

# Технологія Програмування та Створення Програмних Продуктів.

на основі  
*Microsoft Solutions Framework*

**Лекція 1. Вступ.**  
**Основні поняття та історія розвитку**  
**технології програмування.**



# Зміст

- 1. Мета і Характеристика курсу «ТПС ПП»**
  - Структура навчального плану**
2. Термінологія індустрії ПЗ
3. Бізнес і ІТ-проекти. Ринок ПЗ - статистика
4. Основні завдання предмету
5. Причини невдач ІТ-проектів
6. Технології програмування – шлях до успіху в розробці ПЗ
7. Література

# Формальний розподіл

- **Лекції** – 32 години (1 пара в тиждень).
- **Лабораторні** – 32 год. (1 пара в тиждень).
- **Самостійна робота** – 32 год. (по ночах, перед модулями, та перед іспитом).

# Структура курсу - лекції

## **Ч.1. Методологічні основи створення ПП**

1. Вступна лекція
2. Елементи програмної інженерії
3. Організація технологічного процесу розробки ПЗ
4. Управління програмним проектом
5. Аналіз та проектування програмних систем
6. Розроблення архітектури ПП
7. Розроблення структури ПП
8. ...

## **Ч.2. Методологія створення ПП на основі MSF**

1. Візуальне моделювання на основі UML
2. Базові принципи Microsoft Solution Framework
3. MSF Team Model
4. MSF Process Model
5. MSF Project Management Discipline
6. MSF Risk Management Discipline
7. MSF Readiness Management Discipline
8. ...

# Структура курсу - практика

Вибираємо собі індивідуальне завдання (створення ПП)

- ЛР 1. Специфікація замовника на ПП
- ЛР 2. Функціональна специфікація на ПП
- ЛР 3. Планування процесу створення ПП в MS Project
- ЛР 4. Проектування та розроблення інтерфейсу користувача
- ЛР 5. Проектування та розроблення прототипу ПП
- ЛР 6. Використання системи управління версіями при роботі над ПП (MS SourceSafe)
- ЛР 7. Використання систем управління "багами" в розробці ПП
- ЛР 8. Використання систем колективної розробки ПП

# Задачі та мета курсу

- Розкрити основні поняття та суть дисципліни "software engineering" (інженерія ПЗ)
- Отримати відповіді на усі питання, пов'язані з інженерією ПЗ
- Сформуувати етичні та професійні засади інженера з розроблення ПЗ

# Програмування...



В багатьох курсах ви вивчаєте **програмування**...Що ж таке «програмування»?

- **Програмування** (Computer science) – молода область знань, що активно розвивається.
  - А питання про те, що ж це таке, давно турбує як філософів, так і все програмістське товариство.
- **Класична версія:**
  - мистецтво;
  - наука.
- **Сучасні реалії:**  
програмування - це **бізнес (ІТ-проекти)**.

# ІТ-проекти

Під **ІТ-проектами** будемо розуміти проекти в області інформаційних технологій.

Далі будемо розглядати і вивчати такі ІТ-проекти, метою яких являється розроблення **програмного забезпечення (ПЗ)**.



# Розповсюдження ІТ систем

- ✧ На сьогодні всі економіки розвинутих країн залежать від компютерів та ПЗ !!!
- ✧ Все більше і більше технічних систем управляються за допомогою ПЗ !!
- ✧ Інженерія ПЗ базується на використанні теорії, методів і засобів (інструментів) для професійного розроблення ПЗ !
- ✧ Витрати на розроблення, супровід та модернізацію ПЗ становить значну частку ВВП в усіх розвинутих країнах світу .

# Питання



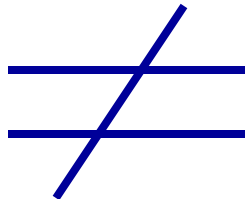
1. **Що таке програмне забезпечення (ПЗ)?**
  2. **Чим ПЗ відрізняється від звичайної програми?**
- **Вчора ми з товаришом написали «Калькулятор»:**
    1. **Це точно програма!**
    2. **А чи є вона програмним забезпеченням (ПЗ)?**
    3. **І чи є вона програмним продуктом (ПП)??**

*Ваші версії...*

# Програми і Програмне Забезпечення

**Програмне забезпечення (Software)** – набір комп'ютерних програм, процедур і зв'язаної з ними документації і даних (ISO/IEC 12207).

Програма



Програмне  
забезпечення

# Програмне Забезпечення (Software) і програмний продукт

Отже,

**ПЗ** = програма + документація + [...].

Замість ПЗ часто вживають терміни:

- «програмний засіб» = ПЗ
- «програмний продукт»  $\neq$  ПЗ

**Програмний продукт**: одна з головних властивостей  
ПП – його можна продавати !

Комерційна цінність ПП – залог успіху ІТ бізнесу з  
розроблення ПЗ.

# Атрибути ПП

- Правильне ПЗ
- Надійне ПЗ

ПП повинен характеризуватись :

- *Супроводжуваність*
  - ПП повинен допускати зміни і еволюціонувати
- *Надійність*
  - ПП повинен бути надійним і викликати довіру користувачів
- *Ефективність*
  - ПП не повинен в пусту використовувати чи блокувати системні ресурси
- *Сприйнятність*
  - ПП повинен сприйматись користувачами, для яких він був створений, тобто він повинен бути: зрозумілим, простим і узгодженим з іншими програмними системами

# Вартість ПП

- ✧ Вартість ПЗ часто переважає вартість компютерної системи. А сумарна вартість ПП на персональному компютері (лаптопі) в декілька разів перевищує вартість самого компютера.
- ✧ Підтримка, супровід і модернізація ПЗ коштує більше, ніж розроблення ПЗ. Для ПЗ з тривалим життєвим циклом, кошти на супровід в декілька разів перевищують кошти на розроблення ПЗ.
- ✧ Покупка існуючого на ринку ПЗ коштує дешевше, ніж самостійне розроблення (або замовлення) нового ПЗ
- ✧ Інженерія ПЗ направлена на підвищення ефективності і зниження вартості розробки нового ПЗ

# Класифікація ПП

## ✧ Загально-вживане ПЗ:

- Незалежні (stand-alone) системи, які рекламуються, розповсюджуються, продаються та встановлюються будь-яким користувачом, хто хоче їх отримати
  - *Приклади* – PC software such as graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists.

## ✧ Заповне ПЗ:

- Програмні системи, які розробляються для окремого замовника (чи типу замовників), яке виконує специфічні завдання чи задачі цього замовника
  - *Приклади* – embedded control systems, air traffic control software, traffic monitoring systems.

# Специфікація на ПП

## ✧ Загально-вживане ПЗ:

- Специфікація того, що ПЗ повинен робити, належить розробнику ПЗ, і рішення щодо зміни ПЗ також виробляється розробником

## ✧ Замовне ПЗ:

- The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.
- Специфікація того, що ПЗ повинен робити, належить замовнику ПЗ, і він приймає рішення про необхідні зміни в ПЗ.



# Бізнес та ІТ-проекти... (1)



- Зараз **програмування** - це не тільки наука і мистецтво, але і вигідний **бізнес**
- Для того, щоб цей бізнес був **успішним**, необхідно виконання багатьох умов...
- **Яких?**

*Ваші версії...*

# Бізнес та ІТ-проекти... (2)

**!** Необхідна умова **успіху** фірми з розроблення ПЗ:

- **Випуск програмного продукту на ринок**
  - Належної якості;
  - Вчасно;
  - Цікавого для потенційних користувачів.
- **Витрати повинні відповідати наперед затвердженому бюджету**

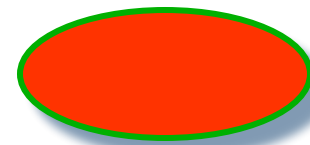
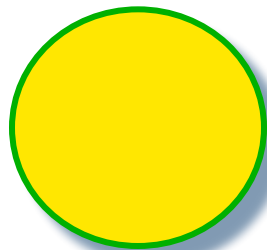
# Бізнес та ІТ-проекти... (3)

Багато ІТ проектів мають проблеми:

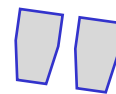
Ціль:

Результат:

Функціональність



Час



Бюджет



# Статистика по ІТ-проектах... (1)

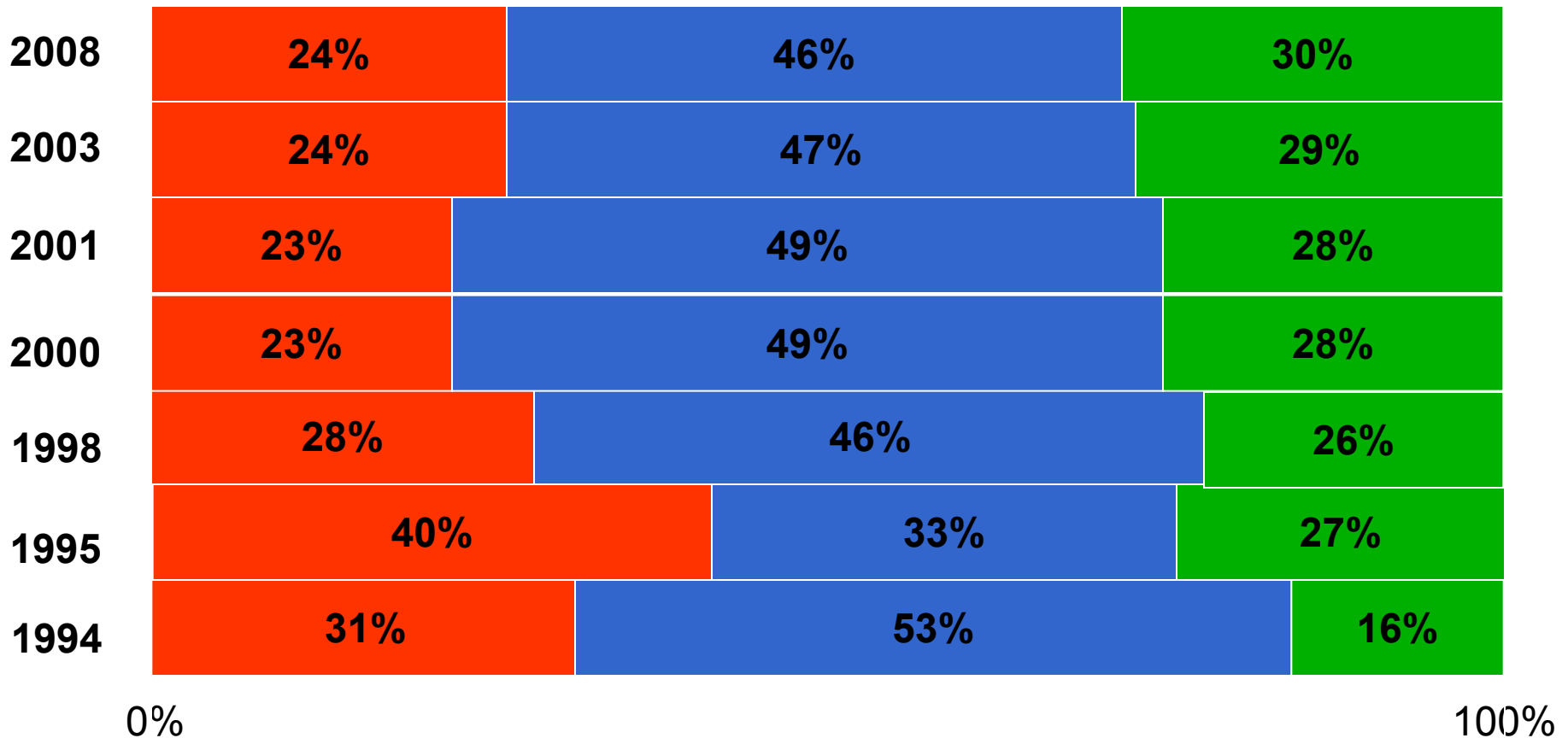
- **Провалені:** закінчились невдачою – мету проекту взагалі не було досягнуто
- **Проблемні:** закінчились створенням ПЗ, однак превищили **бюджет**, або/і не були завершені **вчасно**, або/і запланована **функціональність** не була реалізована повністю
- **Успішні:** закінчились створенням продукту, вклались в **бюджет**, завершені **вчасно** і вся запланована **функціональність** реалізована

# Статистика по ІТ-проектах... (2)

*Провалені*

*Проблемні*

*Успішні*



\* Джерело: The Standish Group International, Extreme Chaos.

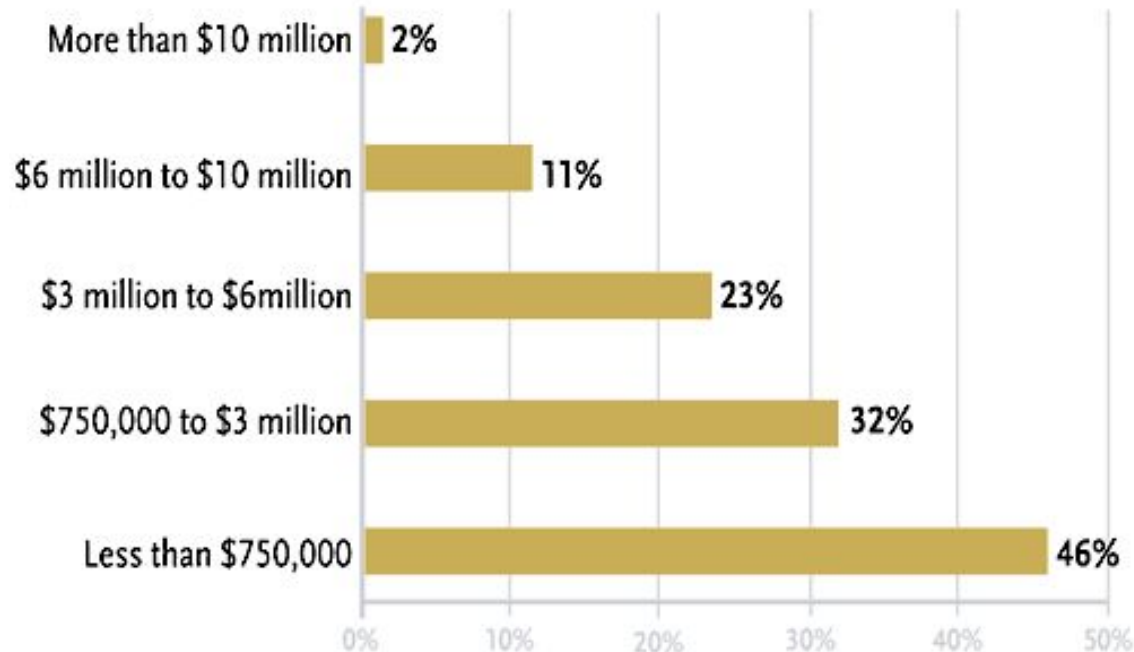
Дані взято з <http://www.softwagemag.com/archive/2001feb/CollaborativeMgt.html>,  
<http://www-128.ibm.com/developerworks/rational/library/feb06/marasco/>

# Ще трохи статистики...(1)

В **2004 р.** на розробку ПЗ витрачено **3 700 000 000 \$**

## Project Success

Smaller initiatives fare better at reaching goals than larger projects do.



SOURCE: THE STANDISH GROUP

\* Джерело: The Standish Group International, Extreme Chaos.

Дані взято з <http://www.softwagemag.com/archive/2001feb/CollaborativeMgt.html>,  
<http://www-128.ibm.com/developerworks/rational/library/feb06/marasco/>

# Характеристика галузі розробки ПЗ

- Є добре підготовані програмісти
- Є грамотні аналітики
- Проблема з хорошими менеджерами
- Проблеми з документуванням і локалізацією ПП
- Проблеми з рекламою і просуванням власних ПП і торгових марок
- В основному офшорне програмування

# Тенденції ринку ПП

- Швидкий ріст обсягів ІТ-ринку і ринку ПЗ.
- Створення професійних організацій і кластерів фірм-розробників ПЗ.
- Україна вийшла на 4-те місце в світі по обсягах офшорного програмування
- Мізерна доля власне українського ПЗ на світовому ринку

## **Що потрібно для розвитку галузі ?**

- **грамотні спеціалісти**, здатні підвищити **успішність** ІТ проектів



# Основні завдання курсу:

- Вивчити причини невдач ІТ-проектів
- Виявити способи усунення цих причин
- Навчитись застосовувати ці способи на практиці
- Ознайомитись з технологіями створення ПП провідних світових ІТ-компаній

# Причини ускладнень в ІТ-проектах

## **Спеціалісти виділяють наступні причини:**

1. Нереліастичні часові рамки
  - *пояснення...*
2. Недостатня кількість виконавців
  - *пояснення ...*
3. Розмиті границі проекту
  - *пояснення ...*
4. Недостатня фінансова забезпеченість
  - *пояснення ...*
5. Нехватка кваліфікованих кадрів
  - *пояснення ...*

# Хто винуватий?



- Всі попередні 5 пунктів відносяться до компетенції управлінців (менеджерів) різного рівня
- Проектами з розроблення ПЗ дуже важко ефективно **управляти**
- Чому?

*Ваші версії...*

# Чому з'являється складність управління?

- **Людський фактор:** Традиційні складнощі управління інтелектуальним колективом
- **Нова молода галузь:** Не вистарчає кваліфікованих кадрів, методів, технологій, досвіду
- **Складність задач:** Ріст можливостей приводить до росту складності поставлених задач і навпаки
- **Труднощі в роботі с замовниками і майбутніми користувачами:** Замовники не знають, що вони дійсно хочуть. Користувачі погано володіють ПК і не розуміють як він працює і що від нього очікувати. Всі розмовляють на різних «мовах»...
- ...

# Як боротися зі складністю?

1. Зі складністю задач?
2. Зі складністю управління?

## Відповідь:

- Слід розробляти і впроваджувати спеціальні **технології**

- (1) – *технології програмування* (курс Основи програмування, ООП і т.д.)
- (2) – *методології розроблення ПЗ: MSF, RUP, XP, ...* (даний курс)



# Стратегії розробки ПЗ

Існує 3 основних стратегії розробки ПЗ:

1. Послідовна стратегія – (водоспадна) тобто лінійна послідовність етапів програмування
2. Інкрементна стратегія – програмування виконується у вигляді послідовності версій
3. Еволюційна стратегія - програмування у вигляді послідовності версій, але вимоги уточнюються в результаті розробки

Характеристика стратегій:

- Початкова специфікація
- Цикли
- Проміжний продукт

# Використання Методологій

pmcommunity

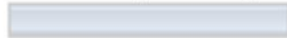
- Є результати опитування, які стосуються пострадянського простору. Не впевнені, що на них можна опиратись, але певного роду висновки все ж таки можна зробити. А саме – нам є куди покращуватись!

## Какая методология используется в вашем проекте?

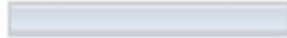
Через %опу (30.0%, 276 Votes)



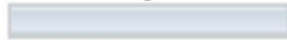
Scrum (18.0%, 167 Votes)



Agile-based (не Scrum и не XP) (18.0%, 163 Votes)



Как получится (18.0%, 162 Votes)



Водопад/Waterfal (5.0%, 48 Votes)



RUP-based (5.0%, 44 Votes)



Другое (3.0%, 28 Votes)



[www.happy-pm.com](http://www.happy-pm.com)





# Технології програмування...(1)

**Технологія** – це сукупність виробничих процесів в певній галузі виробництва, а також науковий опис способів виробництва\*.

В ІТ під “виробництвом” розуміють створення (розроблення) програмних систем.

---

**Технологія програмування** – це сукупність методів, способів і засобів (інструментів) для скорочення вартості і підвищення якості розробки програмних систем.

# Технології програмування...(2)

Над створенням програмного продукту працюють представники різних спеціальностей:

- Аналітики;
- Управлінці (менеджери);
- Тестувальники;
- Кодувальники (програмісти);
- Технічні письменники (компютерні лінгвісти);
- Дизайнери;
- Системні адміністратори;
- Спеціалісти з повторного використання;
- Спеціалісти з ергономіки;
- інші ...

# Технології програмування...(3)

- Створення ПП – **технологічний процес**, який має свої етапи (стадії, фази, кроки, тощо)
- При створенні ПП можуть застосовуватись різні **технології**
- Ці технології відносяться не тільки до кодування, але і до аналізу, проектування, управління проектом, розробки документації, тощо...

---

В проектуванні і програмуванні активно застосовують такі технології:

- Структурне програмування
- Модульне програмування
- Об'єктно-орієнтоване програмування
- Компонентне програмування
- Сервіс-орієнтоване програмування

# I. Структурне програмування...(1)

Э. Дэйкстра (60-і роки):

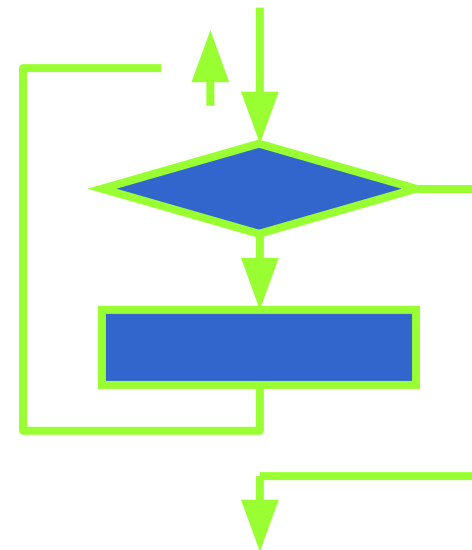
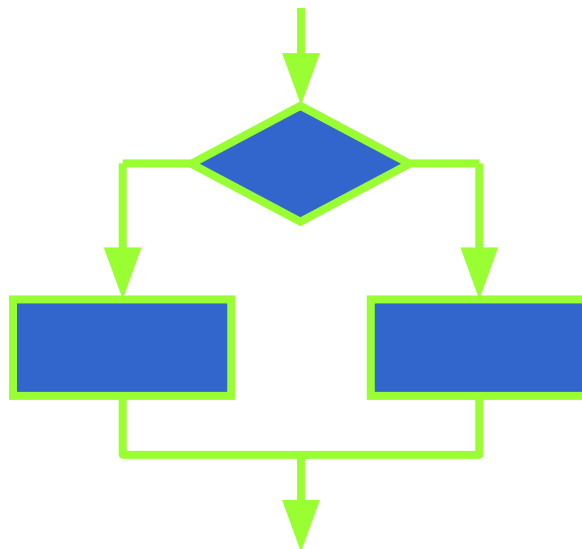
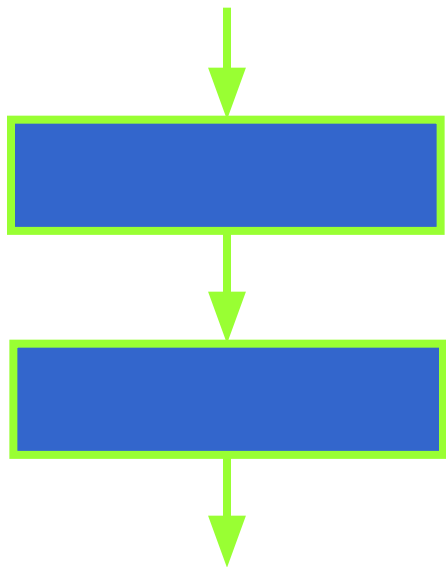
Для кожної **простої програми** можна побудувати функціонально-еквівалентну їй **структурну програму**, тобто програму, що сформована на основі фіксованої базової множини, що включає:

1. структуру послідовних дій,
2. структуру вибору одної з 2-х дій
3. структуру циклу, тобто багатократного повторення деякої дії з перевіркою умови зупинки повторення

# I. Структурне програмування...(2)

**Проста програма** -> один вхід і один вихід.

**Базові конструкції структурного програмування:**



# I. Структурне програмування (3)

- **Стандартизація і лінійність програми – спосіб зниження складності**

Для зменшення складності:

- Алгоритм повинен мати 1 вхід і 1 вихід
- Ніяких "goto"
- Відсутність залежності від мови програмування
- Стандартний набір операторів для будь-якої мови програмування

## II. Модульне програмування...(1)

- **Основна ідея:** розбиваємо складну задачу на підзадачі, кожну з яких при необхідності розбиваємо знову і т.д.
- Отримуємо прості задачі, їх розв'язуємо і об'єднуємо

## II. Модульне програмування (2)

- Структурне програмування – універсальний базис алгоритмічних конструкцій.
- **Модульне програмування – специфічний для задачі базис з модулів**
  - Вищий рівень абстракції
  - Налаштування на конкретну задачу
  - Можливості повторного використання
  - Можливості колективної розробки – розподіл завдань та робіт



# III. Об'єктно-орієнтоване програмування...(1)

- Подальша боротьба зі складністю
- Технологія стартує з етапу аналізу
- Аналіз -> Проектування -> Програмування
- В основі – об'єктна модель і об'єктна декомпозиція

# III. Об'єктно-орієнтоване програмування (2)

- Основні принципи **об'єктної моделі**:
  - абстракція;
  - інкапсуляція;
  - ієрархія (наслідування, агрегація);
  - поліморфізм;
  - модульність.
- **Об'єктна декомпозиція** (на відміну від алгоритмічної): елементи проекту – класи і об'єкти (а не алгоритми)  
І лише потім дані і алгоритми

# IV. Компонентне програмування...(1)

- **Компонентне програмування** – розвиток об'єктно-орієнтованої ідеології
- Вводиться наступний рівень абстракції – класи об'єднують в **компоненти**
- **Основний принцип компонентного програмування:** збирання програми з готових компонент, які можуть в загальному випадку бути написані на різних мовах

# IV. Компонентне програмування (2)

- **Компоненти:**

- програмний код у вигляді самостійного модуля
- може використовуватись в незмінному виді
- может піддаватись налаштуванню
- Має власну поведінку (функціональність)

- Компонент ізольований від зовнішнього світу своїм **інтерфейсом** – набором методів (їх сигнатурами)

- **Компонентна програма** – набір незалежних компонентів, зв'язаних один з одним через інтерфейси

# V. Сервіс-орієнтоване програмування (1)

- **Сервіси:**

- програмний код у вигляді відокремленого платформи-незалежного “модуля”
- використовується в незмінному виді, але піддається налаштуванню
- має власну поведінку (функціональність)

- **Найбільше розповсюдження:**

- Мережеві сервіси
- Web-сервіси (інтернет-сервіси)
- Хмарні сервіси

# Література до лекції

**1. И. Соммервиль.** Инженерия программного обеспечения, 6 изд. – И.д. "Вильямс", 2002.

- <http://www.cs.st-andrews.ac.uk/~ifs/Books/SE9/>

Ian Sommerville. Software Engineering. 9th Edition.

- <http://www.cs.st-andrews.ac.uk/~ifs/Books/SE8/>

Ian Sommerville. Software Engineering. 8th Edition.

**1. Г. Буч.** Объектно-ориентированный анализ и проектирование с примерами приложений на С++. 2-е изд. – Бином, 1998.

**2. Р. Лингер, Х. Миллс, Б. Уитт.** Теория и практика структурного программирования. – М.: Мир, 1982.

# Висновки

- **Програмування** - молода область, що активно розвивається, за півстоліттям свого розвитку що пододала величезний шлях. Будучи як мистецтвом, так і наукою, у наш час термін програмування придбав якісно нове забарвлення, ставши однією з галузей бізнесу.
- **Під ІТ-проектами** можна розуміти будь-які проекти в області інформаційних технологій. Ми далі розглядатимемо лише ті ІТ-проекти, метою яких є розробка ПП.
- **Програмне забезпечення (Software)** - набір комп'ютерних програм, процедур і пов'язаної з ними документації і даних. Програмне забезпечення - це не просто програма, це ще і документація, і керівництво користувача. Замість терміну програмне забезпечення часто використовують термін програмний продукт.
- Для того, щоб **ІТ-бізнес**, пов'язаний з розробкою ПЗ, був успішним, необхідно випускати якісне ПЗ, цікаве потенційним користувачам, робити це у задані терміни та вкладатися до наявного бюджету. На жаль, частка провалених проектів як і раніше катастрофічно висока.
- **Ринок ПЗ** в світі показує великі темпи зростання. У цю галузь вкладаються величезні гроші. Зараз у галузі ІТ на підйомі - втішний факт для зміцнення національних ІТ-компаній.
- **Основними причинами невдач** ІТ-проектів є:
  1. Нереалістичні часові рамки.
  2. Недостатня кількості виконавців.
  3. Розмиті межі проекту.
  4. Недостатнє фінансування (брак коштів).
  5. Недостатня кваліфікація наявних кадрів.
- **Технології програмування** . Саме використання сучасних технологій – це шлях до успіху в розробці ПЗ. Використання різних технологій дозволяє долати складність вирішуваних завдань і, відповідно, складність створення якісного ПЗ. Серед основних можна виділити наступні технології програмування: структурне, модульне, об'єктно-орієнтоване, компонентне та сервіс-орієнтоване програмування.

# Література

- 1. И. Соммервиль.** Инженерия программного обеспечения, 6 изд. – И.д. "Вильямс", 2002.
- 2. Г. Буч.** Объектно-ориентированный анализ и проектирование с примерами приложений на C++. Второе издание. – Бином, 1998.
- 3. N. Wirth.** Program Development by Stepwise Refinement // Communications of the ACM vol.26(1).– 1971, 1983.
- 4. O. Dahl, E. Dijkstra, C.A.R. Hoare.** Structured Programming.–London, England: Academic Press, 1972.
- 5. Р. Лингер, Х. Миллс, Б. Уитт.** Теория и практика структурного программирования. – М.: Мир, 1982.
- 6. Э. Салливан.** Время – деньги. – М.:Microsoft Press, Русская редакция, 2002.