

**Sveučilište u Zagrebu**  
**PMF – Matematički odjel**



# **Objektno programiranje (C++)**

Predavanje 01 – Uvod

**Vinko Petričević**

# Ciljevi kolegija Objektno programiranje (C++)

- Uvod u objektno orijentirano i generičko programiranje kroz programski jezik C++;
- Detaljna prezentacija jezika C++-a;
- Napredno korištenje standardne biblioteke STL;
- Tehnike programiranje, softverski predlošci;

## Pretpostavke

- Poznavanje osnova programskog jezika C++.
- Poznavanje osnova STL-a: ulaz/izlaz, stringovi, vektori, itd.

# Organizacija kolegija

- Predavanja: srijeda 15-17
- Vježbe: srijeda 17-19, četvrtak 11-13
- Konzultacije:
  - Vinko Petričević: srijeda 14-15 i 19-20, ili po dogovoru
  - Matej Mihelčić: četvrtak 14-16, ili po dogovoru
- Ispit: dva kolokvija koja se pišu na papiru.
- Popravni kolokvij: Za studente koji ne budu zadovoljni ili ne zadovolje, bit će organiziran popravni kolokvij koji će pokrivati cijelo gradivo
- Aktivnost na nastavi

# Zaključivanje ocjene

<b>bodovi</b>	<b>ocjena</b>
0 do 44	nedovoljan
45 do 59	dovoljan
60 do 74	dobar
75 do 89	vrlo dobar
90 do 100	izvrstan

# Literatura

- Stanley B. Lippman, Josée Lajoie, Barbara E. Moo: *C++ Primer, Fifth Edition*, Addison Wesley Professional, 2012.
- Materijali na web-u: Web je vrlo bogat materijalima koji se odnose na C++. Na primjer [en.cppreference.com/w/](http://en.cppreference.com/w/) je temeljni resurs za učenje i korištenje STL-a. Ili [www.cplusplus.com/reference/](http://www.cplusplus.com/reference/) ili općenito svašta ima na [www.stackoverflow.com](http://www.stackoverflow.com)

# Povijest programskog jezika C++

- Programski jezik C++ nastao je razvojem jezika C i zadržava visoku kompatibilnost s njime.
- Kreator jezika je **Bjarne Stroustrup** ([www.stroustrup.com](http://www.stroustrup.com)). Razvoj je započeo 1979. g.; ime C++ skovano je 1983. godine, a prva komercijalna implementacija pojavila se 1985.
- Između 1985. i 1989. jezik je doživio veće inovacije: zaštićeni članovi, parametrizirane klase, višestruko nasljeđivanje itd.
- Jezik je 1997. godine definiran ISO standardnom. To je prva standardizacija jezika.
- Godina 2003. — manja revizija prvog standarda.
- Godina **2011.** — novi standard s brojnim novostima i poboljšanjima.
- Godina 2014. — Revizija standarda iz 2011. godine.
- Godina 2017. — Revizija standarda iz 2014. godine.
- Godina 2020. — Novi standard.

Za informacije o standardu vidjeti: [isocpp.org/std/](http://isocpp.org/std/)

Na ovome kolegiju ćemo raditi po standardu 2011, a novosti će se raditi na kolegiju *Napredni C++* profesora Juraka (zimski semestar). Također na stranici [web.math.hr/nastava/opepp/old/](http://web.math.hr/nastava/opepp/old/) možete naći stare materijale prof. Juraka.

# Karakteristike jezika

- Jezik opće namjene visoke efikasnosti.
- Omogućava programiranje niske razine i programiranje s apstrakcijama.
- Slijedi princip da apstrakcija ne smije negativno utjecati na efikasnost koda.

# Prevodioci

## Linux

- [GCC](#)
- [Clang](#)

## Microsoft Windows

- [Microsoft Visual C++](#)

Ukoliko netko nema mogućnost instaliranja kod kuće niti jednog prevodioca, napraviti ćemo neki serveru (npr. [op.math.hr](http://op.math.hr)), na kojem će biti instaliran neki noviji g++ prevodioc.



# Softverske biblioteke

- Prevodilac za C++ dolazi uvijek zajedno sa standardnom bibliotekom koja se kratko naziva **STL** (*standard template library*).
- Druga najznačajnija softverska biblioteka za jezik C++ je [www.boost.org](http://www.boost.org). Boost je na određen način ekstenzija standardne biblioteke.

Standardna biblioteka ne uključuje biblioteku za konstrukciju grafičkog sučelja. S druge strane, postoji čitav niz različitih C++-biblioteka za konstrukciju grafičkog sučelja. Na primjer:

- [Qt](#)
- [WxWidgets](#)
- [Fltk](#)
- [Juce](#)
- [Ultimate++](#)

# Integrirane okoline za razvoj programa

Integrirane okoline za razvoj programa nude editor za pisanje programa i različite alate za prevođenje, analizu, ispravljanje programa itd. One često integriraju i sustav za upravljanjem kodom i sustav za distribuciju koda.

- **Visual Studio:** [msdn.microsoft.com/en-us/vstudio](https://msdn.microsoft.com/en-us/vstudio)
- Visual Studio Express: [visualstudio.microsoft.com/vs/express/](https://visualstudio.microsoft.com/vs/express/)
- Qt Creator: [www.qt.io](https://www.qt.io)
- **Visual Studio Code:** [code.visualstudio.com](https://code.visualstudio.com)
- Code::Blocks: [www.codeblocks.org](https://www.codeblocks.org)
- NetBeans IDE: [netbeans.org](https://netbeans.org)
- Eclipse CDT: [www.eclipse.org/cdt](https://www.eclipse.org/cdt)
- CodeLite: [www.codelite.org](https://www.codelite.org)

# Objektno orijentirano programiranje

- U centru interesa softverskog inženjerstva su složene aplikacije. Problemi:
- Održivost — mogućnost lakog održavanja koda.
- Proširivost — mogućnost lakog dodavanja nove funkcionalnosti.
- Prenosivost — mogućnost prijenosa koda na druge platforme bez velikih izmjena.

# Tehnike programiranja:

## **Proceduralna.**

- Razlaganje programa na manje cjeline — implementacija pomoću funkcija, procedura.
- Nemogućnost definiranja apstrakcije koncepata iz aplikacijske domene.
- Ne nudi podršku za postizanje održivosti i proširivosti.

## **Objektno orijentirana.**

- Klase predstavljaju koncepte iz aplikacijske domene.
- Program se sastoji od objekta (instance klase) i njihove komunikacije.
- Objekti imaju punu odgovornost za svoje ponašanje.

# Osnovni elementi OO tehnike:

- *Apstrakcija* — Koncepti iz aplikacijske domene se reprezentiraju neposredno u programu, a izvedbeni detalji su skriveni iza sučelja (eng. interface) koje reprezentira koncept.
- *Enkapsulacija* — Sposobnost osiguravanja da se apstrakcija koristi prema svojim specifikacijama. Enkapsulacijom se sprečava narušavanje apstrakcije, odnosno prodiranje implementacijskih odluka izvan granica apstrakcije.
- *Polimorfizam* — Skrivanje različitih implementacija iza istog sučelja. Osigurava široku primjenjivost koda i reducira zavisnost o implementaciji.
- *Naslijeđivanje* — Konstrukcija novih apstrakcija polazeći od već postojećih.

# Generičko programiranje

- Pojam generičkog programiranja odnosi se na generalizaciju softverskih komponenti kako bi se lako mogle koristiti u različitim situacijama. Osnovni elementi generičkog programiranja u C++-u su *parametrizirane klase i funkcije*.