

ВЪВЕДЕНИЕ В ОБЕКТНО-ОРИЕНТИРАНОТО ПРОГРАМИРАНЕ

1. Еволюция на езика C
2. Основни принципи на обектния подход

1. Еволюция на езика C

- През 60-те години се въвеждат принципите на **структурното програмиране**;
- Езикът C е създаден от Dennis Ritchie през 70-те години за нуждите на операционна система UNIX.

1. Еволюция на езика C

- **Обектно-ориентираното програмиране** отбелязва нов етап в развитието на софтуера;
- Обектно-ориентираният подход безспорно повишава ефективността на програмиста.

1. Еволюция на езика C

- Корените на C++ трябва да се търсят в Норвежките **въоръжени сили**.
- **От Норвегия произхожда езикът Simula**, един от първите, които използват **класове** (класът е програмна единица, съдържаща както данни, така и свързани с тях функции).
- **Simula** е бил разработен за **моделиране на събития**.

1. Еволюция на езика C

- Bjarne Stroustrup през 1978 г. написва симулатор като част от дисертацията си за докторска степен към компютърната лаборатория на Кеймбриджкия университет.
- Stroustrup забелязва, че използването на **класове** от **Simula** е чудесен начин за отразяване на **взаимодействието** на различни машини в мрежа

1. Еволюция на езика C

- Езикът C++ е създаден от Bjarne Stroustrup в AT&T Bell Laboratories през 1980г.
- През 1983 г. той получава официалното си наименование – C++.
- Първоначалната работна версия на езика се е наричала “C класове”.
- Stroustrup създава C++ като **разновидност на C** и добавя **по-смислените типове данни** на Simula.
- Така C++ приема **уникалната гъвкавост и ефективност на C**.
- Езикът продължава да се развива от AT&T. **Основните му редакции са направени през 1985 г. и 1989 г.**

2. Основни принципи на обектния подход

- **Абстрактен тип данни.** Програмистът може да определя собствени типове данни. Всеки нов тип данни се представя чрез модел, наречен *клас*. Класът се състои от елементи, наречени *членове*. Членовете могат да бъдат *данни* или *функции*.

2. Основни принципи на обектния подход

- **Наследственост.** Механизъм на наследяване позволява един клас да наследява данни от друг клас и в същото време да добави нови членове към този модел. Полученият клас се нарича *производен*, защото е *произхождащ* от друг (други) клас (класове).
- ✓ Производният клас също може да бъде наследяван.
- ✓ Механизмът на наследяване въвежда принцип на подчиненост, което означава, че се създава йерархична схема между класовете.
- ✓ Йерархичната наследствена схема или *просто наследяване* (производен клас наследява само един клас) има вид на дърво.
- ✓ Когато клас наследява два или повече класа тогава е налице *множествено наследяване*.

2. Основни принципи на обектния подход

- **Полиморфизъм.** Тази характеристика се изразява във възможността за динамичен многовариантен избор на член-функция (метод) на клас в йерархична наследствена структура от класове.
- ✓ От програмна гледна точка това се изразява във възможността име на член-функция (метод) да се свърже с няколко възможни реализации.
- ✓ Обикновено на етапа на компилиране на програмата се определя коя функция ще бъде изпълнена.
- ✓ В езиките за ООП този избор за специална категория функции, наречени виртуални, не е известен предварително и се определя при изпълнение на програмата.