Схема процесса трансляции программы

Процесс трансляции кода включает в себя следующие этапы:

1. Лексический анализ — на этом этапе последовательность символов исходного файла преобразуется в последовательность лексем.
2. Синтаксический анализ — последовательность лексем преобразуется в древо разбора.
3. Оптимизация — выполняется удаление излишних конструкций и упрощение кода с сохранением его смысла.
4. Генерация кода — из промежуточного представления порождается код на целевом машинно-ориентированном языке.

Различают два вида трансляторов: компилятор и транслятор.

1. Компилятор — программа, предназначенная для трансляции исходного текста программы с языка программирования в объектный код — машинный язык. Компилятор читает всю программу целиком переводит весь исходный модуль и создаёт законченный вариант программы на машинном языке, который затем и выполняется. Виды компиляции:

— *Пакетная*. Компиляция нескольких исходных модулей в одном задании.

— *Построчная*. Машинный код порождается и затем исполняется для каждой завершённой грамматической конструкции языка.

— *Условная*. На фазе трансляции результат трансляции зависит от условий, прописанных в исходном тексте программы директивами компилятора.

К компилируемым ЯП относят: C, C++, Delphi, Go, ASM (язык ассемблера), Swift, Pascal, Rust и др.

2. Интерпретатор — программа, выполняющая построчный анализ, обработку и выполнение исходного кода программы или запроса. Каждый раз, когда интерпретатор получает на выполнение код языка высокого уровня, то перед его конвертацией в машинный код, он преобразовывает этот код в промежуточный язык. Части кода последовательно интерпретируются и выполняются отдельно; при нахождении ошибок в составляющих кода процесс интерпретации останавливается. У интерпретаторов есть основной недостаток — низкая скорость работы интерпретируемых программ. К интерпретируемым ЯП относят: JavaScript (JS), Python, Php, Lisp, Ruby, Basic.

Компоновщик (Linker, Link editor) — системная обрабатывающая программа, редактирующая и объединяющая объектные (ранее оттранслированные) модули в единые загрузочные, готовые к выполнению программные модули (загрузочный модуль может быть помещён ОС в основную память и выполнен).

Отладчик — позволяет управлять процессом исполнения программы, является инструментом для поиска и исправления ошибок в программе. Базовый набор функций отладчика включает:

1. пошаговое выполнение программы с отображением результатов
2. остановка в заранее определённых точках,
3. остановка в некотором месте программы при выполнении условия
4. изображение и изменение значений переменных,

Редактор исходного кода — текстовый редактор для создания и редактирования исходного кода программ, имеет некоторые возможности, упрощающие и ускоряющие написание и изменение кода, такие как подсветка синтаксиса (выделение однотипных конструкций отдельным цветом), автодополнение (анализирует написанный код и предлагает варианты продолжения), проверка правильности расстановки скобок, отступы, контекстная помощь по коду, разделение рабочей области на два и более секторов и многие другие. Редактор может быть встроенным в среду разработки ПО или выступать в качестве отдельного приложения (Visual Studio Code, Atom, Brackets). Такие редакторы дают разработчику возможность для запуска дополнительных компонентов разработки: транслятора, отладчика и др.

2.2. Современные системы/среды программирования

Современные системы программирования представляют пользователям широкие возможности и удобные программные средства разработки компьютерных программ и являются в основном интегрированными средами разработки приложений (IDE, Integrated Development Environment), ориентированными на определённый входной язык программирования.

2.3. Понятие интегрированной среды разработки ПО

Интегрированная среда разработки (IDE, Integrated Development Environment) — это программное приложение, которое предоставляет программистам совокупность возможностей для разработки программного обеспечения, обычно состоят из редактора исходного кода, средств

автоматизации сборки и отладчика. Некоторые IDE содержат необходимый компилятор, интерпретатор или всё вместе. Термин «интегрированная среда разработки» означает, что предоставляется все необходимое для превращения исходного кода программы в функционирующие приложения. Объединение различных утилит в одном модуле позволяет программисту сосредоточиться на решении алгоритмической задачи и избежать потери времени при совершении аналогичных технических действий [5].

Многие современные IDE также имеют обозреватель классов, инспектор объектов и диаграмму иерархии классов для использования при разработке объектно-ориентированного программного обеспечения. Иногда интегрируются система контроля версий (GIT — система управления версиями, записывающая изменения в течение времени разработки и позволяющая вернуться позже к определённой версии) или различные инструменты для упрощения построения графического пользовательского интерфейса (GUI).

Виды IDE: облачные (онлайн) — взаимодействие с которыми происходит посредством веб-браузера; классические, устанавливаемые, поставляемые в программных пакетах (Desktop).

2.4. Microsoft Visual Studio

Microsoft Visual Studio — это интегрированная среда разработки (IDE) от Microsoft, позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, а также веб-сайтов, веб-приложений, веб-сервисов и мобильных приложений [6, 7].

Visual Studio включает редактор кода, поддерживающий IntelliSense (компонент завершения кода), а также рефакторинг кода. Интегрированный отладчик работает как отладчик на уровне исходного кода, так и как отладчик на уровне компьютера. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения,

веб-редактор, дизайнер классов и дизайнер схемы базы данных. Среда позволяет создавать и подключать плагины, которые расширяют функциональность: добавление поддержки систем управления версиями (таких как Subversion и Git) и новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

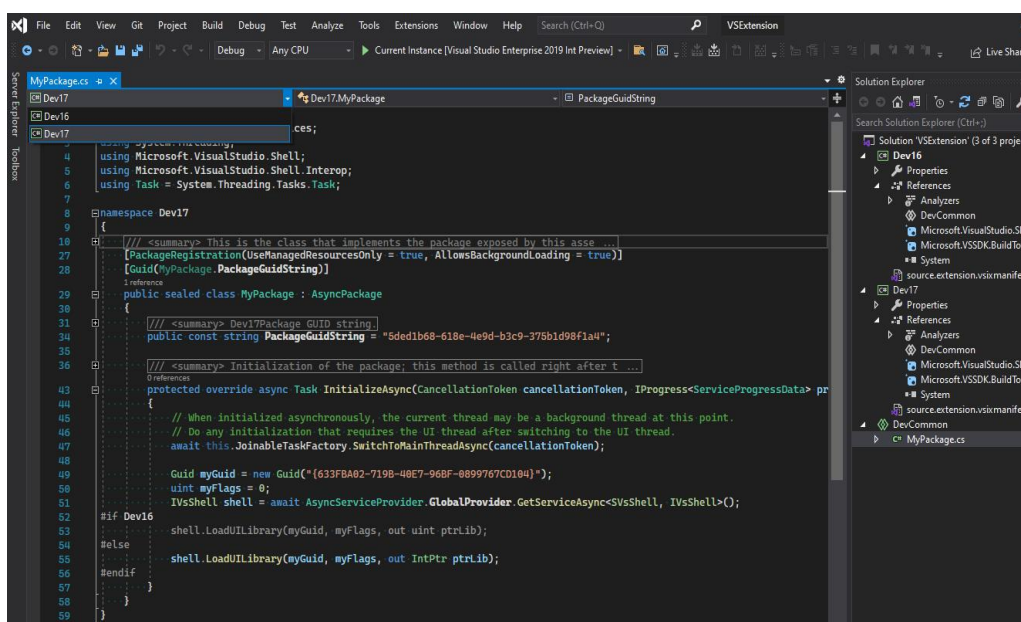


Рисунок 47 — Графический интерфейс системы Visual Studio

Visual Studio не поддерживает какой-либо язык программирования, решение или инструмент по своей сути, вместо этого он позволяет подключать функциональные возможности, закодированные как VSPackage. IDE предоставляет три службы: SVsSolution, которая предоставляет возможность перечисления проектов и решений; SVsUIShell, который предоставляет функциональные возможности окон и пользовательского интерфейса (включая вкладки, панели инструментов и окна инструментов); и SVsShell, который занимается регистрацией пакетов. Все компоненты и инструменты реализованы в виде VSPackages. Пакет SDK (набор средств для разработки) Visual Studio

также включает платформу управляемых пакетов (MPF), которая представляет собой набор управляемых оболочек вокруг COM-интерфейсов (взаимодействие моделей компонентного объекта), позволяющих писать пакеты на любом языке, совместимом с CLI (спецификация общезыковой инфраструктуры).

Поддержка языков программирования добавляется с помощью определённого пакета VSPackage, называемого языковой службой. Языковая служба определяет различные интерфейсы, которые VSPackage может реализовать для добавления поддержки различных функциональных возможностей. Функциональные возможности, которые могут быть добавлены таким образом, включают синтаксическую окраску, завершение инструкций, сопоставление фигурных скобок, подсказки с информацией о параметрах, списки элементов и маркеры ошибок для фоновой компиляции. Поддерживаемые ЯП: JavaScript, AJAX, Visual Basic, Visual C#, Visual C++, XAML, Visual F#, ASP.NET, DHTML и др.

Visual Studio поддерживает запуск нескольких экземпляров среды (каждый со своим собственным набором пакетов VSPackages). Экземпляры используют разные системные реестры для хранения состояния конфигурации и различаются по их AppId (идентификатор приложения). Экземпляры запускаются специализированным исполняемым файлом, который выбирает AppId, задаёт корневой реестр и запускает IDE. Пакеты VSPackage, зарегистрированные для одного AppId, интегрируются с другими пакетами VSPackages для этого AppId. Продукты выпуска Visual Studio Express устанавливаются с собственными AppId, но продукты Standard, Professional и Team Suite используют один и тот же AppId.

Visual Studio включает редактор кода, который поддерживает подсветку синтаксиса, разделение окна среды и автодополнения кода с помощью IntelliSense для переменных, функций, методов, циклов и запросов LINQ (предоставляет простой и удобный язык запросов к источнику данных). IntelliSense поддерживается для включённых в среду языков, а также для XML, CSS (каскадных таблиц стилей) и JavaScript при разработке веб-сайтов и веб-

приложений. Варианты автодополнения отображаются в списке над окном редактора кода, в непосредственной близости от курсора редактирования.

IntelliSense — набор возможностей, отображающих сведения о коде непосредственно в редакторе и в некоторых случаях автоматически создающих небольшие отрывки кода. По сути, это встроенная в редактор базовая документация, которая избавляет от необходимости искать информацию в других источниках. Функции IntelliSense зависят от языка.

По умолчанию Visual Studio форматирует код по мере его ввода, автоматически вставляя необходимые отступы и применяя цветовое кодирование для выделения элементов типа комментариев. Такие незначительные отличия делают код более удобным для чтения и менее подверженным ошибкам.

Навигационные средства включают сворачивание блоков кода, текстовый поиск и замена (которая позволяет отыскивать ключевые слова как в одном файле, так и во всем проекте) и поиск по регулярному выражению. Также встроены инструменты управления фрагментами кода. Редактор кода Visual Studio также поддерживает настройку закладок в коде для быстрой навигации и рефакторинг кода — включает в себя такие операции, как интеллектуальное переименование переменных, извлечение одной или нескольких строк кода в новый метод и изменение порядка параметров методов, автоматическое добавление и удаление комментариев.

Среда имеет инструмент для совместной разработки Live Share — предоставляет возможности совместного редактирования и отладки в реальном времени независимо от типа приложения или языка. Вы можете мгновенно предоставлять общий доступ к своему проекту с поддержкой высокого уровня безопасности. Кроме того, вы можете предоставлять общий доступ к сеансам, экземплярам терминала, веб-приложениям на локальном компьютере, голосовым звонкам и т.п.

Visual Studio имеет фоновую компиляцию — мере написания кода Visual

Studio компилирует его в фоновом режиме, чтобы обеспечить обратную связь об ошибках синтаксиса и компиляции, которые помечаются красным волнистым подчёркиванием, предупреждения помечаются зелёным подчёркиванием. Фоновая компиляция не генерирует исполняемый код, поскольку для неё требуется компилятор, отличный от того, который используется для создания исполняемого кода. Если навести указатель мыши на волнистую линию, на экран будут выведены дополнительные сведения об ошибке. Также в поле слева может отображаться лампочка, указывающая на наличие сведений о «быстрых действиях» для устранения ошибки.

Предлагаемые в Visual Studio инструменты отладки являются наилучшим средством для отслеживания ошибок и диагностирования странного поведения программы. Среда включает отладчик, который может использоваться для отладки приложений, написанных на любом языке, поддерживаемом Visual Studio. Кроме того, он также может подключаться к запущенным процессам, отслеживать и отлаживать их. Отладчик Visual Studio также может создавать дампы памяти (содержимое рабочей памяти процесса), а также загружать их позже для отладки. Также поддерживаются многопоточные программы, имеется возможность просмотра статистики мониторинга производительности в режиме реального времени. Отладчик Visual Studio позволяет устанавливать точки останова (которые позволяют временно останавливать выполнение программы в определённом положении) и контрольные значения (отслеживать значения переменных по мере выполнения). Точки останова могут быть условными, то есть они срабатывают при выполнении условия. Отладчик позволяет редактировать код по мере его отладки. При отладке, если указатель мыши наведён на какую-либо переменную, её текущее значение отображается во всплывающей подсказке, где при желании его также можно изменить.

Visual Studio имеет интегрированный Web-сервер, который позволяет запускать Web-сайт прямо из среды проектирования, а также повышает безопасность, исключая вероятность получения доступа к тестовому Web-сайту с какого-нибудь внешнего компьютера, поскольку тестовый сервер может

принимать соединения только с локального компьютера. Среда позволяет создавать Web-страницы на разных языках, но помещать их все в одно и то же Web-приложение.

Visual Studio включает множество визуальных конструкторов, помогающих в разработке приложений:

— Windows Forms Designer — конструктор Windows Forms используется для создания приложений с графическим интерфейсом. Компоновкой можно управлять, размещая элементы управления внутри контейнеров. Пользовательский интерфейс связан с кодом с помощью модели программирования, управляемой событиями. Конструктор создаёт код C# или VB.NET для приложения.

— WPF Designer — используется для создания пользовательских интерфейсов, ориентированных на Windows Presentation Foundation (аналог WinForms, система для построения клиентских приложений Windows). поддерживает привязку данных и автоматическое управление макетом. Он создаёт код XAML для пользовательского интерфейса.

— Web designer — включает редактор и конструктор веб-сайтов, который позволяет создавать веб-страницы путем перетаскивания виджетов. Он используется для разработки ASP.NET (платформа разработки веб-приложений) и поддерживает HTML, CSS и JavaScript.

Дополнительные инструменты среды:

— Браузер открытых вкладок — используется для перечисления всех открытых вкладок и переключения между ними.

— Редактор свойств — используется для редактирования свойств в области графического интерфейса, в нем перечислены все доступные свойства для всех объектов, включая классы, формы, веб-страницы и другие элементы.

— Обозреватель решений — набор файлов кода и других ресурсов, используемых для построения приложения, используется для управления файлами в решении и их просмотра.

— Team Explorer — используется для интеграции возможностей Azure DevOps, предоставляет возможность просматривать и управлять отдельными рабочими элементами (включая пользовательские истории, ошибки, задачи и другие документы).

Существует три выпуска Visual Studio: Community (бесплатная версия), Professional и Enterprise (стоимость: от 45\$ в месяц).

Достоинства: Понятный интерфейс среды разработки, удобность, автоматическое обнаружение ошибок в коде.

В качестве недостатка можно отметить, что для открытия и запуска приложений требуются значительные ресурсы. Поэтому на некоторых устройствах внесение простых изменений может занять много времени. Для простых задач целесообразно использовать более лёгкий редактор.

Системные требования: двухъядерный процессор с частотой 1,8 ГГц; 2ГБ ОЗУ (рекомендуется 8ГБ ОЗУ); до 210ГБ (минимум 800МБ) свободного места на жёстком диске (в зависимости от установленных компонентов); Windows 8.1 и поздние версии ОС, macOS 10.13 и позднее.

2.5. IntelliJ IDEA

IntelliJ IDEA — интегрированная среда разработки ПО для многих языков программирования, разработана и поддерживается компанией JetBrains (международная компания, которая разрабатывает инструменты для разработки), обладает обширным инструментарием для разработки мобильных приложений и корпоративных технологий для различных платформ [8-9].

Дизайн среды ориентирован на продуктивность работы программистов, позволяя сконцентрироваться на функциональных задачах. Среда доступна в двух редакциях: Community Edition — является полностью свободной версией, доступной под лицензией Apache 2.0 (лицензия на свободное программное обеспечение Apache Software Foundation); Ultimate Edition — доступной под

коммерческой лицензией, реализована поддержка Java EE (платформа для создания корпоративных приложений на языке Java, веб-приложения и веб-сервисы), UML-диаграмм (благодаря им складывается общая картина, дающая представление о возможностях выражения архитектурных идей в рамках бизнес-задач), подсчёт покрытия кода, а также поддержка других систем управления версиями, языков и фреймворков.

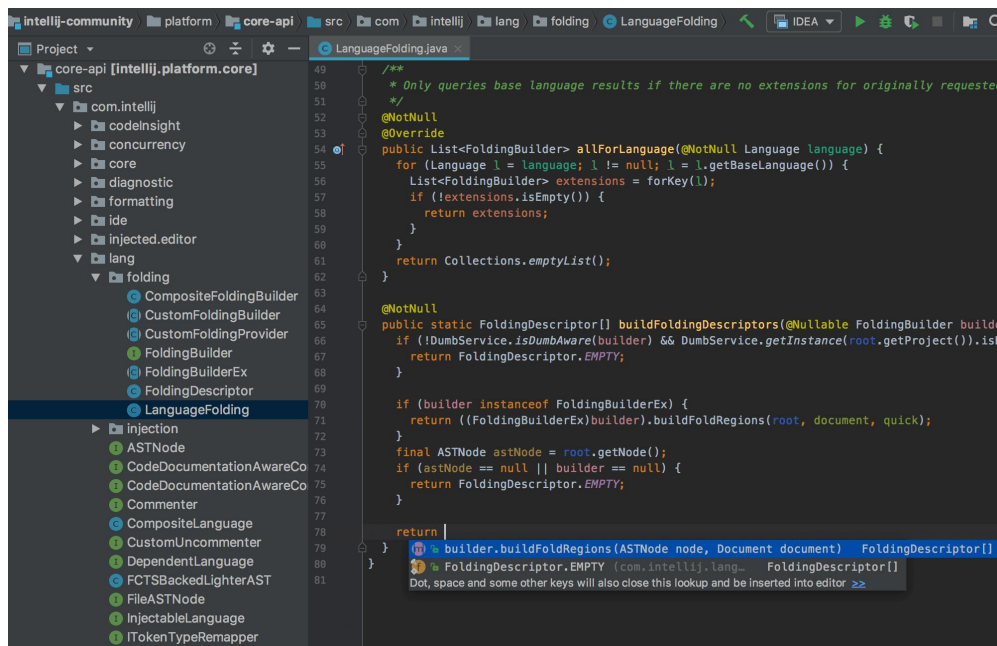


Рисунок 48 — Графический интерфейс среды IntelliJ IDEA

Первая версия IntelliJ IDEA была выпущена в январе 2001 года и была одной из первых доступных Java сред разработки с расширенными возможностями навигации по коду и рефакторинга кода. В декабре 2014 года Google анонсировала версию 1.0 Android Studio, IDE с открытым исходным кодом для приложений Android, основанную на IntelliJ IDEA. Другие среды разработки, основанные на фреймворке IntelliJ, включают AppCode, CLion, DataGrip, GoLand, PhpStorm, PyCharm, Rider, RubyMine, WebStorm и MPS.

IDE включает в себя определённые возможности для удобного редактирования исходного кода программы — автодополнение кода, навигация по проекту, рефакторинг кода, отладка кода.

Диалоговое окно глобального поиска Search Everywhere даёт возможность найти в кодовой базе все что угодно, например классы или символы. В IntelliJ IDEA очень легко проверить, где и как определены в проекте разные символы, такие как теги, классы, поля, методы или функции.

IntelliJ IDEA предлагает сервис совместной разработки и парного программирования: Code With Me. С его помощью вы можете в реальном времени работать над проектом, открытым в вашей IDE, вместе с коллегами. Кроме того, Code With Me поддерживает видео- и голосовые вызовы из IDE, позволяя организовать встречи как с глазу на глаз, так и с участием десятков людей.

Функция автодополнения кода в IntelliJ IDEA работает с учётом контекста, предлагая только варианты, действительные для текущего положения курсора. Вам не нужны никакие сочетания клавиш и дополнительные настройки: функция автодополнения запускается, как только вы начинаете писать код в редакторе. IntelliJ IDEA использует машинное обучение, чтобы самое вероятное предложение оказалось первым в списке.

Рефакторинг проекта — даёт возможность безопасного и эффективного обновления кода. Он становится проще, понятнее, и его легче поддерживать. IDE предлагает широкий выбор рефакторингов для безопасного переименования элементов кода, изменения сигнатуры класса или метода, извлечения фрагмента кода в метод и введения переменных. Шаблоны Live template — удобная функция, позволяющая писать код быстрее, для этого необходимо просто ввести заданное заранее в шаблоне сокращение, чтобы подставить в код стандартную конструкцию. Есть возможность создавать собственные шаблоны и использовать их для вставки регулярно повторяющихся сниппетов кода.

Среда имеет встроенный отладчик позволяет отлаживать приложение в самой IDE, делая процесс разработки и отладки беспроблемным. Система позволяет с лёгкостью выполнять модульное тестирование. Для подготовки к отладке необходимо создать точки останова, в которых выполнение программы

будет прерываться, чтобы можно было проанализировать её состояние и поведение. Во время отладки IDE показывает значения переменных прямо в исходном коде рядом с их использованиями, и вы можете изменить эти значения. Можно также использовать действие Smart Step-Into, чтобы выбрать вызов метода, для которого нужно выполнить отладку.

IntelliJ IDEA поддерживает самые популярные фреймворки тестирования, например JUnit, TestNG, Cucumber, Arquillian JUnit и Selenium (плагины доступны в Marketplace). IDE предлагает среду и набор инструментов для разработки и запуска эффективных и понятных автоматических тестов. Можно использовать сочетание клавиш для переключения между классами тестов и исходным кодом, запускать несколько тестов, просматривать статистику для каждого теста и т. д. Для аналитики можно использовать функцию Code Coverage, которая показывает процент кода, покрытый юнит-тестами.

IDE поставляется со встроенным декомпилятором, позволяющий преобразовать двоичный код в удобный для восприятия код Java, чтобы выполнить его отладку. Например, если вы открываете библиотеку в файле .jar, не имея исходного кода, IDE отобразит содержимое в удобном для чтения виде. При желании в декомпилированном коде можно создать точки останова и выполнить его отладку.

IntelliJ IDEA обладает широкими возможностями проверки качества и валидности кода с помощью инспекций — помогают быстрее писать код в соответствии с самыми строгими стандартами качества и чувствовать себя уверенно на протяжении всего процесса разработки. Если IntelliJ IDEA обнаруживает ошибки, она предлагает устранить их с помощью контекстных действий, включающих в себя быстрые исправления подсвеченных ошибок.

Среда обеспечивает интеграцию с инструментами сборки/упаковки, такими как Maven, Gradle, Ant, Gant, SBT, NPM, Webpack, Grunt и Gulp.

IntelliJ IDEA обеспечивает интеграцию с Git, позволяя клонировать проекты из Git, записывать и отправлять изменения, работать с несколькими ветками, управлять списками изменений и проверять изменения перед

отправкой.

В среду встроена функция Local History — отличный инструмент для отслеживания внесённых изменений. Эта функция позволяет также вернуться к любому состоянию проекта, отменить изменения и восстановить случайно удалённые файлы. Подробнее о Local history можно прочитать в нашем

IDE предоставляет инструменты для работы с базами данных и SQL файлами, которые позволяют подключаться к действующим базам данных; выполнять запросы; просматривать и обновлять данные; и даже управлять своими схемами в визуальном интерфейсе из самой IDE.

Встроенный терминал IntelliJ IDEA позволяет работать с оболочкой командной строки (PowerShell или Bash) прямо из IDE, не переходя в отдельное приложение терминала. Он поддерживает все те же команды, что и операционная система.

IntelliJ поддерживает основные серверы приложений: Tomcat, JBoss, WebSphere, WebLogic, Glassfish и многие другие. При разработке веб-сервисов их обычно нужно протестировать, отправляя и получая HTTP-запросы. Такие запросы хранятся в файлах .http или .rest, которые можно создать, отредактировать и выполнить с помощью HTTP-клиента. Инструмент встроен в редактор IntelliJ IDEA и предлагает помощь при создании HTTP-запросов, в том числе автодополнение кода, подсветку, рефакторинги и т. п.

Через отдельный плагин IntelliJ предоставляет специальное окно инструментов, которое позволяет подключаться к локально работающим машинам Docker (ПО для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации — метод виртуализации, при котором ядро операционной системы поддерживает несколько изолированных экземпляров пространства пользователя вместо одного).

Система поддерживает плагины, с помощью которых можно добавить дополнительные функции в IDE. Подключаемые модули можно загрузить и установить либо с веб-сайта репозитория плагинов IntelliJ, либо с помощью встроенной функции поиска и установки подключаемых модулей IDE. Каждая

редакция имеет отдельные репозитории плагинов

IntelliJ поддерживает различные ЯП в зависимости от выпуска IDE:

— Поддерживается как в Community, так и в Ultimate Edition: Java, Lua, Kotlin, Python, Rust, Scala, Perl, Dart, Clojure и др.

— Поддерживается только в Ultimate Edition: Go, JavaScript, SQL, TypeScript, PHP, HTML/XHTML/CSS, Ruby, CoffeeScript, ActionScript.

Системные требования: двухъядерный процессор с частотой 1,8 ГГц; 1ГБ ОЗУ (рекомендуется 2ГБ ОЗУ); 300Мб + 1Гб (для кэша) свободного места на жёстком диске; Windows 10/8/7 (64-bit включительно), macOS 10.5 и выше.

Недостатки: среда требует времени и усилий для изучения, поэтому может оказаться не лучшим решением для начинающих разработчиков. Имеется много сочетаний горячих клавиш, которые необходимо помнить.

2.6. Eclipse

Eclipse — свободная интегрированная среда разработки модульных кросс-платформенных приложений с открытым исходным кодом, развивается и поддерживается Eclipse Foundation, лицензируется Eclipse Public License [10].

Eclipse построен в виде набора расширяемых подсистем, а не как единое монолитное приложение. Eclipse Software Development Kit (SDK) — это программное обеспечение с открытым исходным кодом, выпущенное на условиях Eclipse Public License (несовместимо с GNU General Public License). В стандартную поставку Eclipse SDK входят три подпроекта: Eclipse Platform — основной, базовый проект Eclipse, предоставляет базовые сервисы; Java Development Tools (или JDT) — набор плагинов платформы Eclipse для организации Java IDE (имеет встроенный компилятор), позволяет разрабатывать приложения Java; Plug-in Developer Environment (или PDE) — инструмент для разработки, отладки тестирования и установки Eclipse плагинов.

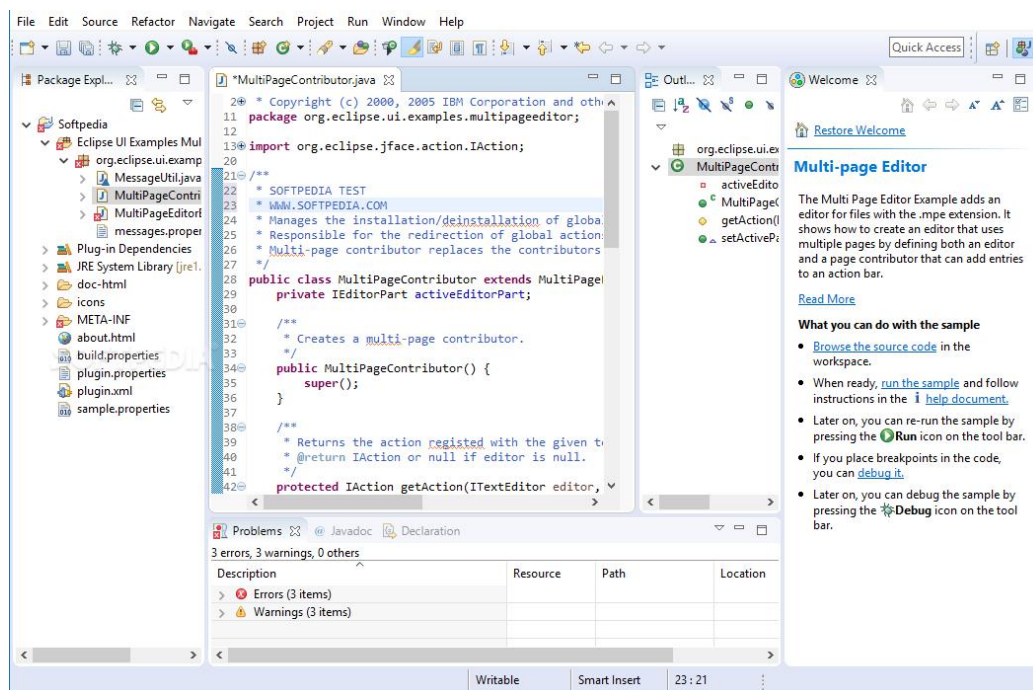


Рисунок 49 — Графический пользовательский интерфейс среды Eclipse

Eclipse JDT (Java Development Tools) — наиболее известный модуль, нацеленный на групповую разработку: среда интегрирована с системами управления версиями — CVS, GIT в основной поставке, для других систем (например, Subversion, MS SourceSafe) существуют плагины. Также предлагает поддержку связи между IDE и системой управления задачами (ошибками).

Platform — это ядро Eclipse. Сама по себе эта подсистема не содержит особенно полезной для пользователей функциональности, но без нее невозможна работа остальных подпроектов Eclipse. Сервисы, которые обеспечивает Platform, позволяют разработчикам определять видимые пользователю артефакты, создавать пользовательские интерфейсы, работать с системами контроля версий (GIT), средами отладки и справочной системой. Соответствующими компонентами Platform, реализующими эти сервисы, являются Workspace (управление контентом), Workbench (базовый пользовательский интерфейс Eclipse), Team, Debug и Help.

Workspace — это рабочая область, всегда существует в единственном экземпляре, но можно переключаться между несколькими рабочими областями,

но только одна из них может быть открыта в конкретный момент времени. Рабочая область представляет собой контейнер для проектов, в ней содержатся настройки, сохраняемые компонентами Eclipse, и вспомогательные данные. Рабочая область представляет собой директорию, поддиректории которой соответствуют проектам, а директория папки `.metadata` используется как хранилище настроек и данных. Каждый из проектов, содержащихся в рабочей области, объединяет набор ресурсов (файлов и папок), правил для их обработки и других свойств проекта (таких, как настройки компилятора). Проекты могут зависеть друг от друга. Для ресурсов, представляющих файлы, сохраняется история последних изменений их содержимого, таким образом, пользователь имеет возможность просматривать сделанные им изменения, сравнивать между собой различные версии файлов и в любой момент вернуться к одной из них. Ресурсы могут быть локальными или находиться в репозитории системы контроля версий.

Eclipse Runtime — исполняющая среда, в которой выполняются коды расширений и модулей, обеспечивает всю базовую функциональность платформы — управление расширениями и обновлениями, взаимодействие с операционной системой, обеспечение работы системы помощи; основана на Equinox (среда выполнения модуля, которая позволяет разработчикам реализовать приложение в виде набора «пакетов» с использованием инфраструктуры общих служб), реализации спецификации ядра OSGi (стандартная среда поставки компонентов, формирует модель для построения приложения из узлов, связанных вместе посредством сервисов).

IDE включает редактор кода, который поддерживает подсветку синтаксиса, разделение окна среды и автодополнения кода для программных конструкций (варианты дополнения отображаются рядом с текстовым курсором). Пользователь может создать экземпляр редактора для существующего файла, внести необходимые изменения, сохранить их и закрыть редактор. Изменения, сделанные в процессе редактирования, остаются в буфере и не записываются в открытый файл до тех пор, пока пользователь не

выберет команду сохранения. В редакторе, созданном для одного файла, невозможно открыть какой-либо другой файл.

IDE отвечает за управление основными элементами программы, их расположением и настройками, управление проектами, отладку и сборку проектов, поиск по файлам и командную разработку. В среде реализована автоматическая инкрементальная компиляция проекта после каждого изменения, сделанного разработчиком — при сохранении файла с исходным кодом инкрементальный сборщик анализирует сделанные изменения и запускает компиляцию не только сохранённого файла, но и всех зависящих от него компонентов. При первой компиляции проекта в памяти строится дерево зависимостей файлов в проекте, что позволяет в дальнейшем производить компиляцию очень быстро, поскольку нет необходимости анализировать все исходные файлы в проекте. Информация, собранная в процессе компиляции проекта, также используется для выполнения рефакторинга и поиска без задержек, которые были бы необходимы для её сбора в случае отсутствия автоматической компиляции. (И рефакторинг, и поиск будут работать и для не откомпилированных проектов, однако в этом случае точность полученных результатов не гарантируется.)

GUI среды реализован с использованием платформно-зависимая библиотека SWT (Standard Widget Toolkit), которая использует графические компоненты данной ОС. Библиотека полностью опирается на нижележащую платформу (операционную систему), что обеспечивает быстроту и натуральный внешний вид пользовательского интерфейса, но иногда вызывает на разных платформах проблемы совместимости и устойчивости приложений. Пользовательский интерфейс Eclipse использует JFace (набор Java-классов, реализующий наиболее общие задачи построения GUI), который упрощает построение пользовательского, основанного на SWT.

Для среды Eclipse существует целый ряд свободных и коммерческих модулей. Чтобы расширить возможности среды доступна система установки плагинов, написанные для платформы Eclipse, такие как наборы средств

разработки для других языков программирования (C/C++ - CDT, PHP - PDT, Python - PyDev, JavaScript - JSEclipse), а также могут писать и вносить свои собственные подключаемые модули.

Стоит выделить преимущества IDE: чрезвычайно простая в установке и использовании, бесплатный доступ к среде, доступна для любой платформы, поддерживает всевозможные типы ЯП, возможность подключения к разным базам данных, благодаря разнообразию плагинов тонко настраивается и расширяется дополнительными функциями, настраиваемый графический интерфейс, среда сохраняет и восстанавливает сессии, встроенное юнит-тестирование. Недостатки: новичкам может быть сложно разобраться в многообразии возможностей, существенная нехватка документации, нет единого сообщества разработчиков.

Системные требования: двухъядерный процессор с частотой 1,5 ГГц или больше; 512МБ ОЗУ (рекомендуется 1ГБ и больше ОЗУ); 1Гб свободного места на жёстком диске; Windows 10/8/7 (Win32 и Win64), macOS (x86, x86-64, PPC/ Cocoa), дистрибутив Linux (x86, x86-64).

2.7. Xcode

Xcode — интегрированная среда разработки компании Apple, которая предоставляет разработчикам инструменты для создания ПО под iPhone, iPad, Mac, Apple Watch и Apple TV. Среда распространяется бесплатно, годовая лицензия разработчика для публикации приложения в магазине App Store стоит \$99 (7299руб.).

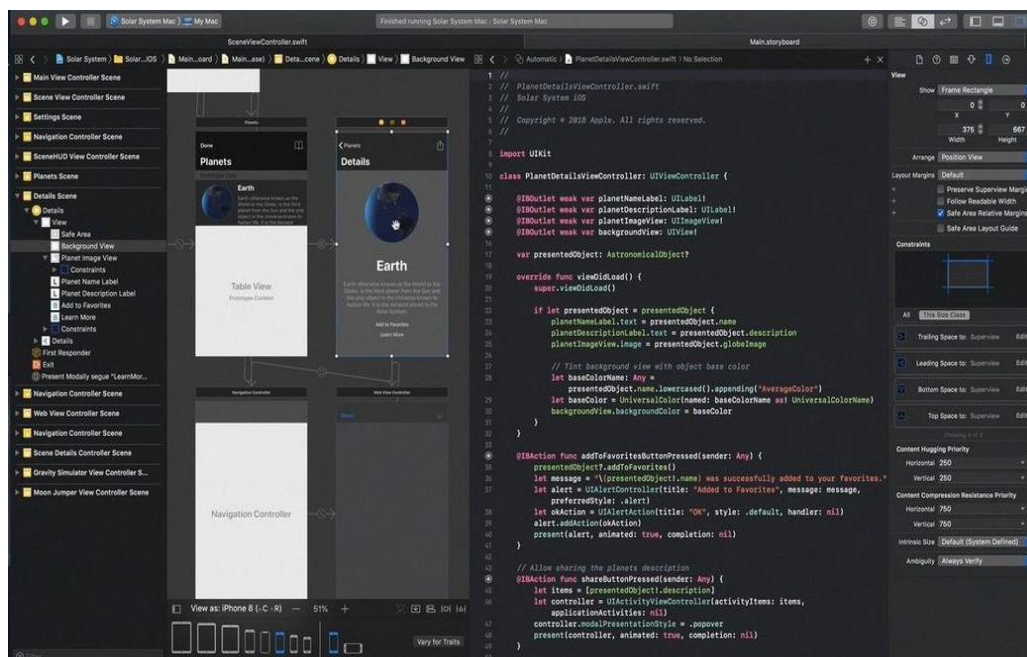


Рисунок 50 — Графический интерфейс среды Xcode

Xcode поддерживает исходный код для языков программирования C, C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, ResEdit (Rez) и Swift с различными моделями программирования, включая, но не ограничиваясь Cocoa, Carbon и Java. Прежде коды писались языком Objective-C, но в июне 2014 Apple представила Swift (универсальный, многопарадигмальный, скомпилированный язык программирования, разработанный Apple для создания мобильных приложений). Допускается присутствие в одном проекте как Swift, так и Objective-C.

IDE может создавать толстые двоичные файлы, содержащие код для нескольких архитектур с исполняемым форматом Mach-O, они называются универсальными двоичными файлами, которые позволяют ПО работать на разных платформах. Используя iOS SDK, tvOS SDK и watchOS SDK, также можно использовать для компиляции и отладки приложений для разных устройств корпорации Apple.

Xcode также интегрирует встроенную поддержку управления исходным кодом с помощью системы и протокола управления версиями GIT, позволяя пользователю создавать и клонировать репозитории GIT, а также фиксировать,

отталкивать и извлекать изменения, автоматизируя задачи, которые традиционно выполнялись бы с помощью Git из командной строки.

Среда разработки Xcode обеспечивает эффективность работы как небольших, так и крупных команд разработки ПО. В Xcode IDE используется схема разделения данных приложения Model-View-Controller (Модель-Представление-Контроллер или MVC) для сегментации каждого слоя приложения.

Редактор кода имеет функцию подсветки программных конструкций в файле с исходным кодом, где были допущены ошибки и, когда возможно, предлагает фиксировать её. XCode позволяет ускорять написание программы при помощи интеллектуального завершения кода, пользоваться готовыми к использованию фрагментами кода и шаблонами исходного файла.

Для удобства создания новых проектов присутствует богатый арсенал готовых шаблонов для разработки под все устройства Apple. Помимо шаблонов для разработки приложений есть выбор для создания различных программных библиотек.

2.8. NetBeans

NetBeans — бесплатная интегрированная среда разработки с открытым исходным кодом, работает на платформах Windows, macOS и Linux. Среда IDE NetBeans предоставляет все средства, необходимые для создания компьютерных, мобильных и веб приложений на языках Java, C/C++ и других поддерживаемых средой ЯП [11].

Все функции IDE обеспечиваются модулями, где каждый предоставляет чётко определённые функции, позволяют расширить среду. Новые функции, такие как поддержка других языков программирования, могут быть добавлены путём установки дополнительных модулей. Платформа предоставляет службы, общие для практически всех крупных приложений рабочей среды: управление

окнами, меню, параметры настройки и хранение данных, диспетчер обновления и доступ к файлам.

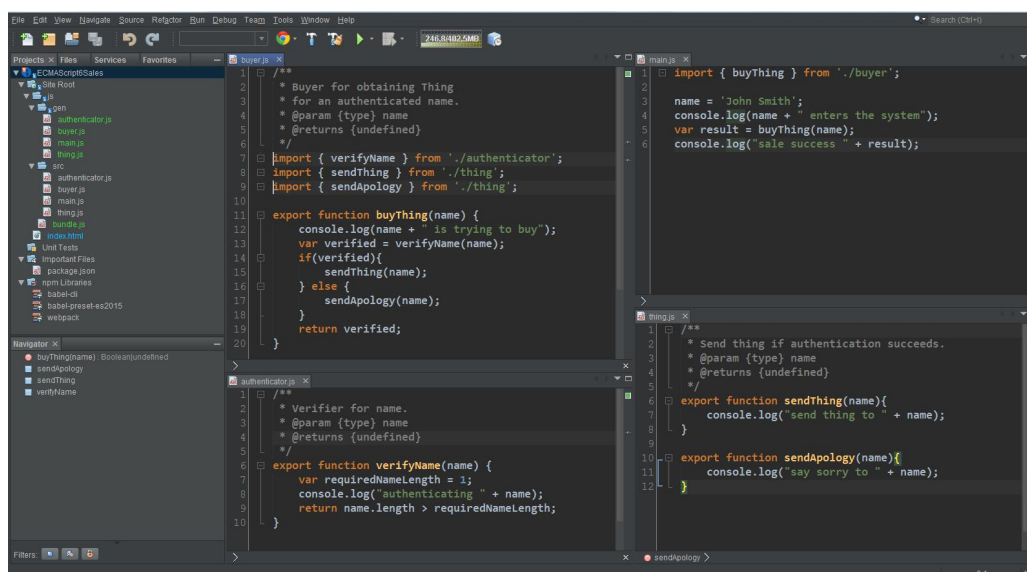


Рисунок 51 — Графический интерфейс среды NetBeans

Рабочая область среды IDE является полностью настраиваемой — существует возможность пользовательской настройки действий, выполняемых с помощью панели, назначения "горячих" клавиш и т.д.

Текстовый редактор NetBeans делает отступы строк, проверяет соответствие скобок и слов, подсвечивает синтаксис исходного кода. Производится проверка ошибок во время ввода, отображение вариантов для автозавершения кода и фрагментов документации по требуемому языку программирования. Редактор может генерировать и вставлять в исходный код стандартные фрагменты кода на Java или других языках.

IDE имеет интегрированный браузер классов, который позволяет просматривать иерархию и структуру любого класса Java — отображаются интерфейсы, базовые классы, производные классы и члены классов.

Существует возможность перемещения любой вкладки редактора в пределах рабочего пространства IDE и за её пределы, создавая независимое окно, которое можно переместить на второй экран. Расширенные средства для

выполнения контекстно-зависимого поиска по всей среде IDE, справочным материалам и всем открытым проектам и файлам.

Разработка Java настольных приложений с профессиональными графическими интерфейсами пользователя, для этого используется визуальный редактор — Swing GUI Builder. Работа осуществляется путём перетаскивания элементов графического интерфейса из палитры на холст. Предварительное позиционирование элементов можно осуществлять с помощью указателя мыши. Панель свойств и инспектор компонентов предоставляют возможность тонкой настройки каждого компонента интерфейса.

Возможность создания, тестирования, отладки и внедрения приложений, функционирующих на мобильных телефонах, карманных компьютерах и встраиваемых системах. Visual Mobile Designer (VMD) создаёт всю необходимую модульную инфраструктуру проекта и обеспечивает быструю разработку графических интерфейсов путём создания в рабочей области компонентов. Возможность создания GUI на основе SVG.

Поддерживается возможность группирования связанных проектов — создавая группы проектов, можно быстро открывать и закрывать несколько сгруппированных проектов одновременно.

Среда имеет расширенные средства для выполнения контекстно-зависимого поиска по всей среде IDE, справочным материалам и всем открытым проектам и файлам. Существует возможность создания проектов в свободном формате или начинать работу с проектом с шаблона. В комплекте со средой IDE поставляются шаблоны и примеры проектов для приложений Java SE, мобильных, веб-приложений и приложений уровня предприятия, приложений JavaFX, подключаемых модулей NetBeans.

Заключение

В результате проделанной работы был получен опыт перевода чисел из одной системы счисления в заданную с использованием табличного редактора Microsoft Excel, а также производить математические операции с числами в различных системах счисления.

В настоящее время системы программирования занимают лидирующие позиции на рынке средств разработки ПО. Они позволяют избежать большого объёма одинаковых операций и тем самым способствуют повышению эффективности процесса разработки и отладки исходного кода программы. Разработчики ПО отдают предпочтение системам, которые являются лёгкими в использовании, позволяют получить эффективные программы, имеют богатые библиотеки функций и мощные возможности для отладки разрабатываемых программ.

К числу возможностей современных интегрированных сред относятся: использование встроенного многофайлового текстового редактора, система сборки бинарных кодов из исходных кодов, отладчик, автоматическая диагностика выявленных при компиляции ошибок, менеджер проектов (предоставляет возможность параллельной работы над несколькими проектами), генератор unit-тестов, сокращение количества перекомпиляции (только редактировавшиеся модули), инструменты поддержки коллективной разработки, инструменты связи с системой управления версиями (GIT), средства создания различных видов проектов и их визуализации, средства расширения функциональности и видов проектов (plug-ins).

Для каждого из языков программирования сегодня имеется немало систем программирования, выпускаемых различными фирмами и ориентированных на различные модели персональных компьютеров и операционные системы.

Библиографический список

1. Лекция на тему «СИСТЕМЫ ПРОГРАММИРОВАНИЯ: ОСНОВНЫЕ ПОНЯТИЯ» / [Электронный ресурс] — Режим доступа: https://mf.grsu.by/UchProc/livak/b_lecture/lec32_SysPr (дата обращения: 19.11.2021).

2. Волкова И.А., Головин И.Г., Карпов Л.Е. Системы программирования: Учебное пособие. - М.: Издательский отдел факультета ВМК МГУ, 2009 (дата обращения: 19.11.2021).

3. Понятие языка программирования / [Электронный ресурс] — Режим доступа: https://ru.wikipedia.org/wiki/Язык_программирования (дата обращения: 20.11.2021).

4. Т. Пратт, М. Зелковиц. Языки программирования. Разработка и реализация. Санкт-Петербург, 2002 (дата обращения: 20.11.2021).

5. Определение интегрированной среды разработки программного обеспечения IDE / [Электронный ресурс] — Режим доступа: https://ru.wikipedia.org/wiki/Интегрированная_среда_разработки (дата обращения: 25.11.2021).

6. Интегрированная среда разработки Microsoft Visual Studio / [Электронный ресурс] — Режим доступа: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio (дата обращения: 25.11.2021).

7. Документация по среде разработки Microsoft Visual Studio / [Электронный ресурс] — Режим доступа: <https://docs.microsoft.com/ru-ru/visualstudio/windows/?view=vs-2022> (дата обращения: 25.11.2021).

8. Документация по среде разработки IntelliJ IDEA / [Электронный ресурс] — Режим доступа: <https://www.jetbrains.com/help/idea/documentation-tool-window> (дата обращения: 29.11.2021).

9. С. Давыдов, А. Ефимов. IntelliJ IDEA. Профессиональное программирование на Java. — СПб.: БХВ, 2005 (дата обращения: 29.11.2021).

10. Документация по среде разработки Eclipse IDE / [Электронный ресурс] — Режим доступа: <https://help.eclipse.org/latest/index> (дата обращения: 01.12.2021).

11. Руководство по использованию NetBeans IDE / [Электронный ресурс] — Режим доступа: https://netbeans.apache.org/kb/docs/java/quickstart_ru (дата обращения: 01.12.2021).