Введение в разработку программного обеспечения (ИС и ЦД)

1. Дисциплина: «Введение в РПО» для специальности ИС и ЦД
   Семестр I.
   Всего 36 часов, из них лекций 18 часов, лабораторных 18 часов, ***экзамен у ИС, зачет у ЦД***

Лектор: ***Север Александра Сергеевна***, преподаватель-стажер, кафедры программной инженерии (а.408, к.1).
email: alya.sever.01@gmail.com
telegram: @sanialo_1567

Цели и задачи курса.
Целью курса является ознакомление с принципами организации и создания надежного, качественного программного обеспечения, удовлетворяющего предъявляемым к нему требованиям.
В рамках изучения курса предполагается решение следующих задач:

- рассмотрение технологических основ процесса разработки программного обеспечения;
- изучение основ унифицированного языка UML для визуального моделирования элементов предметной области в рамках проектирования программной системы и ее основных компонент;
- приобретение навыков анализа, проектирования, документирования и разработки небольших программных комплексов.

Лекции и задания для лабораторных работ доступны в электронном виде:

https://diskstation.belstu.by:5001/
логин: student
пароль fitfit

папка:
/Для_студентов_ФИТ_БГТУ/ПРЕПОДАВАТЕЛИ/Север/Введение в РПО

2. Литература:

Основная литература:

1. **Орлов, С. А.** Программная инженерия / С. А. Орлов. – Санкт-Петербург : Питер, 2016. – 640 с.
2. **Чакон С., Штрауб Б.** Git для профессионального программиста. – СПб.: Питер, 2016. — 496 с.
3. Лаврищева, Е.М. Программная инженерия. Парадигмы, технологии и CASE-средства: учебник для вузов / Е.М.Лаврищева. –2-е изд., испр. и доп. – М.: Издательство Юрайт, 2016. – 280 с. – Серия: Университеты России.
4. Липаев, В. В. Программная инженерия. Методологические основы / В. В. Липаев. – М. : ТЕИС, 2006. – 608 с.
5. Липаев, В. В. Процессы и стандарты жизненного цикла сложных программных средств: справочник / В. В. Липаев. – М. : Синтег, 2006. – 276 с.
6. Мацяшек, Л. А. Практическая программная инженерия на основе учебного примера / Л. А. Мацяшек. – М. : БИНОМ, 2009. – 956 с.
7. Вигерс, К. И. Разработка требований к программному обеспечению / К. И. Вигерс. – М. : Русская редакция, 2004. – 576 с.

Дополнительная литература:

8. Батоврин, В. К. Толковый словарь по системной и программной инженерии / В. К. Батоврин. – М. : ДМК Пресс, 2012. – 280 с.
9. Единая система программной документации. ИПК Издательство стандартов, 2001. − 164 с.
10. Городняя, Л. В. Парадигма программирования : курс лекций / Л. В. Городняя ; Новосиб. гос. ун-т. – Новосибирск : РИЦ НГУ, 2015. – 206 с.
11. Макконнел С. Профессиональная разработка программного обеспечения. – СПб., Питер, 2007 – 240 с.

Электронные ресурсы

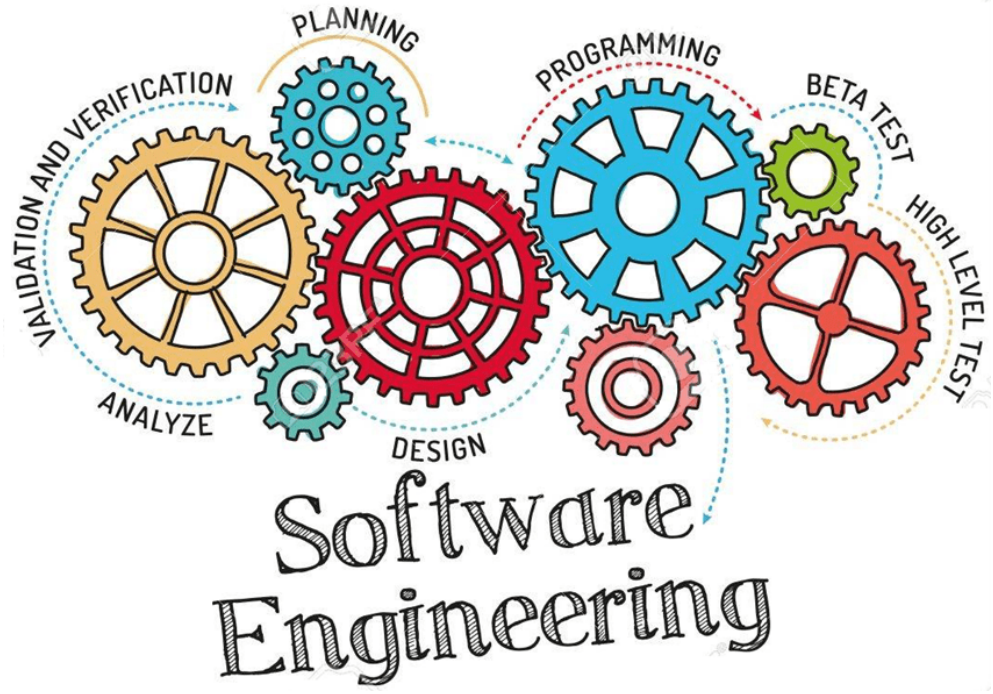
12. Git How To [Электронный ресурс]. – Режим доступа: https://githowto.com/ru/ – Дата доступа: 24.06.2021.
13. Pro Git [Электронный ресурс]. – Режим доступа: https://git-scm.com/book/ru/v2/ – Дата доступа: 24.06.2021.

Лекция 01
БГТУ, ФИТ, 1 семестр

Введение в разработку программного обеспечения (ИС и ЦД)

ВВЕДЕНИЕ. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОЙ ИНЖЕНЕРИИ (SOFTWARE ENGINEERING)

План лекции:
- становление программной инженерии;
- основные определения;
- отличия от других инженерий;
- профессиональные и этические требования
- системы программирования;
- данные, представление данных, кодировки;
- кодировка ASCII;
- стандарт кодирования Unicode;
- прямой (LE) и обратный (BE) порядок байт;
- маркер последовательности байтов (BOM).



В эпоху информационного общества (общества, основанного на знаниях):

производство **программного обеспечения** (ПО) является крупнейшей отраслью мировой экономики	***Количество программистов в 2020 году:*** **США:** оценивается в 680 тыс., а все IT-комьюнити – в 10 млн человек.

Беларусь: 71,5 тыс. IT-специалистов в 3,4 тыс. компаний (это около 2% всех работников в стране, данные Белстата)

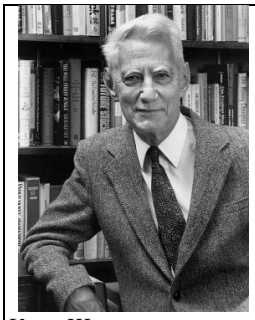
Программная инженерия (промышленное программирование) ассоциируется с разработкой сложных программ коллективами разработчиков

Проблемы становления и развития отрасли – высокая стоимость программного обеспечения, сложность его создания, необходимость управления и прогнозирования процессов разработки.

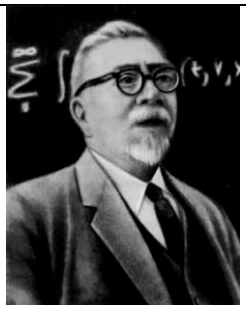
Цель программной инженерии – сокращение стоимости программ.

История

Понятие «информационное общество» зародилось в 1940-х гг. с появлением кибернетики и связано с именами ученых:



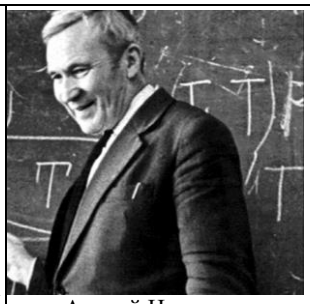
Клод Шенон - создатель теории информации, 1948 г.



Норберт Винер - основоположник кибернетики и теории искусственного интеллекта

Джон фон Нейман - создатель современной архитектуры компьютеров

Алан Тьюринг предложил в 1936 году абстрактную вычислительную «Машину Тьюринга» - модель компьютера общего назначения, которая позволила формализовать понятие алгоритма



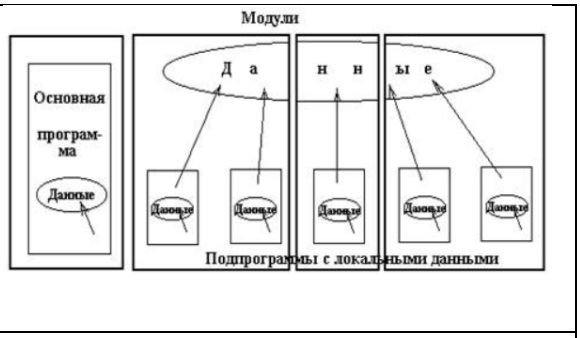
Андрей Николаевич Колмогоров - один из самых выдающихся математиков XX века. Им получены фундаментальные результаты в математической логике и др., внес важный вклад в теорию информации, теорию сложности алгоритмов.

Программирование – стадии эволюции

50-е годы 20 века.

Программирование в машинном коде для решения, главным образом, научно-технических задач (расчет по формулам).
Наличие достаточно четко сформулированного технического задания.
Отсутствие этапа проектирования.
Составление документации после завершения разработки.
Зарождение концепции модульного программирования.

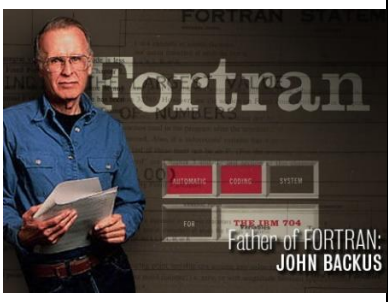
Машинный код	Ассемблер
0E	PUSH CS
1F	POP DS
BA0E00	MOV DX,000E
B409	MOV AH,09
CD21	INT 21
B8014C	MOV AX,4C01
CD21	INT 21
54	PUSH SP
68	DB 68
69	DB 69
7320	JNB 0033
7072	JO 0087
6F	DB 6F
67	DB 67
7261	JB 007A
6D	DB 6D
206361	AND [BP+DI+61],AH
6E	DB 6E
6E	DB 6E
6F	DB 6F

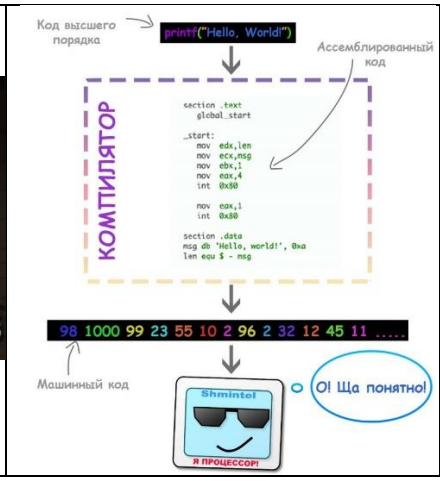


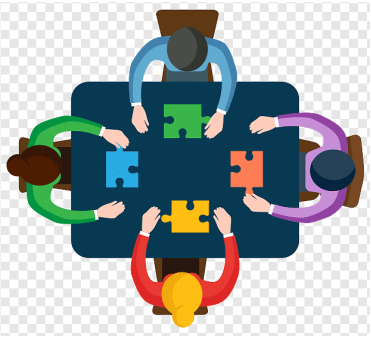
60-е годы.

Широкое использование языков программирования высокого уровня (Алгол 60, Фортран, Кобол и др.).

Возрастание сложности задач, решаемых с помощью компьютеров. Использование методов коллективной работы при создании больших программных систем.







70-е годы.

Широкое распространение информационных систем и баз данных.

Развитие абстрактных типов данных.

Исследование проблем обеспечения надежности и мобильности программных средств.

Создание методики управления коллективной разработкой программ. Появление инструментальных средств поддержки программирования.

БД

содержит обширную информацию самого различного типа: текстовую, графическую, звуковую, мультимедийную.

Примеры:

1. **в БД законов** – тексты самих законов,
2. **в БД эстрадной песни** – тексты и ноты песен, биографию авторов, информация о поэтах, композиторах и исполнителях, звуковые и видеоклипы.

Абстрактный тип данных – это совокупность данных и операций над ними

Структура данных является частью реализации АТД

Перед реализацией АТД необходимо тщательно описать все операции, которые необходимо выполнять

80-е годы.

Широкое внедрение персональных компьютеров во все сферы человеческой деятельности.

Бурное развитие пользовательских интерфейсов и создание четкой концепции качества ПО.

Внедрение объектного подхода к разработке программных систем.

Развитие концепции компьютерных сетей.

Интерфейс включает в себя:

- способы взаимодействия с внутренней частью программы (операционной системой, платформой, сервером и т.д.);
- дизайн;
- доступные функции.

90-е годы.

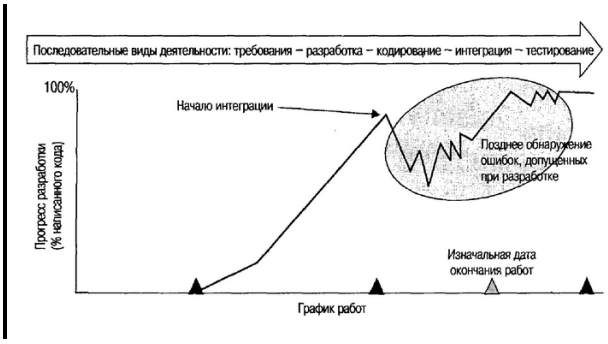
В 1989 году реализован проект Всемирной паутины.

Актуальность проблемы защиты компьютерной информации и передаваемых по сети сообщений.

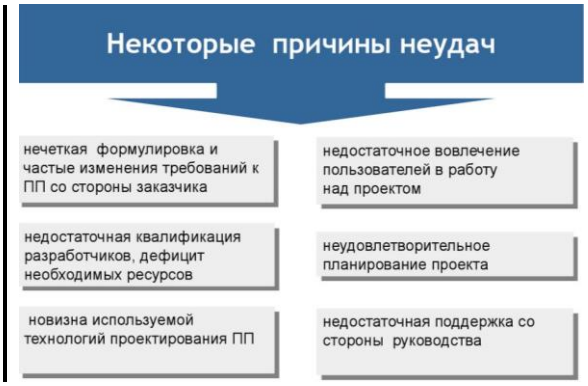
Развитие CASE-средств разработки программного обеспечения.

Разработка ПО по-прежнему остается непредсказуемой

Процент успешных проектов по созданию программного обеспечения достаточно низок



Некоторые причины неудач



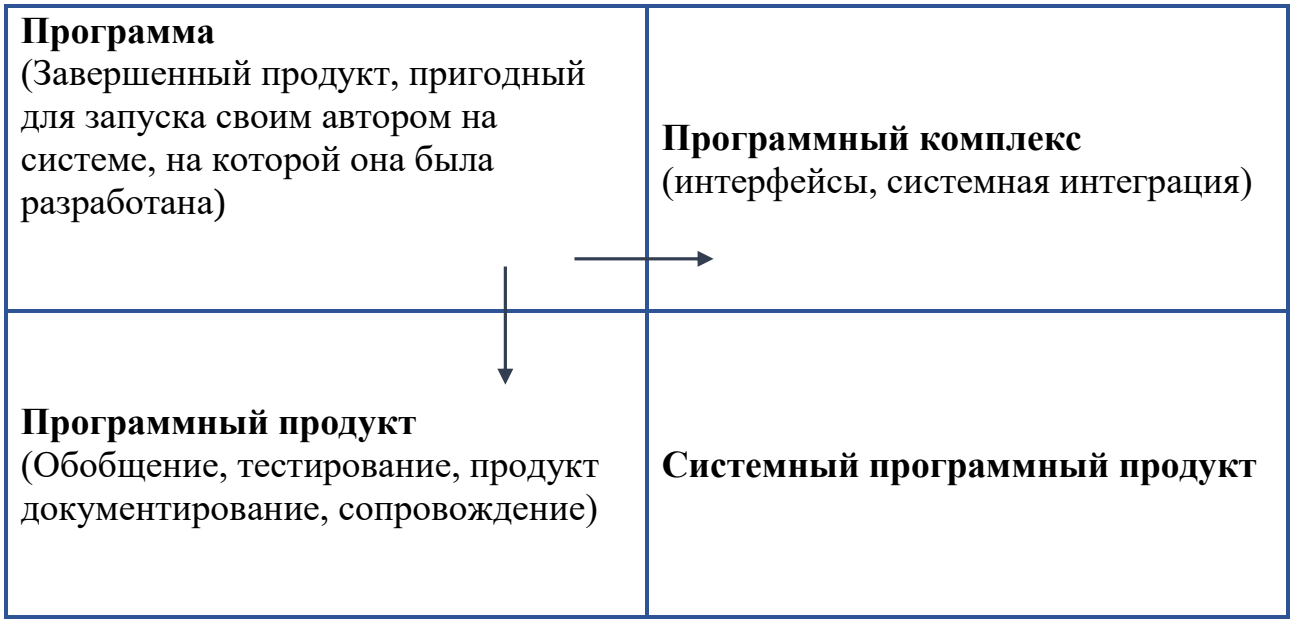
Предпосылки

Повторное использование кода (модульное программирование)

Рост сложности программ (структурное программирование)

Модификация программ (ООП)

Цикл разработки программной системы:



Основные определения:

Программа – это объект разработки, который не является осязаемым (нельзя пощупать, взвесить и т. п.), доступен пониманию ЭВМ, для которой написан.

Свойства хорошей программы

Выполнение функциональных требований
Соответствие нефункциональным требованиям
Сопровождаемость (maintainability)
Надежность (dependability)
Эффективность (efficiency)
Удобство использования (usability)

Программный продукт (ПП)**:** программа, работающая без авторского присутствия. Программный продукт исполняется, тестируется, конфигурируется без присутствия автора и сопровождается документацией.

Программное обеспечение (ПО) – совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ (ГОСТ 19781-90)

Программная инженерия (Software Engineering) ориентирована на разработку программного обеспечения прикладных и информационных систем разного назначения.

Определение из свода знаний по программной инженерии SWEBOK:

> 1) **Программная инженерия** – это применение систематического, дисциплинированного и измеряемого подхода к разработке, эксплуатации и сопровождению программного обеспечения (ПО) с применением инженерных методов к разработке ПО.
>
> 2) **Программная инженерия** – учебная дисциплина, изучающая указанные выше подходы.

Отличие от других инженерий

Программная инженерия – это система методов, средств и дисциплин планирования, разработки, эксплуатации и сопровождения программного обеспечения, готового к внедрению.

Программа – не материальный объект: фазы производства и изготовления образца отсутствуют.
Стоимость программы зависит от стоимости и качества проектирования.
Нет объективных законов контроля проекта: тестирование – единственный способ проверки

Теоретический фундамент программной инженерии

Главные положения фундаментальных наук: теория алгоритмов, математическая логика теория управления теории множеств, и т.п.

Формальные методы программирования: спецификация программ, их доказательство, верификация и тестирование, математические модели надежности, риска и т.п.;

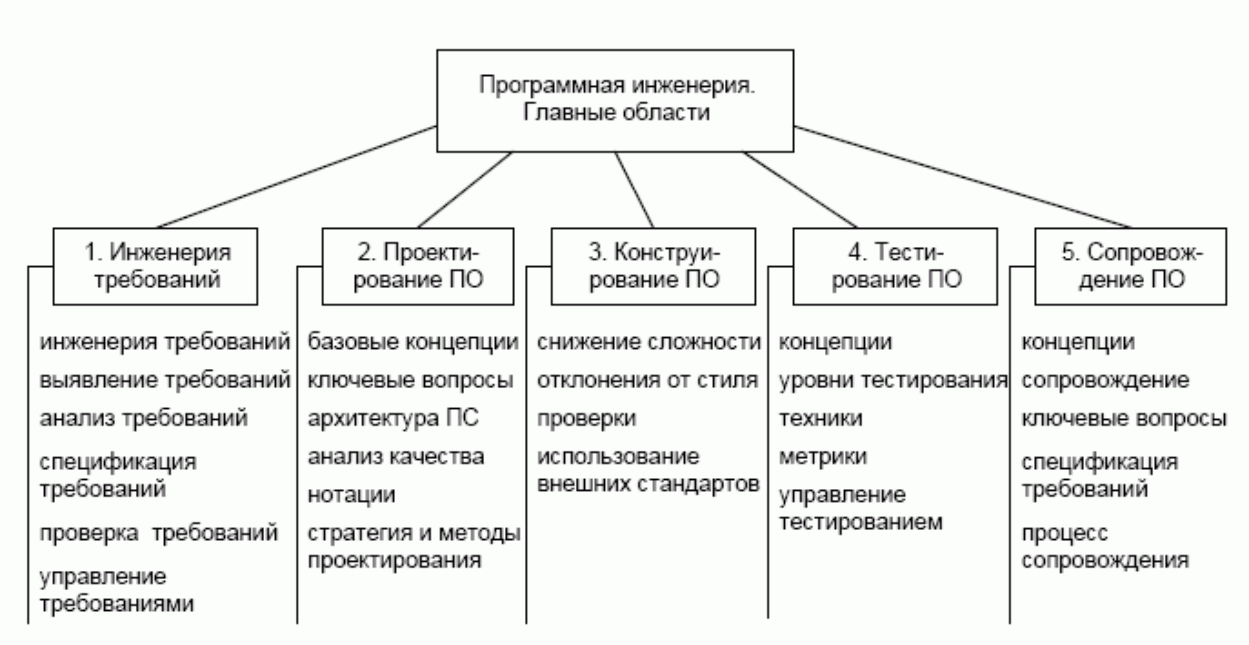
Прикладные методы: приемы, принципы, правила, отдельные действия и цельные процессы жизненного цикла (ЖЦ) производства компьютерных систем, которые являются инструментами коллективной разработки, применяемыми исполнителями крупных программных проектов;

Методы управления коллективами: планирование по сетевым графикам, контроль работ в процессах ЖЦ, измерение и оценка качества промежуточных результатов производства, прогнозирования и регулирования сроков и стоимости изготовления продукта, а также его сертификации.

Успех реализации проекта ПО обусловлен пятью взаимосвязанными аспектами:

Жизненный цикл ПО – непрерывный процесс с момента принятия решения о создании ПО до снятия его с эксплуатации.

Программная инженерия (Software Engineering)



Стандарты программной инженерии

Стандарт (standard) – норма, образец, мерило

– нормативно-технический документ, устанавливающий нормы и правила по отношению к объекту стандартизации, утверждается компетентным органом;

– типовой образец, эталон, модель, принимаемые за исходные для сопоставления с ними других объектов.

Основные типы стандартов

Корпоративные стандарты разрабатываются крупными фирмами с целью повышения качества своей продукции. Создаются на основе собственного опыта компании, но с учетом требований мировых стандартов. Не сертифицируются, но являются обязательными для применения внутри корпорации.

Отраслевые стандарты действуют в пределах организаций некоторой отрасли (министерства). Разрабатываются с учетом требований мирового опыта и специфики отрасли. Являются обязательными для отрасли. Подлежат сертификации.

Государственные стандарты (ГОСТы) принимаются государственными органами и имеют силу закона. Разрабатываются с учетом мирового опыта или на основе отраслевых стандартов. Могут иметь как рекомендательный, так и обязательный характер. Для сертификации создаются государственные или лицензированные органы сертификации.

Международные стандарты разрабатываются специальными международными организациями на основе мирового опыта и лучших корпоративных стандартов. Имеют сугубо рекомендательный характер.

Разработчики стандартов в области программной инженерии

ISO – ***The International Standards Organization*** международная организация по стандартизации, работающая в сотрудничестве с **IEC** – ***The International Electrotechnical Commission*** – международной электротехнической комиссией

IEEE Computer Society – профессиональное объединение специалистов в области программной инженерии

ACM – ***Association for Computing Machinery*** – Ассоциация по вычислительной технике

SEI – ***Software Engineering Institute*** – Институт Программной Инженерии при университете КарнегиМелон

PMI – ***Project Management Institute*** – Международный Институт Проектного Менеджмента

Объекты стандартизации в программной инженерии

- процессы разработки ПО;
- продукты разработки;
- ресурсы, которые используют процессы для создания программного продукта.

Основные стандарты программной инженерии

> ***ISO/IEC 12207 – Information Technology – Software Life Cycle Processes –*** процессы жизненного цикла программных средств**.**
>
> ***SEI CMM – Capability Maturity Model (for Software)*** – модель зрелости процессов разработки программного обеспечения.
>
> ***ISO/IEC 15504 – Software Process Assessment –*** оценка и аттестация зрелости процессов создания и сопровождения ПО. Является развитием и уточнением ISO 12207 и SEI CMM.
>
> ***PMBOK – Project Management Body of Knowledge –*** свод знаний по управлению проектами.
>
> ***SWEBOK – Software Engineering Body of Knowledge –*** свод знаний по программной инженерии.
>
> ***ACM/IEEE CC2001 – Computing Curricula 2001 –*** кадемический образовательный стандарт в области компьютерных наук.

Ядро профессиональных знаний SWEBOK (Software Engineering Body of Knowledge)

> ***Software Requirements*** – требования к ПО
>
> ***Software Design*** – проектирование ПО
>
> ***Software Construction*** – конструирование ПО
>
> ***Software Testing*** – тестирование ПО
>
> ***Software Maintenance*** – сопровождение ПО
>
> ***Software Configuration Management*** – управление конфигурацией
>
> ***Software Engineering Management*** – управление IT проектом
>
> ***Software Engineering Process*** – процесс программной инженерии
>
> ***Software Engineering Tools and Methods*** – методы и инструменты
>
> ***Software Quality*** – качество ПО

Свод знаний по управлению проектами PMI PMBOK (Project Management Body of Knowledge)

> ***Управление интеграцией – Project Integration Management***
>
> ***Управление содержанием – Project Scope Management***
>
> ***Управление временем – Project Time Management***
>
> ***Управление затратами – Project Cost Management***
>
> ***Управление рисками – Project Risk Management***
>
> ***Управление персоналом – Project Personnel Management***
>
> ***Управление коммуникациями – Project Communication Management***

Управление закупками – Project Procurement Management
Управление качеством – Project Quality Management

Кодекс этики программной инженерии (краткая версия)

- ***программные инженеры*** будут действовать соответственно общественным интересам;
- ***программные инженеры*** будут действовать в интересах клиентов и работодателя, соответственно общественным интересам;
- ***программные инженеры*** будут добиваться, чтобы произведенные ими продукты и их модификации соответствовали высочайшим профессиональным стандартам;
- ***программные инженеры*** будут добиваться честности и независимости в своих профессиональных суждениях;
- ***менеджеры и лидеры программных инженеров*** будут руководствоваться этическим подходом к руководству разработкой и сопровождением ПО, а также будут продвигать и развивать этот подход;
- ***программные инженеры*** будут улучшать целостность и репутацию своей профессии соответственно с интересами общества;
- ***программные инженеры*** будут честными по отношению к своим коллегам и будут всячески их поддерживать;
- ***программные инженеры*** в течение всей своей жизни будут учиться практике своей профессии и будут продвигать этический подход к практике своей профессии.

Принципы, положенные в основу кодекса этики программной инженерии

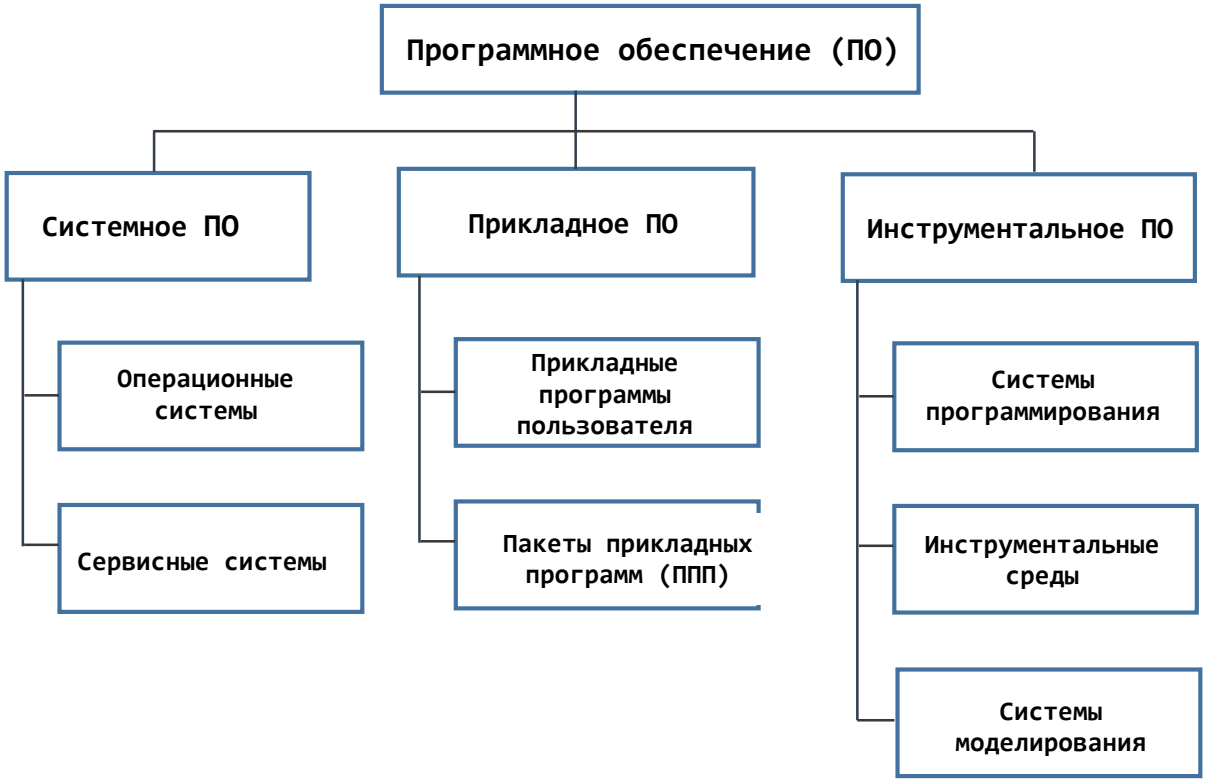
- согласование профессиональной деятельности инженеров-программистов с интересами общества;
- взаимоотношения между клиентом, работодателем и исполнителем разработки;
- достижение соответствия качества продукта лучшим профессиональным стандартам;
- честность и независимость профессиональных оценок;
- соблюдение этических норм в менеджменте и в
- сопровождении разработок;
- поддержка становления профессии в соответствии с кодексом этики;
- соблюдение этических норм во взаимоотношениях с коллегами;
- усовершенствование специальности

Определение (ГОСТ Р 51904-2002)

Программное обеспечение (ПО) – совокупность компьютерных программ и программных документов, необходимых для эксплуатации этих программ.

Определение (ISO/IEC 26514:2008)

> **Программное обеспечение (ПО)** – программа или множество программ, используемых для управления компьютером.



Классификация программного обеспечения:

Системное ПО – комплекс программ, которые обеспечивают управление компонентами компьютерной системы:

- управление ресурсами компьютера;
- создание копий используемой информации;
- проверка работоспособности устройств компьютера;
- и др.

Прикладные ПО – предназначено для выполнения определённых пользовательских задач и рассчитанная на непосредственное взаимодействие с пользователем.

Инструментальное ПО – для автоматизации процесса разработки.

Операционная система — комплекс системных программ, расширяющий возможности вычислительной системы, обеспечивающий управление её ресурсами, загрузку и выполнение прикладных программ, взаимодействие с пользователями.

Системы программирования – системные программы, предназначенные для разработки программного обеспечения.

Система программирования:

комплекс программных средств, предназначенных для автоматизации процесса разработки, отладки программного обеспечения и подготовки программного кода к выполнению

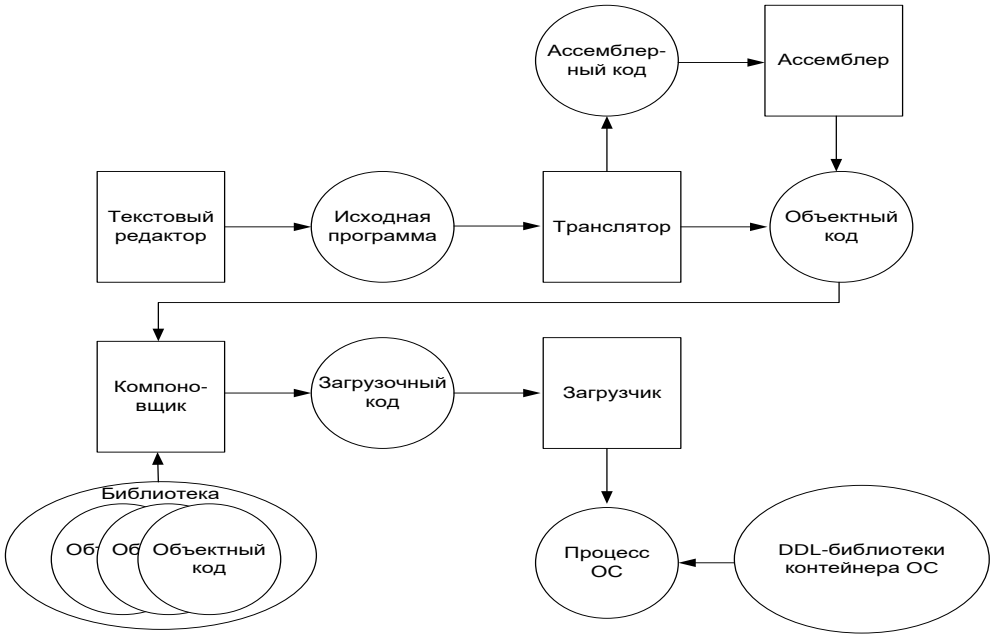
Новые требования (тенденции) в современной технологии разработке программного обеспечения:

- распространение промышленных методов организации (планирование трудозатрат, учет, контроль результатов, и т.п.) при проведении работ по работ по разработке программного обеспечения:
- перенос акцента с процесса программирования на более ранние стадии – анализ предметной области, формирование требований.

Состав системы программирования	трансляторы компоновщики отладчики профилировщики программные библиотеки редакторы кода системы поддержки версий и пр.

Система программирования является основным инструментом программиста.

Структура классической системы программирования



От исходного кода к исполняемому модулю, основные этапы преобразования:

Классическая схема создания исполняемого файла выполняется для компилируемых языков:

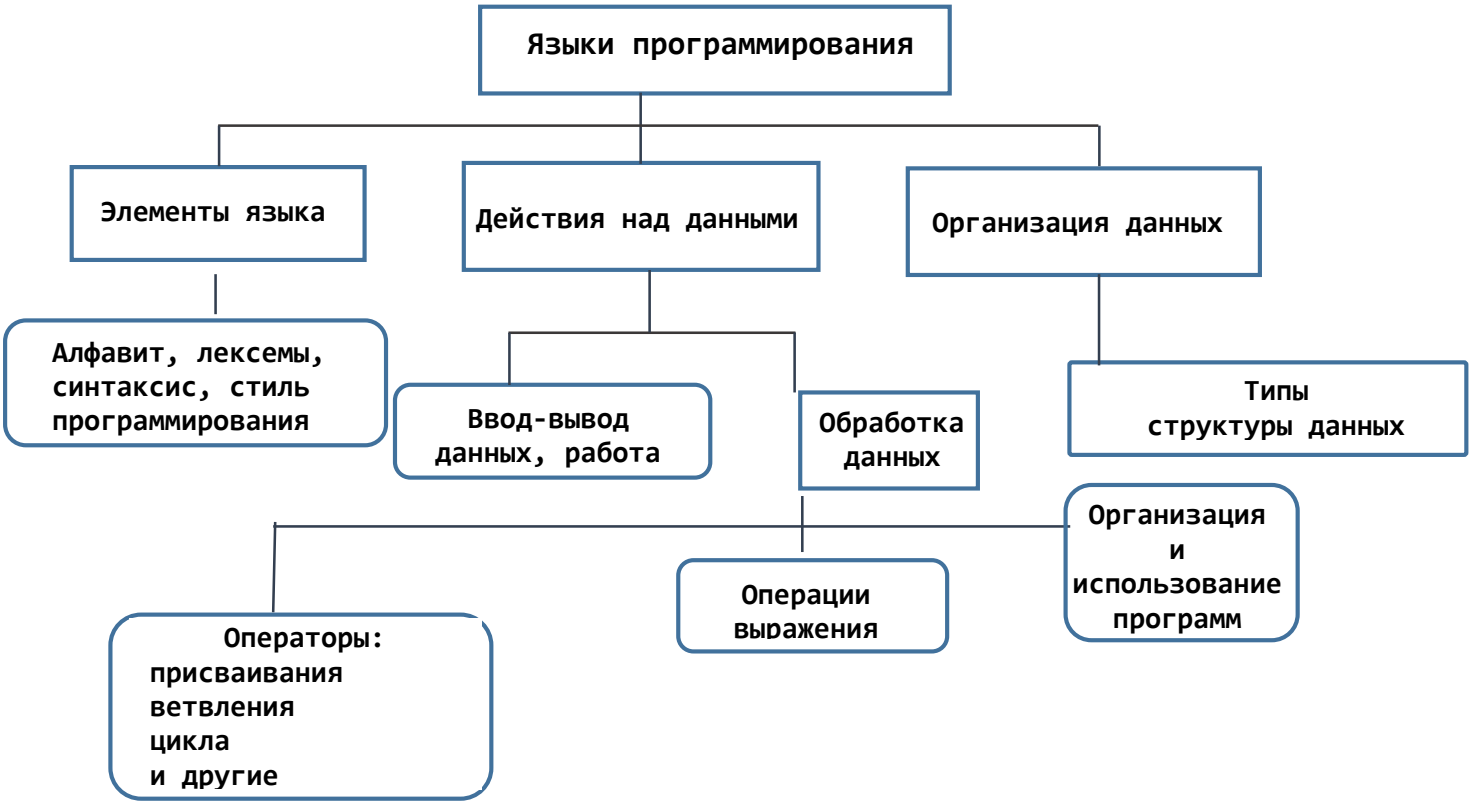
(1) обработка исходного кода препроцессором,
(2) компиляция в объектный код и
(3) компоновка объектных модулей, включая модули из объектных библиотек, в исполняемый файл.

Язык программирования	формальная знаковая система, предназначенная для записи компьютерных программ. Знаковая система определяет набор лексических, синтаксических и семантических правил написания программы (программного кода). Язык программирования представляется в виде набора спецификаций, определяющих его синтаксис и семантику.

Язык программирования представляется в виде набора спецификаций, определяющих его синтаксис и семантику.

Язык программирования определяется не только через спецификации стандарта языка, формально определяющие его синтаксис и семантику, но и через реализации стандарта (программные средства, обеспечивающих трансляцию или интерпретацию программ на этом языке), которые различаются по производителю, версии, времени выпуска, полноте воплощения стандарта, дополнительным возможностям; могут иметь определённые ошибки или особенности реализации.

Спецификация системы программирования: набор требований к системе программирования, достаточный для ее разработки.



Текстовая информация выражается с помощью естественных или формальных языков в письменной или печатной форме.

Пример:

преподаватель читает лекцию → процесс кодирования → *студент* делает для себя пометки → процесс декодирования → *студент* использует конспект

Системы счисления		
десятичная	двоичная	шестнадц.
0	0000	0
1	0001	1
2	0010	2
...	...	...
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Степень двойки	
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256

Данные в памяти компьютера

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	1	0	0	0

старший бит — младший бит

байт – минимальная адресуемая единица

бит – минимальная единица хранения (0 или 1)

Решение проблем с кодировкой текста

Текстовый символ кодируется его порядковым номером (0-127), представленным в двоичной системе счисления. 1963 ASCII.

Ранние языки, возникшие в эпоху 6-битных символов, использовали более ограниченный набор.

> ***Пример***:
> алфавит Фортрана включает 49 символов: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 = + - * / () . , $ ' : пробел

a. Американский стандартный код для обмена информацией. ASCII

ASCII (American Standard Code for Information Interchange) — американский стандартный код для обмена информацией.
ASCII — 8-битная кодировка для представления десятичных цифр, латинского и национального алфавитов, знаков препинания и управляющих символов.

Таблица кодов ASCII делится на две части:
Международным стандартом является первая половина таблицы, т.е. символы с номерами от 0 (00000000), до 127 (01111111).
К концу 1980-х годов стандартом стали 8-битные кодировки.

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

***Расширенные таблицы*:** в расширенных таблицах символы с порядковыми номерами 128-255 представляют символы национальных языков.

Переносимый набор символов	является базовым алфавитом для практически всех современных языков программирования.

Переносимый набор символов (portable character set) – набор из 103 символов, которые (стандарт POSIX) должны присутствовать в любой используемой кодировке.
Переносимый набор символов включает в себя все печатные символы US-ASCII и часть управляющих и является базовым алфавитом для практически всех современных языков программирования.

- Альтернативная кодировка **CP866** (операционная система MS-DOS):

Все специфические европейские символы во второй половине таблицы CP866 заменены на кириллицу, а псевдографические символы оставлены без изменения.

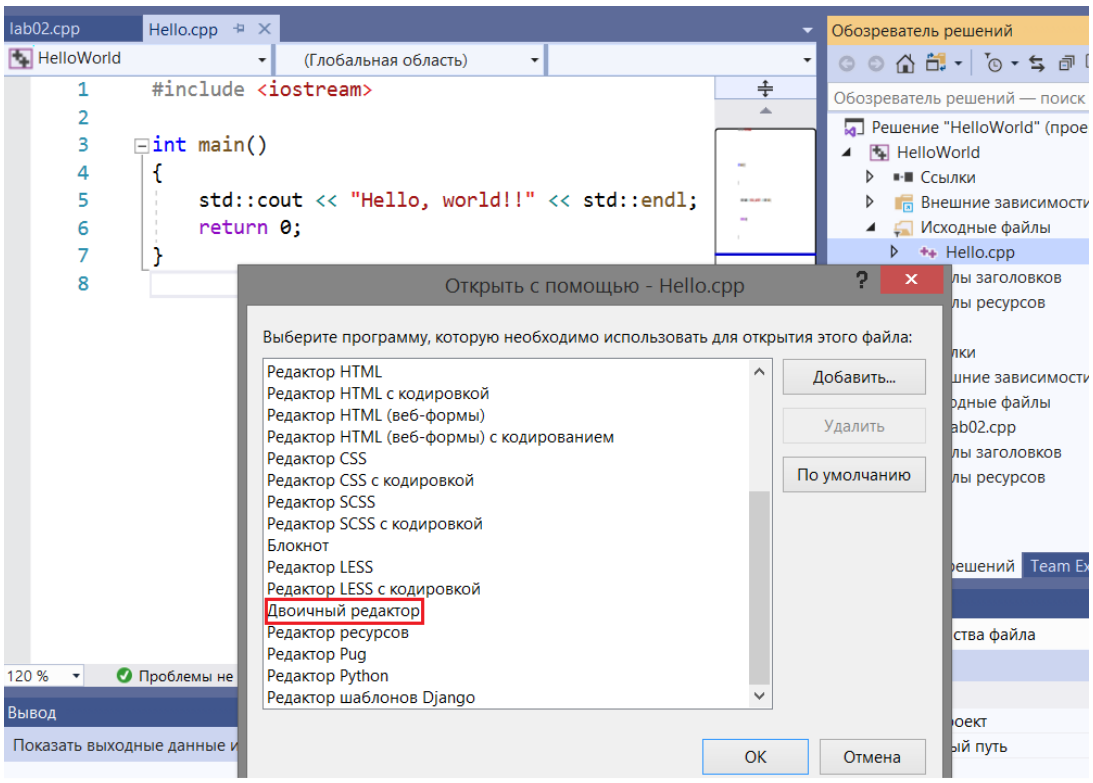
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
9	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
B	▒	▒	▒	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	▀	▄	▌	▐	▀
E	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F	Ё	ё	Є	є	Ї	ї	Ў	ў	°	∙	·	√	№	¤	■	

- русская Windows-кодировка (**Windows-1251**, синоним **CP1251**)

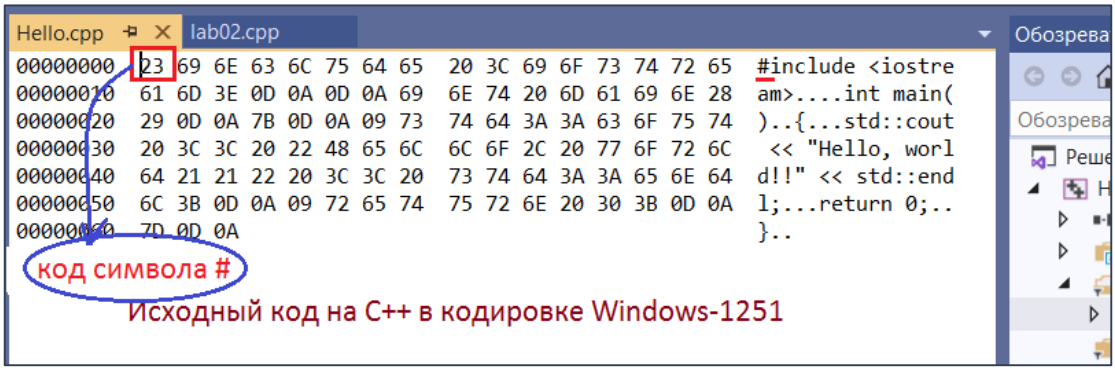
Windows-1251 — набор символов и кодировка, являющаяся стандартной 8-битной кодировкой для русских версий Microsoft Windows до 10-й версии.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	Ђ 0402	Ѓ 0403	‚ 201A	ѓ 0453	„ 201E	… 2026	† 2020	‡ 2021	€ 20AC	‰ 2030	Љ 0409	‹ 2039	Њ 040A	Ќ 040C	Ћ 040B	Џ 040F
90	ђ 0452	‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014		™ 2122	љ 0459	› 203A	њ 045A	ќ 045C	ћ 045B	џ 045F
A0	NBSP 00A0	Ў 040E	ў 045E	Ј 0408	¤ 00A4	Ґ 0490	¦ 00A6	§ 00A7	Ё 0401	© 00A9	Є 0404	« 00AB	¬ 00AC	- 00AD	® 00AE	Ї 0407
B0	° 00B0	± 00B1	І 0406	і 0456	ґ 0491	µ 00B5	¶ 00B6	· 00B7	ё 0451	№ 2116	є 0454	» 00BB	ј 0458	Ѕ 0405	ѕ 0455	ї 0457
C0	А 0410	Б 0411	В 0412	Г 0413	Д 0414	Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E	П 041F
D0	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C	Э 042D	Ю 042E	Я 042F
E0	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C	н 043D	о 043E	п 043F
F0	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C	э 044D	ю 044E	я 044F

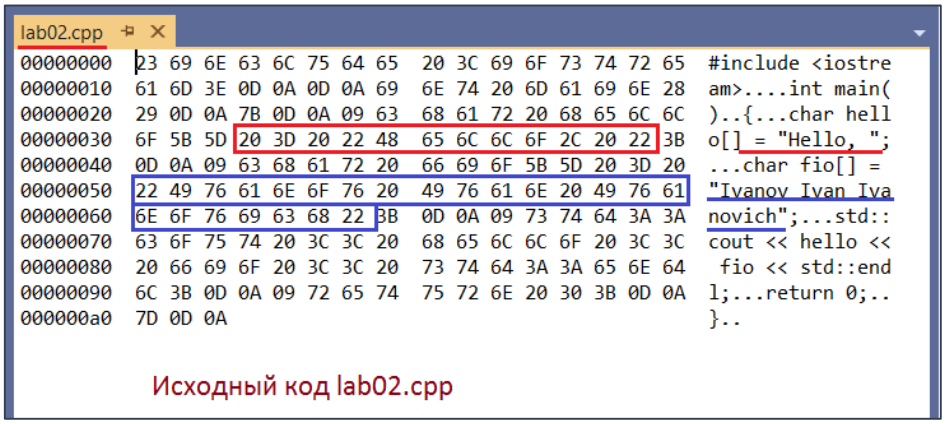
Интегрированная среда разработки Visual Studio. Открытие файла в двоичном редакторе:



Представление в памяти файла с исходным кодом:



Представление символьной информации в кодировке Windows-1251:



b. Международный стандарт UNICODE

Решение проблем	неправильного декодирования; ограниченность набора символов; преобразования из одной кодировки в другую; проблема дублирования шрифтов.

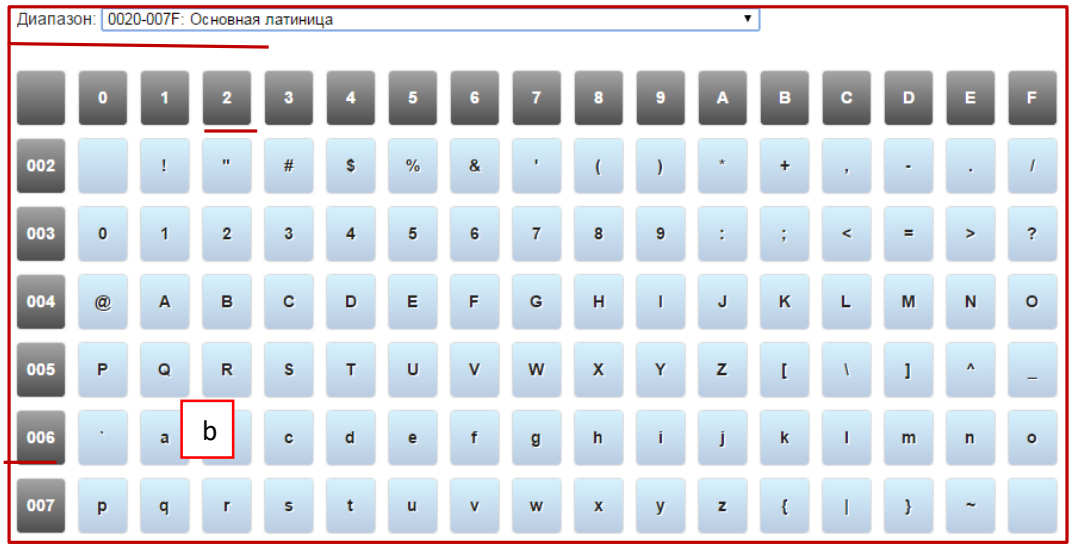
Стандарт предложен в 1991 году некоммерческой организацией Unicode Consortium, стандарт ISO/IEC 10646:2020.

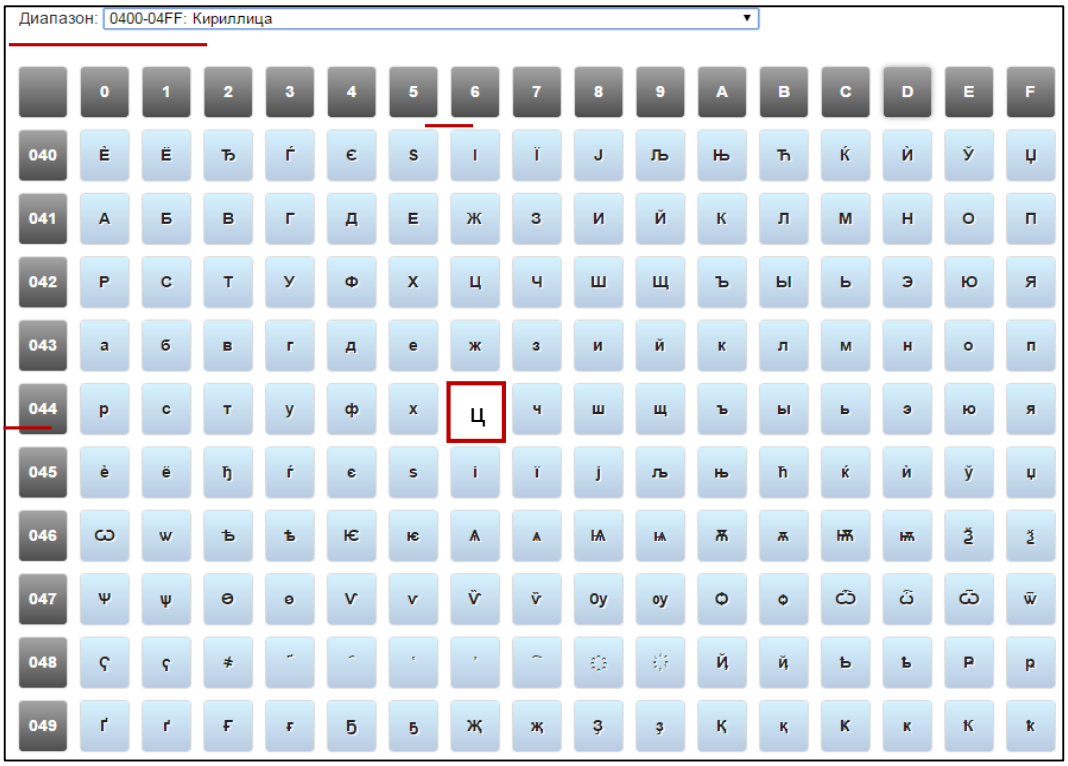
	Юникод – стандарт кодирования символов, позволяющий представить знаки почти всех письменных языков, состоит из 2х разделов: – **UCS** – universal character set (универсальный набор символов); – **UTF** – Unicode transformation format (семейство кодировок).

Принято обозначение символа **U+xxx**, где **xxx**- число в шестнадцатеричном формате.

- **UNICODE:**
  - UCS расположены в 17 плоскостях (0-16);
  - в каждой плоскости 2^{16} (65 536) символов;
  - плоскость 0 – основная (основные символы);
  - 1-14 – дополнительные;
  - 15-16 – для частного использования.

- **UNICODE:** http://foxtools.ru/Unicode





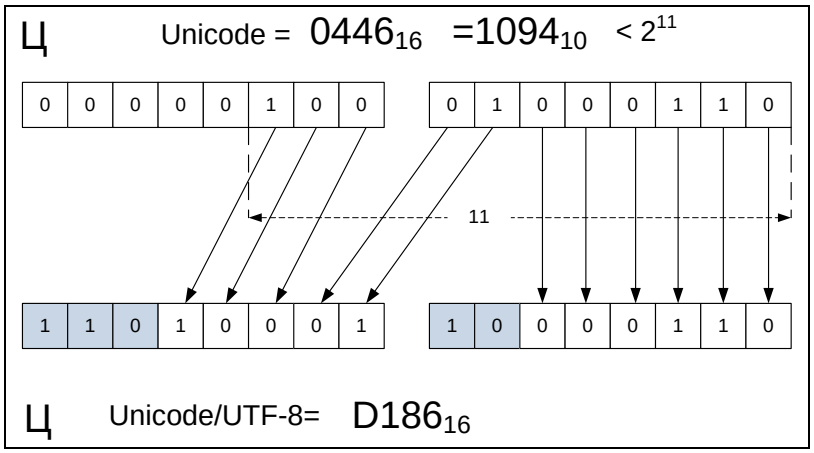
- **UNICODE:** ***кодировка UTF-8***

UTF-8 — представление Юникода, обеспечивающее совместимость со старыми системами, использовавшими 8-битные символы.

Алгоритм кодирования в UTF-8:

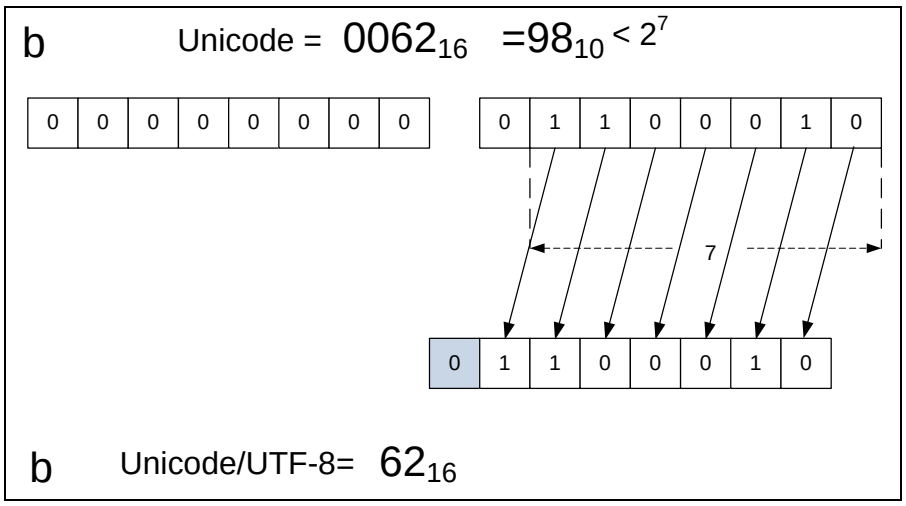
1) определить количество октетов (октет: 8 битов или 1 байт) – т.е. в какой диапазон значений попадает количество значащих символов (7, 11, 16, 21, 26, 31);
2) подготовить старшие биты первого октета:
   a. 0xxxxxxx для ***одного*** октета;
   b. 110xxxxx – для ***двух***;
   c. 1110xxxx – для ***трех*** и т.д..
   d. 10xxxxxx – для ***остальных*** октетов;
3) заполнить оставшиеся биты (выше обозначены как x) в октетах кодом символа Юникода в двоичном виде. Начать с младших битов, поставив их в младшие биты последнего октета кода. И так далее, пока все биты кода символа не будут перенесены в свободные биты октетов.

Пример:



$0446_{16} = 4*16^2 + 4*16 + 6 = 1094_{10}$

Пример:



UNICODE: ***кодировка UTF-8.*** Для символов в диапазоне:
0x00000000 ÷ 0x0000007F: 0xxxxxxx (один октет)
0x00000080 ÷ 0x000007FF: 110xxxxx 10xxxxxx (два октета)
0x00000800 ÷ 0x0000FFFF: 1110xxxx 10xxxxxx 10xxxxxx (три октета)
0x00010000 ÷ 0x001FFFFF: 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

UNICODE: ***кодировка UTF-16***
В UTF-16 символы кодируются двухбайтовыми словами (16 битов) с использованием всех возможных диапазонов значений (от 0 до FFFF16).

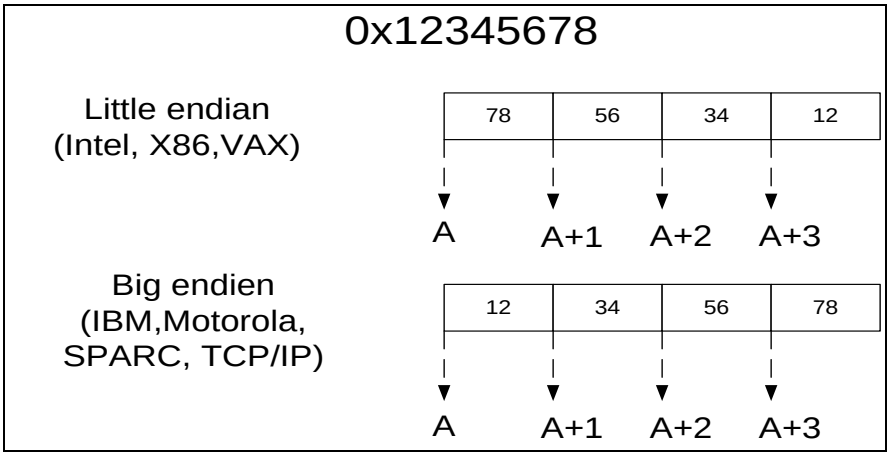
- **Маркер последовательности байтов UNICODE: BOM** (Byte Order Mark)

Для определения формата представления Юникода в начало текстового файла записывается сигнатура (обозначение) — символ U+FEFF — маркер последовательности байтов.

Шестнадцатеричное представление маркера последовательности байтов для кодировок:

Кодировка	Представление (hex)
UTF-8	EF BB BF
UTF-16 (BE)	FE FF
UTF-16 (LE)	FF FE
UTF-32 (BE)	00 00 FE FF
UTF-32 (LE)	FF FE 00 00

- **Порядок следования байтов:**
  - **LE** (Little endian order, прямой порядок, от младшего к старшему);
  - **BE** (Big endian order, обратный порядок, от старшего к младшему).



Представление в памяти целочисленного числа на платформе x86:
порядок следования байтов LE
(Little endian order, прямой порядок, от младшего к старшему)