

impera

Objektorienterad programmering med C#

Kursträff 1 - Föreläsning - Kursintroduktion

Lärare

- Mevludin Memedi (kursansvarig)
 - mevludin.memedi@oru.se
 - Rum: N4055
- Kalle Räisänen
 - kr@impera.se
- Niklas Magnusson
 - nm@impera.se
- Amanuenser

Kursens huvudsakliga innehåll

- Utifrån det objektorienterade synsättet på programmering konstrueras, implementeras och analyseras algoritmer och objektorienterade datastrukturer i relation till olika givna problems karaktär. Inom kursens ramar behandlas även utformningen av objektorienterade och komponentbaserade programsystem och dess systemarkitektur.
- Nyckelbegrepp för de moment som fokuseras är: arv, inkapsling, polymorfism, återanvändning, I/O-hantering (vilket även innefattar filhantering och serialisering), undantagshantering, händelsehantering, strängar, arrayer och samlingar samt grafiskt gränssnitt.
- Fundamentala termer som frekvent används inom kursen är: klass, objekt, interface (gränssnitt), metod, medlemsvariabel, synlighetsmodifierare, överlagring, överskuggning, referens, datatyp, operator, indentering, komponent, formulär och läsbarhet.

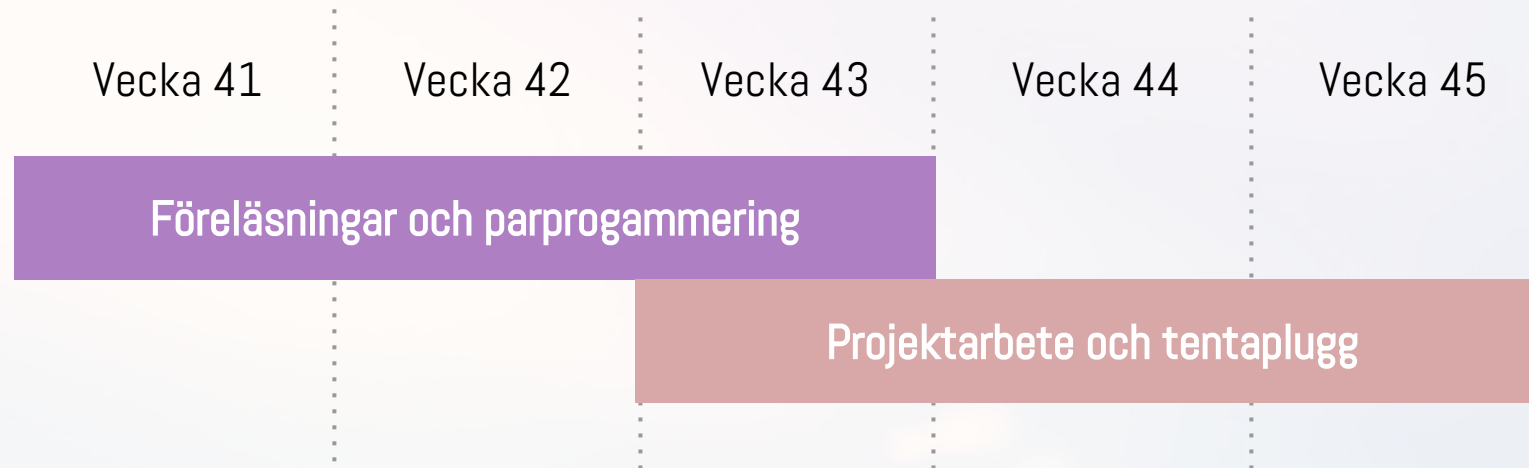
Kursens mål

- Den studerande skall efter avslutad kurs ha ökade kunskaper om
 - objektorienterade programmerings principer: arv, inkapsling, polymorfism, återanvändning
 - generella objektorienterade datastrukturer
 - komponentbaserad systemarkitektur.
- Efter avslutad kurs skall den studerande med ökad självständighet kunna
 - jämföra och prioritera alternativa programmeringstekniska lösningar
 - implementera generella objektorienterade datastrukturer i ett objektorienterat programspråk
 - implementera komponentorienterade lösningar i ett objektorienterat programspråk
 - söka, tyda och använda programmeringsdokumentationer (APIs).
- Efter avslutad kurs har studenten ökad förmåga att
 - tillämpa objektorienterade programmerings principer i programutvecklingsprojekt
 - tillämpa generella objektorienterade datastrukturer i programutvecklingsprojekt
 - tillämpa komponentorienterad utveckling i ett programutvecklingsprojekt.

Examinationsformer

- *Skriftlig tentamen*, 6 högskolepoäng. (Provkod: 0110)
Individuell skriftlig salstentamen
 - Som betyg används Underkänd (U), Godkänd (G) eller Väl Godkänd (VG)
- *Projektuppgift*, 1,5 högskolepoäng. (Provkod: 0120)
Redovisning av en obligatorisk projektuppgift i grupp
 - Som betyg används Underkänd (U) eller Godkänd (G)

Upplägg på övergripande nivå



- Litteratur

- Anne Boehm och Joel Murach. *Murach's C# 2015*. ISBN 978-1-890774-94-3

- Schema

- <http://schema.oru.se/setup/jsp/Schema.jsp?startDatum=idag&intervallTyp=m&intervallAntal=6&sprak=SV&sokMedAND=true&forklaringar=true&resurser=k.IK202G-21029H18->

Kurslitteraturen

Content

Introduction xix

Section 1 An introduction to Visual Studio

Chapter 1	How to get started with Visual Studio	3
Chapter 2	How to design a Windows Forms application	35
Chapter 3	How to code and test a Windows Forms application	57

Section 2 The C# language essentials

Chapter 4	How to work with numeric and string data	99
Chapter 5	How to code control structures	137
Chapter 6	How to code methods and event handlers	163
Chapter 7	How to handle exceptions and validate data	187
Chapter 8	How to use arrays and collections	215
Chapter 9	How to work with dates and strings	259
Chapter 10	More skills for working with Windows forms and controls	287
Chapter 11	How to debug an application	329

Section 3 Object-oriented programming

Chapter 12	How to create and use classes	361
Chapter 13	How to work with indexers, delegates, events, and operators	409
Chapter 14	How to work with inheritance	439
Chapter 15	How to work with interfaces and generics	477
Chapter 16	How to organize and document your classes	507

Section 4 Database programming

Chapter 17	An introduction to database programming	527
Chapter 18	How to work with data sources and datasets	559
Chapter 19	How to work with bound controls and parameterized queries	601
Chapter 20	How to use ADO.NET to write your own data access code	637

Section 5 More skills for working with data

Chapter 21	How to work with files and data streams	679
Chapter 22	How to work with XML files	701
Chapter 23	How to use LINQ	723
Chapter 24	How to use the Entity Framework	753

Section 6 Enhancement and deployment

Chapter 25	How to enhance the user interface	799
Chapter 26	How to deploy an application	823

Reference aids

Appendix A	How to install and use the software and files for this book	855
Index		863

Aktivitetsschema - vecka 41

- **KT1:** *.NET, C# och Visual Studio* [Kap. 1, 5]
 - FL
 - Kursintroduktion
 - Vad är C#?
 - Vad är .Net?
 - LK: Visual Studio och grunder i C#
 - PP/PW: Komma igång med VS
 - Skapa och köra ett projekt
- **KT2:** *Grunder i C# + Debugger* [Kap. 4, 5, 11]
 - FL/LK
 - Mer grunder i C#
 - Debugga C#
 - PW/PP: Använda debuggern
- **KT3:** *Arrayer och samlingar* [Kap. 8, 9]
 - FL/LK
 - Arrayer och samlingar
 - Mer om strängar
 - PP: Arrayer, samlingar, strängar
- **KT4:** *Repetition + WinForms* [Kap. 2, 3]
 - FL: WinForms
 - LK: Repetition
- **KT5:** *Grunderna i C#* [Kap. 1-5, 8, 9]
 - PP: Grunderna i C#

Aktivitetsschema - vecka 42

- **KT6:** *Arv, interfaces, generics, design patterns* [Kap. 12, 14-16]
 - FL/LK
 - Arv, interfaces, polymorfism
 - Generics
 - Design patterns, SOLID, clean code
- **KT7:** *Arv, interfaces, generics (forts)*
 - Parprogrammering
- **KT8:** *Exception-handling, I/O* [Kap. 7, 21]
 - FL/LK
 - Exceptions, try/catch
 - using
 - I/O: Läs och skriva filer
 - PP: Exceptions + I/O
- **KT9:** *Repetition*
 - LK
 - Exempelprojekt som sammanfattar det vi gått igenom hittills
- **KT10:** *XML, serialisering, generics* [Kap. 15, 22]
 - FL/LK
 - XML och serialisering
 - Fördjupning kring generics
 - PP: XML och serialisering
- **KT11:** *Events, delegat, LINQ* [Kap. 6, 13, 23]
 - FL/LK
 - Events och delegat
 - Introduktion till LINQ
 - PP: Events, delegat och LINQ

Aktivitetsschema - vecka 43

- **KT12: *Threading + async***
 - FL/LK
 - Threading och async
 - Mer om delegat
 - PP
 - Threading + async
- **KT13: “Avancerad C#” [Kap. 13, 23]**
Projektpresentation
 - FL/LK
 - Fördjupning kring LINQ
 - Operator overloading, indexers, extension methods
 - PP
 - LINQ, async, Extension methods
- **KT14: *C# i verkligheten***
 - FL/LK: Hur vi använder C# i skarpt projekt
- **KT15: *Repetition***
 - FL/LK
 - Sammanfattning
 - Era frågor!

Aktivitetsschema - vecka 44-45

- **Eget arbete med projektet**
 - Inlämning tis 6 nov
- **Tentamen**
 - Fre 9 nov

impera

Objektorienterad programmering med C#

Kursträff 1 - Föreläsning - Introduktion till C#, .Net och Visual Studio

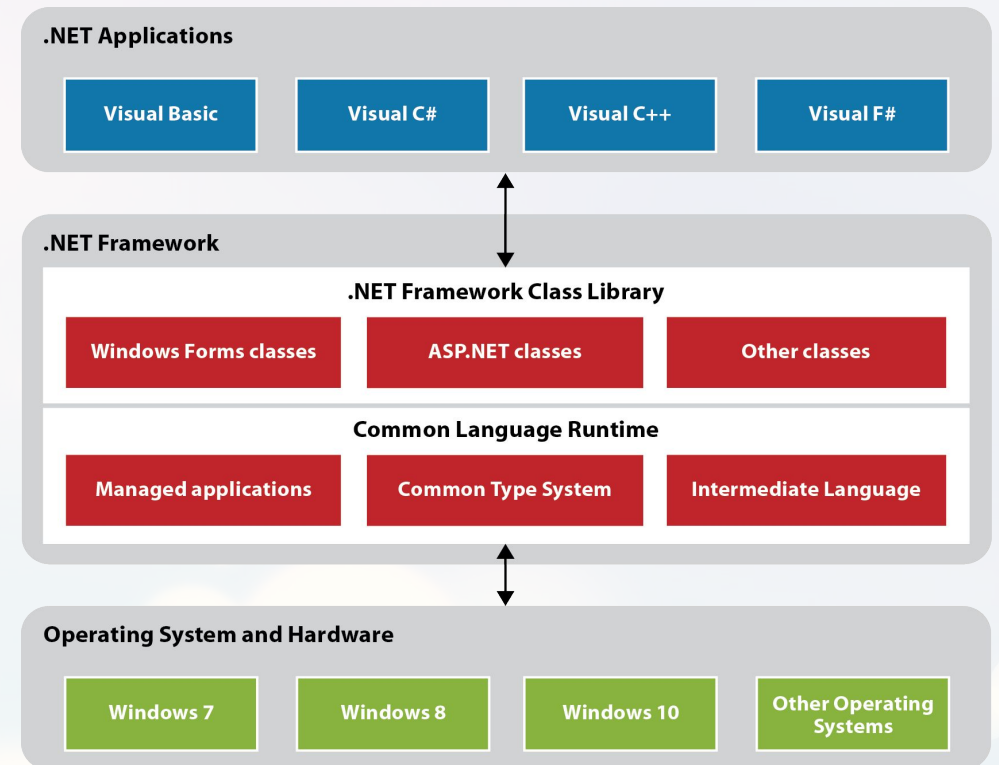
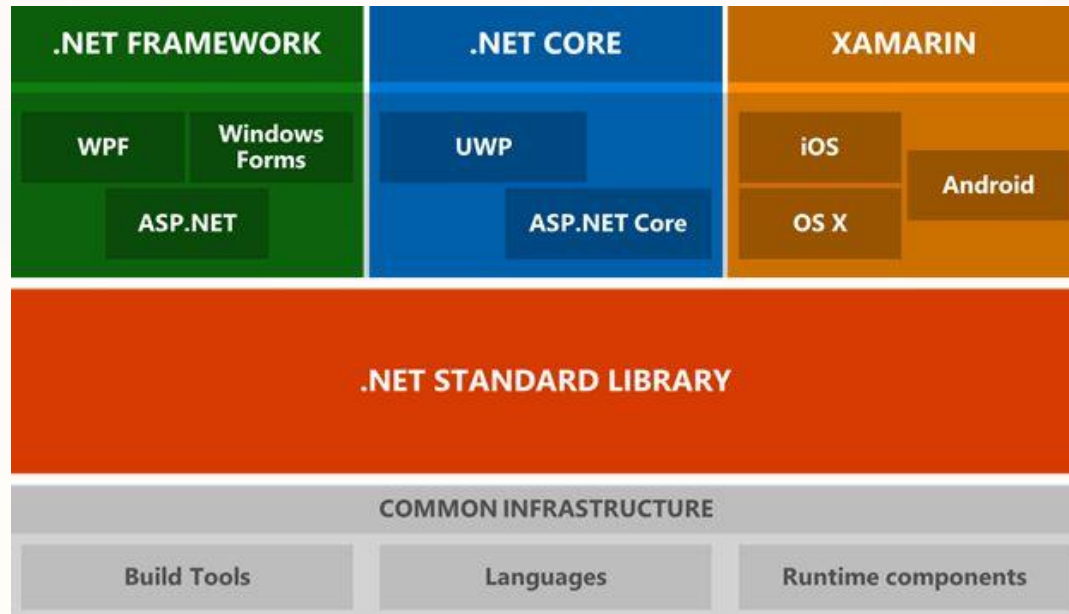
C#

- Microsoft, 2000
- Ett objektorienterat språk som liknar C++ och Java
- Objektorienterat: Inkapsling, Arv, Polymorfism
- Inslag av funktionell programmering

```
namespace ORU.Math.Geometry {  
    class Point {  
        public double X {get; set; }  
        public double Y {get; set; }  
  
        public Point(double x = 0m, double y = 0) {  
            X = x;  
            Y = y;  
        }  
  
        public double Distance() {  
            return Math.sqrt((X * X) + (Y * Y));  
        }  
    }  
}
```

.Net

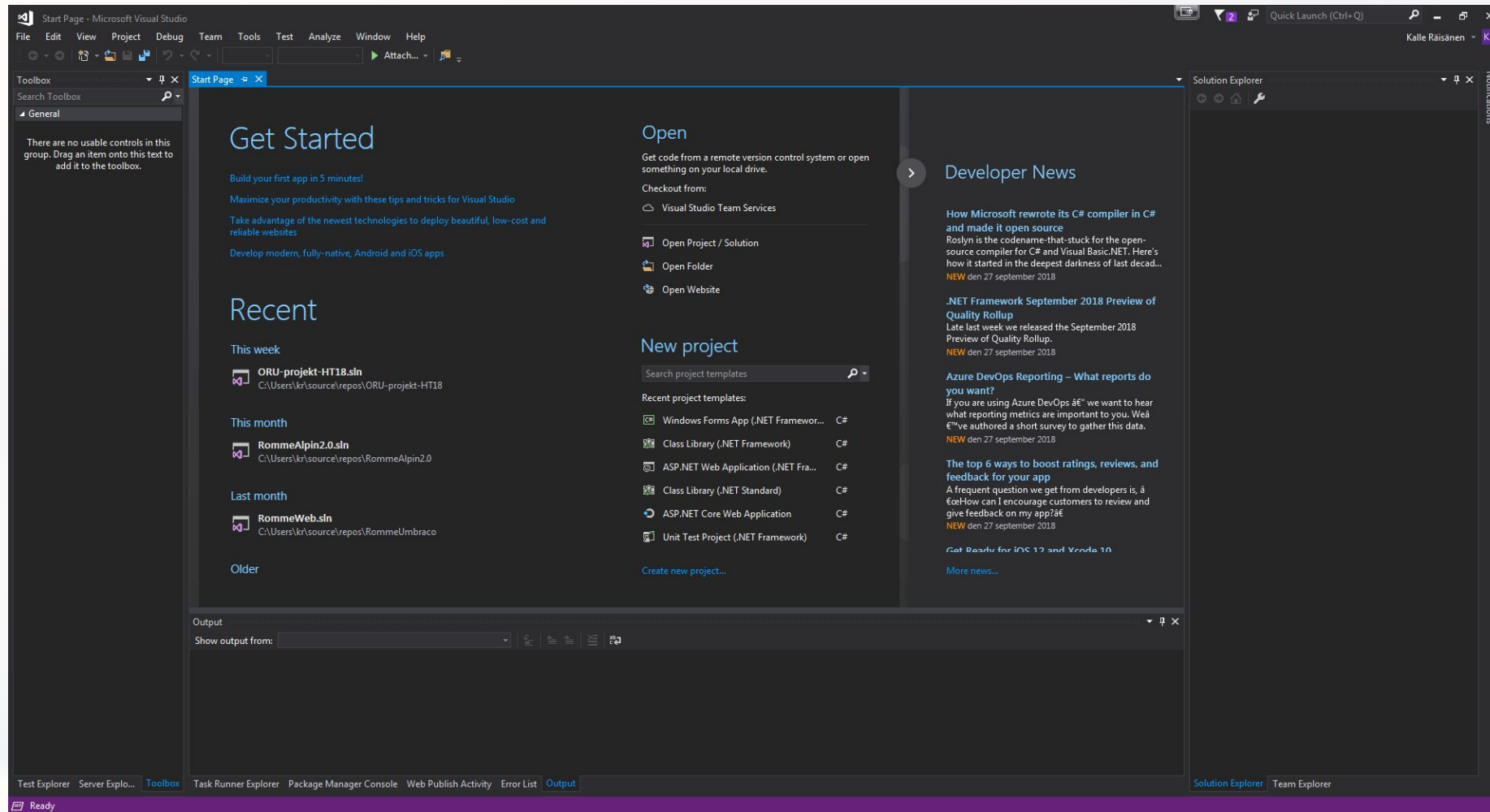
- Ett ramverk som generaliserar operativsystemfunktioner



Visual Studio

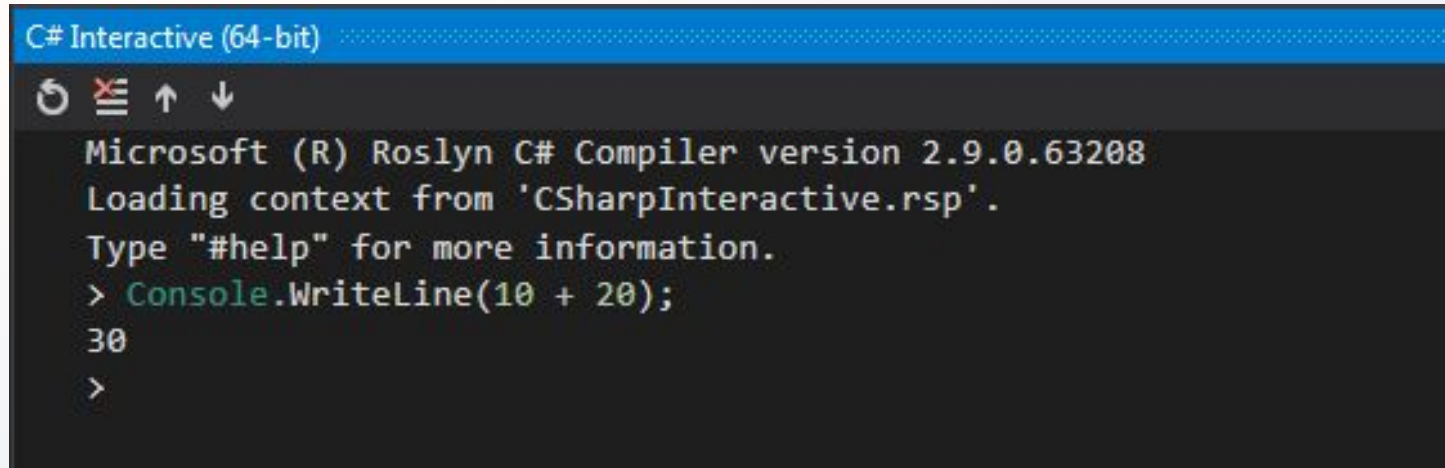
Supported Features	Visual Studio Community	Visual Studio Professional	Visual Studio Enterprise
⊕ Supported Usage Scenarios	●●●○	●●●●	●●●●
Development Platform Support ²	●●●●	●●●●	●●●●
⊕ Integrated Development Environment	●●●○	●●●○	●●●●
⊕ Advanced Debugging and Diagnostics	●●○○	●●○○	●●●●
⊕ Testing Tools	●○○○	●○○○	●●●●
⊕ Cross-platform Development	●●○○	●●○○	●●●●
⊕ Collaboration Tools and Features	●●●●	●●●●	●●●●

Visual Studio



C# interactive

View → Other Windows → C# Interactive



```
C# Interactive (64-bit)
Microsoft (R) Roslyn C# Compiler version 2.9.0.63208
Loading context from 'CSharpInteractive.rsp'.
Type "#help" for more information.
> Console.WriteLine(10 + 20);
30
>
```

C#-länkar

- [Microsoft Programming Guide C#](#)
- [Microsoft Virtual Academy](#)
- [Stack Overflow](#)
- [VisualStudio](#)

C# - grunder

```
namespace ORU.Auth {  
    using System;  
  
    class User {  
        public string Username { get; set; }  
        public string Password { get; set; }  
  
        public User(string username, string password) {  
            this.Username = username;  
            this.Password = password;  
        }  
  
        public bool IsRightPassword(string password) {  
            return this.Password == password;  
        }  
    }  
}
```

Livekodning

- VisualStudio
 - a. C# Interactive-fönstret
 - b. Skapa projekt - ConsoleApp (.Net framework)

```
Console.WriteLine("Hi! Who are you?");  
string input = Console.ReadLine();  
if (input.Length > 0) {  
    Console.WriteLine("Nice to meet you, " + input + "!");  
}  
Console.ReadKey();
```

- c. Skapa metod

Parworkshop

- Skapa en ny Console App (.Net framework) i Visual Studio
- Appen ska skriva ut “Do you want to play a game?” och sen vänta på inmatning
 - Om användaren skriver “Tic-Tac-Toe” ska programmet skriva ut “A strange game. The only way to win is not to play.”
 - Annars ska programmet skriva ut “I’m sorry, I don’t know that game”.
- Programmet ska sen vänta på en knappnedtryckning innan det avslutas

Tips

- `Console.WriteLine("...")`
- `string input = Console.ReadLine()`
- `if (variabel == "stränginnehåll")`
- `Console.ReadKey()`