



САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ

- **Сотавов Абакар Капланович**
- **Ассистент кафедры Информатики**(наб. канала Грибоедова, 30/32, ауд. 2038)
- e-mail: sotavov@unecon.ru
- Материалы на сайте: <http://de.unecon.ru/course/view.php?id=440>



Задача курса - обучение основам создания коммерческих программных продуктов:

- разработка эффективных **алгоритмов**
- выбор наиболее подходящих **языковых средств**
- выбор оптимальных или близких к ним **структур данных**
- профессиональное **кодирование и тестирование**
- введение в **технологии разработки ПО**



- Парадигмы программирования
- Типы данных и управляющие конструкции языка высокого уровня
- Структуры данных
 - массивы, стеки, очереди, списки, деревья
- Объектно-ориентированное программирование
 - теория ООП
 - основы программирования под Windows
 - библиотеки классов



Программист должен обладать способностью первоклассного математика к абстракции и логическому мышлению в сочетании с эдисоновским талантом сооружать всё, что угодно, из нуля и единиц.

Он должен сочетать аккуратность бухгалтера с пронциательностью разведчика, фантазию автора детективных романов с трезвой практичностью экономиста, сочетать в себе лёгкость и полет таланта Моцарта с усидчивостью и скрупулёзностью Сальери.

Академик Андрей Петрович Ершов



Виды учебной нагрузки

РУП	Код ФГОС ВО (для 14 и 15 года поступления)	Код ФГОС	Наименование	Дата и номер приказа об утверждении ФГОС ВО
ФГОС ВО	09.03.03	230700.62	Прикладная информатика	№207 от 12.03.2015

Наименование	Трудоемкость в ЗЕ	Трудоемкость в час.		Распределение часов							
				Аудиторная				Самостоятельная			
				Всего	Лекции	Практ.	Лаб	Всего	КСР	СРС	Экз.
Высокоуровневые методы информатики и программирования	5	180	5 семестр Экзаме н	72	24	48	0	108	6	66	36



	Код формируемой компетенции согласно ФГОС 3 +
ОК-7	Способностью к самоорганизации и самообразования
ПК-2	способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение;
ПК-4	способностью документировать процессы создания информационных систем на стадиях жизненного цикла;
ПК-8	способностью программировать приложения и создавать программные прототипы решения прикладных задач;
ПК-12	способностью проводить тестирование компонентов программного обеспечения ИС;



Художественная литература по программированию (must read)

- Брукс Ф. Мифический человеко-месяц, или как создаются программные комплексы.
- Макконнелл С. Совершенный код
- Спольски Дж. Джоэл о программировании (www.jelonsoftware.com).



«Программист, подобно поэту, работает почти непосредственно с чистой мыслью. Он строит свои замки в воздухе и из воздуха, творя силой воображения. <...> Однако программная конструкция, в отличие от поэтических творений, реальна, в том смысле, что она движется и работает, производя видимые результаты, которые отделимы от самой конструкции. Она печатает результаты, рисует картинки, производит звуки, приводит в движение рычаги. В наше время осуществилось волшебство мифа и легенды. С клавиатуры вводится верное заклинание, и экран монитора оживает, показывая то, чего никогда не было и не могло быть».

Ф. Брукс



«Чтобы стать специалистом, настоящим экспертом лишь в одной области программирования, нужны годы. Конечно, есть масса толковых подростков, которые за неделю осваивают Delphi, еще за неделю Python, еще за неделю Perl, после чего они считают себя знатоками. При этом у них нет даже самого отдаленного представления о том, сколь многого они не знают.»

"... изучить 90% необходимого в работе материала можно за неделю. Но на освоение оставшихся 10% может уйти пара лет. Вот тут-то и проявляется во всем блеске превосходство действительно опытных программистов над теми, кто говорит, что если он чего-то не знает, то возьмет книжку и прочтет".



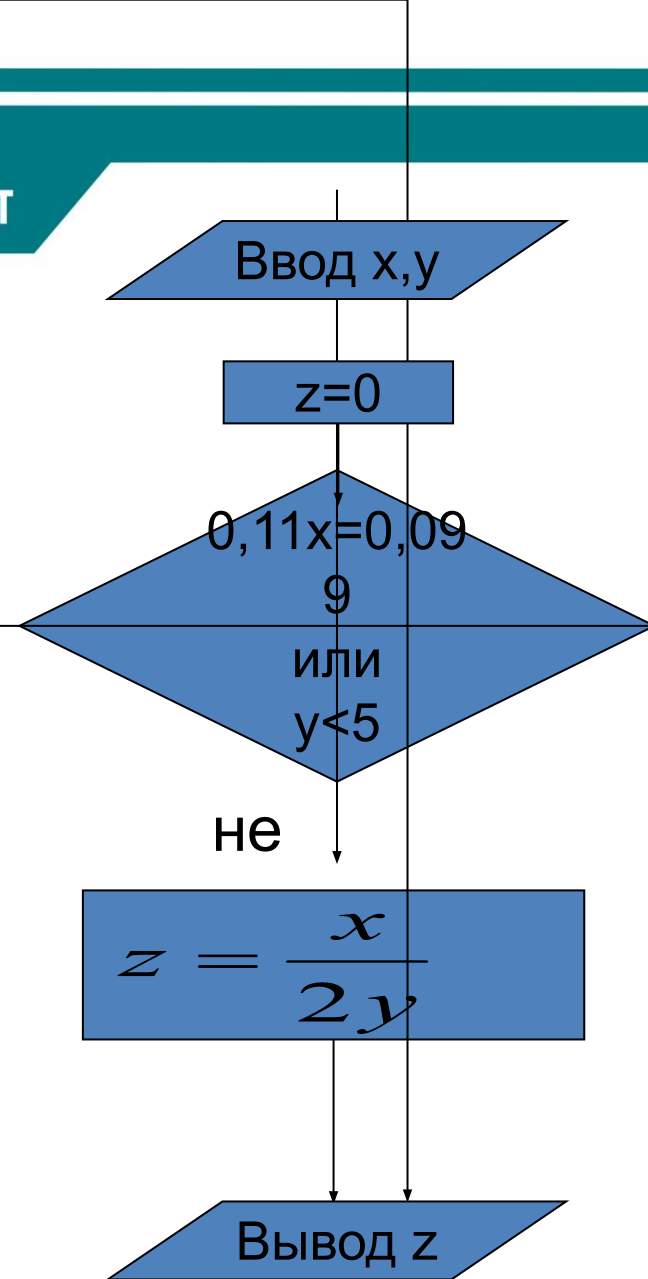
- Вирт Н. Алгоритмы + структуры данных = программы
- Кнут Д. Искусство программирования, том 1. Основные алгоритмы, 3-е изд. – М.: Издательский дом "Вильямс", 2000.
- Кнут Д. Искусство программирования, том 3. Сортировка и поиск, 2-е изд. – М.: Издательский дом "Вильямс", 2000 (есть на моем сайте в разделе «Комбинаторные алгоритмы»).
- Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.: МЦНМО, 1999 (есть на моем сайте в разделе «Комбинаторные алгоритмы»).
- Фаулер М. Рефакторинг: улучшение существующего кода. СПб: Символ-Плюс, 2007. 432 с.



1. Павловская Т.А. С#. Программирование на языке высокого уровня. Учебник — СПб.: ПИТЕР, 2010. — 432 с.
2. Павловская Т.А. «Основы языка С#. Учебно-методическое пособие» — РИО СПбГУИТМО, 2008.
3. Павловская Т.А. «Программирование на языке С#. Методические рекомендации по лабораторному практикуму» — РИО СПбГУИТМО, 2008.
4. Павловская Т.А. «Программирование на языке С#. Методические рекомендации по выполнению самостоятельной работы студентов» — РИО СПбГУИТМО, 2008.



1. Фамилия, имя, номер группы
2. Опыт программирования (языки, уровень, стаж, достижения)
3. Есть ли у Вас возможность приносить на занятия ноутбук?
4. Напишите программу, соответствующую приведенной схеме алгоритма, на любом языке программирования.
*Все переменные – вещественные, **блоки в программе не дублировать.***





Качество – степень соответствия требованиям (потребностям и ожиданиям заинтересованных лиц)

- надежность
- возможность точно планировать производство и сопровождение

- Программу должно быть легко читать и легко изменять.
- Для этого она должна иметь простую структуру (состоять из максимально независимых частей, связанных между собой логично и просто).



Внешние характеристики

- корректность
 - наличие/отсутствие дефектов в спецификации, проекте и реализации
- практичность
 - легкость изучения и использования
- эффективность
 - степень использования системных ресурсов
- надежность
 - способность системы выполнять необходимые функции; интервал между отказами
- целостность
 - способность предотвращать неавторизованный или некорректный доступ
- адаптируемость
 - возможность использования в других областях и средах
- правильность
 - степень безошибочности данных, выдаваемых системой
- живучесть
 - способность продолжать работу при недопустимых данных или в напряженных условиях



Контроль качества ПО – планомерная систематичная программа действий, призванная гарантировать, что система обладает желательными характеристиками.

Главный закон контроля качества ПО: повышение качества системы снижает расходы на ее разработку.



Парадигма — способ организации программы, то есть принцип ее построения. Наиболее распространенными являются процедурная и объектно-ориентированная парадигмы.

Они различаются способом декомпозиции, положенным в основу при создании программы.

Процедурная декомпозиция состоит в том, что задача, реализуемая программой, делится на подзадачи, а они, в свою очередь — на более мелкие этапы, то есть выполняется пошаговая детализация алгоритма решения задачи.

Объектно-ориентированная декомпозиция предполагает разбиение предметной области на объекты и реализацию этих объектов и их взаимосвязей в виде программы.

Кроме того, существуют *функциональная* и *логическая* парадигмы.



САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ
ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ

СПАСИБО ЗА ВНИМАНИЕ!



ОБЪЕДИНЯЯ ЛУЧШЕЕ

