

Увод в програмирането

Лекция 1: **Въведение**

доц. д-р Димитър Димитров

Добре дошли в СУ и ФМИ!

- Реализация във водещи IT фирми
- Комбинация от
 - теоретични познания – напр. математика, теоретични компютърни науки, AI и др.
 - практични познания – голям избор от курсове като Java, web технологии, Android, iOS, .NET, Python, Network Security и много други
- Възможност за преподавателска и научна кариера
- Какво се изисква от мен, за да постигна това?

- Лични акаунти, служебна поща
- Платформа за е-обучение Moodle – <https://learn.fmi.uni-sofia.bg>
 - Страница на курса по УП
 - Съобщения за контролни и други новини
 - Присъствено обучение в ел. среда при нужда
 - Контакти с екипа
 - Домашни, контролни, изпити
 - И др.

Курс по УП - организационни въпроси

Формат

- Лекции
 - Презентациите се качват в Moodle – няма нужда да бъдат преписвани или снимани
 - Секциите, означени с “Допълнителен материал”, не се включват в изпита
- Семинарни упражнения в компютърни зали
- Практикум в компютърни зали
 - ИС: отделен избираем курс, но е препоръчително да бъде записан
 - Информатика, Математика, ПМ, Статистика: част от основния курс

Оценяване на ИС

- Основна дисциплина (задължителна за бак., избираема за маг.):
 - Две контролни през семестъра (с тежести 40:60 помежду си)
 - Поне първото е на хартия
 - Домашни
 - Писмен изпит – задачи (30%)
 - Освобождаване при:
 - Среден успех от контролните поне 4.50
 - Предадени всички домашни
 - Теоретичен изпит (30%)
 - Тест с отворени и затворени въпроси
- Практикум (избираем само за бак., отделна оценка в книжката):
 - Две контролни на компютри
 - Домашни
- При кандидатстване за работа 6-ца без покритие не помага

Оценяване на **Инф, М, ПМ, Стат**

- Три контролни през семестъра
- Домашни
- Писмен изпит – задачи (30%)
 - Освобождаване при:
 - среден успех от контролните поне 4.50
 - предадени всички домашни
- Теоретичен изпит (30%)
 - Тест с отворени и затворени въпроси
- При кандидатстване за работа висока оценка без покритие не помага

Литература

- Учебник на Магдалина Тодорова
- Сборник със задачи на М. Тодорова и колектив
- Интернет – неограничен брой ресурси на английски език

Какво е програмиране?

Компютър

- “Като телевизор с много копчета отпред” - НЕ
- Смартфони, фотоапарати, IoT устройства...
- Въпреки сложното устройство на съвременните компютри, те могат да бъдат разгледани като дори глупави машини, които изпълняват поредица от (прости) операции
- Компютърът не мисли, той изпълнява
- Допълнителен материал:
 - Машина на Тюринг
 - Архитектура на von Neumann от 1946 г.

Програма

- Редица от инструкции, водеща до решаване на определена задача
- Пример:
 1. Премести стойността на клетка 15000 от паметта в регистър AX.
 2. Извади 10 от регистъра AX.
 3. Ако резултатът е положително число, изпълни командата, намираща се в клетка 25000 от паметта.
- Хардуер vs софтуер

Програмиране

- Работата на програмиста е именно да създаде такава последователност от инструкции

Алгоритъм

- Няма общоприета формална дефиниция на алгоритъм
- Механизъм за намиране на решение, който е еднозначен, изпълним и завършващ
- Алгоритъмът е последователност от инструкции (указания, команди, стъпки, оператори) за действие, която при изпълнението си реализира поставена цел
- Примери

Език за програмиране

- Тъй като процесорът поддържа ограничен набор от прости инструкции, директното писане на такава редица е много трудно
 - Горната програма при процесор 80386 на Intel:
161 15000 45 10 127 25000
- За целта са създадени най-различни езици за програмиране:
 - По-разбираеми за човека
 - Достатъчно формални, за да може по недвусмислен начин да се преведе програмата до изпълнима поредица от инструкции

Видове езици за програмиране според нивото на абстракция (1)

- Езици от ниско ниво
 - По-близо до машината, по-далече от човека
 - Дават пълен контрол върху хардуера
 - Позволяват писането на много бързи програми
 - Пример: Асемблер
 - Вместо с числа инструкциите се записват с малко по-разбираеми изрази, напр.
MOV AX, [15000]
SUB AX, 10
JG 25000

Видове езици за програмиране според нивото на абстракция (2)

- Езици от високо ниво
 - По-близо до човека, по-далече от машината
 - Скриват детайли от компютърните системи, напр. управлението на паметта
 - Много по-лесни за писане на програми
 - Примери – четими дори от неспециалист:
 - Java/C++: `totalPrice = product.getPrice() * quantity;`
 - C# (и по-конкретно LINQ):

```
var results = from student in Students
               where student.firstName == "Dimitar"
               select new {student.lastName, student.age};
```


C++

- В курса по УП ще използваме C++
 - Първата версия на езика е от 1983 г.
- Като цяло език от високо ниво, но предоставя и възможности като език от пониско ниво => език от средно ниво
 - Различни мнения по въпроса

Цел на курса по УП

- Увод в програмирането, а не само увод в програмирането на C++
- C++ е просто един от възможните езици, на които можеше да се провежда курсът
- Покриват се основни знания, с помощта на които лесно може да се научи следващ език
 - Инф, ИС: стига се до рекурсия и struct
 - ПМ, М, Стат: изучава се и въведение в ООП
- Ще разработваме конзолни приложения
 - Без user-friendly потребителски интерфейс
 - За по-различни – говорете с асистентите; запишете избираеми курсове

Програмиране с език ОТ ВИСОКО НИВО

- Програмистът пише текст (source code) на даден програмен език
- Друга програма – компилатор или интерпретатор, превежда текста до поредица от инструкции, която компютърът разбира
 - Разлика

Среди за разработка

- IDE (Integrated Development Environment)
- Значително улеснява процеса по създаване на софтуер
- Обикновено съдържа:
 - Редактор за програмен код
 - Съвременните среди поддържат автоматично допълване на текста по време на писане
 - Може да има компилатор / интерпретатор
 - Инструменти за автоматизиране на процеса на създаване на изпълнимите файлове
 - Debugger

Какъв софтуер ще ползваме

- Microsoft Visual Studio
 - Безплатен лицензиран софтуер на Microsoft за образователни цели – <http://elms.fmi.uni-sofia.bg>
- Или Visual Studio Code (различно е)
 - Отделно си инсталирайте MinGW
- Или...

C++ - да започваме

Особености на синтаксиса на C++

- Синтаксисът на много съвременни езици за програмиране е базиран на синтаксиса на C
 - Java, C#, JavaScript, PHP и др.
 - Този факт улеснява изучаването на нов език
- Case-sensitive, т.е. има значение дали нещо ще бъде изписано с главни или малки букви
- А какво е синтаксис? Ще видим на следващия слайд...

Описание на език за програмиране

- Синтаксис – определя се от множество от правила, които дефинират как програмите се изграждат като редици от символи
 - Ще го описваме формално
- Семантика – определя се от множество от правила, дефиниращи как се изпълняват програмите
 - Ще го описваме неформално

Метаезик на Бекус-Наур

- Удобен начин строго да опишем точно как се пише дадено нещо на даден формален език
- $\langle \rangle$ - ограждат синтактичните конструкции
 - Пример: $\langle \text{програма} \rangle$, $\langle \text{число} \rangle$, $\langle \text{цяло_число} \rangle$
- $::=$ - това е
- $|$ - или
- Примери:
- $\langle \text{число} \rangle ::= \langle \text{цяло_число} \rangle | \langle \text{реално_число} \rangle$
- $\langle \text{цифра} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

- {...} - ограденото се повтаря 0, 1 или повече пъти
 - Пример:
$$\langle \text{редица_от_променливи} \rangle ::= \langle \text{променлива} \rangle \{, \langle \text{променлива} \rangle \}$$

означава
 - $\langle \text{променлива} \rangle$ или
 - $\langle \text{променлива} \rangle, \langle \text{променлива} \rangle$ или
 - $\langle \text{променлива} \rangle, \langle \text{променлива} \rangle, \langle \text{променлива} \rangle$ и т.н.
- [...] - ограденото в скобите може да участва в описанието, но може и да бъде пропуснато.
- $\langle \text{цяло_число} \rangle ::= [+|-]\langle \text{цяло_число_без_знак} \rangle$

Демонстрация

- Hello World!
 - Или примерът от Демонстрация 1, ако е имало упражнения
- Ще бъдат разгледани:
 - Идентификатори: $\langle \text{идентификатор} \rangle ::= _ | \langle \text{английска_буква} \rangle \{ \langle \text{английска_буква} \rangle | \langle \text{цифра} \rangle | _ \}$
 - Запазени думи: int, return, ...
 - Литерали (явни константи):
 - Числови (1, -5, +2.34, 1e-02, 012, 0x123)
 - Низови ("Hello World!")
 - Какво е 0x123? Ще разберем по-късно
 - Операции (+, -, *, /)
 - Коментари (// коментар до края на реда, /* коментар */)

Бит, байт, мегабайт...

- Бит – две състояния – 0, 1 b
- Байт – (обикновено) 8 бита В
 - 256 възможни състояния (2^8)
- Килобайт – 1024 (2^{10}) байта KB
- Мегабайт – 1024 килобайта MB
- Гигабайт, терабайт, ... GB, TB...
- Срещат се и представките “киби” (Ki), “меби” (Mi) и т.н., понеже по стандарт “кило” означава 1000, а не 1024

(Позиционни) бройни системи

- Десетична – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Преобразуване от и към десетична система
- Двоична – само 0 и 1
 - На хардуерно ниво всичко се представя така
- Шестнадесетична – 0, 1, ..., 9, A, B, C, D, E, F
 - Един байт се представя с 2 цифри
 - Лесно преобразуване от и към двоична система
 - Всяка 16-ична цифра се преобразува в групичка от 4 двоични цифри (и обратно)
 - Виждали сме я например при указване на цветове в HTML/CSS – напр. #0080FF

Променливи

- Място в паметта
- Име (идентификатор)
- Тип
- Стойност
- Примери:

```
int a; // декларация на променливата a
int b = 5, c; // същото като int b; b = 5; int c;
a = b + 1; // на a присвояваме нова стойност
```

Вход (четене) от клавиатурата

- `cin >> <променлива> { >> <променлива> };`
- `cin >> a; cin >> b; cin >> c;`
е същото като
`cin >> a >> b >> c;`
- Допълнителен материал: по-общо казано – четене от стандартния вход; може например да четем текста, отпечатан от друга програма

Изход (отпечатване) на екрана

- `cout << <израз> { << <израз> };`
- `cout << a << b << c;`

Тип

- Всяка променлива, всеки израз в C++ има тип
- Типът определя
 - Множество от допустимите стойности
 - Операции
- Когато декларираме променлива, трябва да определим нейния тип
 - В рамките на курса ще разгледаме най-различни типове

Целочислени типове

- В примерите използвахме `int`
- Множество от стойности: $[-2^{31}; 2^{31}-1]$ (4 байта)
- Модификатори:
 - `short` $[-2^{15}; 2^{15}-1]$
 - `long` $[-2^{63}; 2^{63}-1]$
 - `unsigned` $[0; 2^x-1]$ ($x = 16, 32, 64$)
- Операции:
 - `+a`, `-a` (унарни операции)
 - `+`, `-`, `*` (умножение), `/` (частно), `%` (остатък)
 - Пример: `cout << 13 % 10`; ще отпечата 3

Числа с плаваща запетая (floating point)

- Ако искаме да можем да работим с дробни числа, `int` не ни върши работа
- `double` (8 байта), `float` (4 байта)
- `<число_с_плаваща_запетая> ::=`
`<цяло_число>.[<цяло_число_без_знак>]`
`[(E|e)<цяло_число>]`
- Примерни литерали: 2.34, 12e-2, 10.14E+03, .23
- Операции: всички за целочислен тип без %
- Целите числа са подмножество на реалните, не може ли да използваме винаги `double`?
 - Не, не е редно

Константи

- **const** <тип> <идентификатор> **=** <израз>;
- стойността
 - трябва да бъде зададена при дефиниране
 - не може да се променя след това

Математически функции

- `#include <cmath>`
- `abs(x)` – абсолютна стойност, `x` е целочислен израз, `fabs(x)` – абс. стойност, `x` е floating point
- `sin(x)`, `cos(x)`, `tan(x)`, `asin(x)`, `acos(x)`, `atan(x)`
- `exp(x)`, `log(x)` – натурален, `log10(x)`
- `ceil(x)` – закръгляне нагоре, `floor(x)` – надолу
- `sqrt(x)` – корен квадратен, `pow(x, n)` – x^n

Демонстрация №1

- Програма, която намира стойността на следния израз за дадена стойност на $x \in \mathbb{R}$:

а)
$$\frac{\sqrt{2x^3 - 1} + \sin 2x}{|\sin x + \cos^2 x| + \frac{1}{2}}$$

б)
$$\sin \sqrt{\frac{\frac{x}{2x-1}}{\cos^2 \frac{x}{|x^5-1|}}}$$

- Студент₁: Аз ги знам тези работи, защо да губя време да идвам в часовете по УП?!
- Преподавател:
 - Използвай часовете да научиш неща, които не знаеш (има много такива :)
 - Искай от преподавателите да ти обясняват по-интересни неща, особено на практикума
 - Всяка година се случва да напишем ниски оценки на студенти, които имат опит в програмирането – значи е трябвало да прегледат материала и да отсеят новите неща от скучните

- Студент₂: Тези неща са ми супер неразбираеми
- Преподавател:
 - Упражнявай се САМ – имаш компютър (или комп. клуб във ФМИ), пиши програми, компютърът ще ти показва кое е грешно
 - Питай за всичко преподавателите по УП – на почти еднаква възраст сме с теб, ще те разберем
 - Програмирай, недей само да четеш сборника
 - Измисляй си интересни програмки и питай Google или нас, когато стигнеш до нещо непознато

- Студент₁: Ама аз наистина ги разбирам тия неща, няма да идвам поне на първите лекции
- Преподавател: ако смяташ, че следният код няма да отпечата нищо, трябва да дойдеш :)

```
int a = 0, b = 1;  
if (a = b) cout << "Strange";
```

Демонстрация №2

- Искаме да боядисаме стена, но не знаем колко кутии боя да купим
- Знаем:
 - размерите на стената в метри
 - колко грама се изразходват на кв. метър
 - колко кг е една кутия
- Трябва да предвидим 10% резерв

За домашно

- Инсталирайте си IDE
- Осигурете си, че ще следите редовно служебната си поща

- Въпроси?



Допълнителни теми, свързани с университета

- СУСИ
- Стипендии за висок успех
 - И поправителна сесия през август за нисък успех
- Програма Erasmus за обучение в чужбина
- Държавен изпит за получаване на диплома