

## Övning Arbeta med klasser

Innan övning: öppna webbläsare och skriv in följande URL:

[https://easec.se/c/Modul\\_7.zip](https://easec.se/c/Modul_7.zip) , packa upp arkivet förslagsvis till katalogen C:\Easec.

Övning 1: Skapa och ärv från basklassen User

Arbetsuppgift 1: Skapa abstrakt basklass User

1. Starta Visual Studio, klicka på File – Open – Project/Solution.
2. Bläddra till C:\Easec\Modul\_7\Exercise 1, klicka på GradesPrototype.sln och sedan på Open.
3. I Visual Studio, klicka på View – Task List.
4. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 1a: Create the User abstract class with common functionality for Teachers and Students.
5. I code editor, klicka i slutet av kommentar, klicka på Enter och skriv in följande:

```
public abstract class User  
{
```

6. Klicka på slutet av sista kommentar i blocket (innan deklaration för Grade-klass), klicka på Enter.
7. Skriv in följande kod:

```
}
```

8. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 1b: Add the UserName property to the User class.
9. I code editor, klicka på slutet av kommentar, klicka på Enter och skriv in följande kod:

```
public string UserName { get; set; }
```

10. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 1c: Add the Password property.

11. I code editor, klicka på slutet av kommentar, klicka på Enter och skriv in följande kod:

```
private string _password =
    Guid.NewGuid().ToString(); // Generate a random
password
by default
public string Password
{
    set
    {
        _password = value;
    }
}
```

12. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 1d: Add the VerifyPassword method to the User class.

13. I code editor, klicka på slutet av kommentar, klicka på Enter och skriv in följande kod:

```
public bool VerifyPassword(string pass)
{
    return (String.Compare(pass, _password) == 0);
}
```

**Arbetsuppgift 2: Modifiera klasserna Student och Teacher så dessa ärver från klassen User**

1. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 2a: Inherit from the User class.
2. I code editor, klicka på slutet av kommentar, klicka på Enter och skriv in följande kod, som är markerad med fetstil:

```
public class Student: User, IComparable<Student>
```

3. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 2b: Remove the UserName property (now inherited from User)
4. I code editor, ta bort följande kod, under kommentaren:

```
public string UserName { get; set; }
```

5. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 2c: Remove the Password property (now inherited from User)
6. I code editor, ta bort följande kod, under kommentaren:

```
private string _password =
Guid.NewGuid().ToString(); // Generate a random
password
by default
public string Password
{
set
{
_password = value;
}
}
```

7. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 2d Remove the VerifyPassword method (now inherited from User)
8. I code editor, ta bort följande metod, under kommentaren:

```
public bool VerifyPassword(string pass)
{
return (String.Compare(pass, _password) == 0);
}
```

9. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 2e: Inherit from the User class

10. I code editor, modifiera enligt fet stil:

```
public class Teacher: User
```

11. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 2f: Remove the UserName property (now inherited from User)

12. I code editor, ta bort följande programkod, under kommentaren:

```
public string UserName { get; set; }
```

13. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 2g: Remove the Password property (now inherited from User)

14. I code editor, ta bort följande programkod, under kommentaren:

```
private string _password =  
Guid.NewGuid().ToString(); // Generate a random  
password  
by default  
public string Password  
{  
set  
{  
_password = value;  
}  
}
```

15. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 1: Task 2h Remove the VerifyPassword method (now inherited from User)
16. I code editor, ta bort följande metod, under kommentaren:

```
public bool VerifyPassword(string pass)
{
    return (String.Compare(pass, _password) == 0);
}
```

### Arbetsuppgift 3: Kör applikationen och testa att logga på

1. I meny Build, klicka på Build Solution.
2. I meny Debug, klicka på Start Without Debugging.
3. När applikation startas, skriv in vallee i rutan för Username, skriv in password i rutan för Password, klicka på Log on.
4. Verifiera att lista över elever för läraren Esther Alle visas.
5. Klicka på Kevin Liu, verifiera att ett rapportkort visas för Kevin Liu.
6. Klicka på Log off.
7. Skriv in liuk i rutan för Username, skriv in password i rutan för Password, klicka på Log on.
8. Verifiera att ett rapportkort visas för Kevin Liu.
9. Klicka på Log off.
10. Stäng applikationen.
11. Klicka på Close Solution, via meny File.

## Övning 2: Implementera komplext lösenord genom att använda metod som är abstrakt

### Arbetsuppgift 1: Definiera abstrakt metod SetPassword

1. Starta Visual Studio, klicka på File – Open – Project/Solution.
2. Bläddra till C:\Easec\Modul\_7\Exercise 2, klicka på GradesPrototype.sln och sedan på Open.
3. I Visual Studio, klicka på View – Task List.
4. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 2: Task 1a: Define an abstract method for setting the password.
5. I code editor, studera kommentar, i slutet av denna klicka på Enter och skriv in följande:

```
public abstract bool SetPassword(string pwd);
```

6. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 2: Task 1b: Use the SetPassword method to set the password.
7. I code editor, ta bort följande programkod:

```
_password = value;
```

8. Lägg till följande, istället för det som du tog bort:

```
if (!SetPassword(value))
{
    throw new ArgumentException("Password not complex enough", "Password");
}
```

### Arbetsuppgift 2: Implementera metoden SetPassword i klasserna Student och Teacher

1. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 2: Task 2a: Make \_password a protected.

2. I ode editor, modifiera programkod enligt fet stil:

```
protected string _password =
    Guid.NewGuid().ToString(); // Generate a random
    password by default
```

3. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 2: Task 2b: Implement SetPassword to set the password for the student.
4. I code editor, studera kommentar, i slutet av denna klicka på Enter och skriv in följande:

```
public override bool SetPassword(string pwd)
{
    // If the password provided as the parameter is
    at least 6 characters long then
    save it and return true
    if (pwd.Length >= 6)
    {
        _password = pwd;
        return true;
    }
    // If the password is not long enough, then do
    not save it and return false
    return false;
}
```

5. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 2: Task 2c: Implement SetPassword to set the password for the teacher.
6. I code editor, studera kommentar, i slutet av denna klicka på Enter och skriv in följande:

```
public override bool SetPassword(string pwd)
{
    // Use a regular expression to check that the
    password contains at least two
    numeric characters
    Match numericMatch = Regex.Match(pwd, @".*[0-
```

```

9]+.*[0-9]+.*");
// If the password provided as the parameter is
at least 8 characters long and
contains at least two numeric characters then
save it and return true
if (pwd.Length >= 8 && numericMatch.Success)
{
    _password = pwd;
    return true;
}
// If the password is not complex enough, then do
not save it and return false
return false;
}

```

### Arbetsuppgift 3: Sätt lösenord för nya elever

1. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 2: Task 3a: Implement SetPassword method to set the password.
2. I code editor, ta bort programkod efter kommentar och ersätt den med följande:

```

if (!newStudent.SetPassword(sd.password.Text))
{
    throw new Exception("Password must be at least 6
characters long. Student not
created");
}

```

### Arbetsuppgift 4: Ändra lösenord för existerande användare

1. I meny Build, klicka på Build Solution.  
I Solution Explorer, expandera GradesPrototype project, dubbelklicka på MainWindow.xaml.
2. Skrolla ner till rad 27 och studera följande block av programkod:

```

<Button Grid.Column="2" Margin="5"
HorizontalAlignment="Right"
Click="ChangePassword_Click">

```



```
<TextBlock Text="Change Password" FontSize="24"/>
</Button>
```

3. I Solution Explorer, expandera MainWindow.xaml och dubbelklicka på MainWindow.xaml.cs.
4. I code editor, expandera region Event Handlers och lokalisera metoden ChangePassword\_Click.
5. Studera programkod för denna metod:

```
private void ChangePassword_Click(object sender,
EventArgs e)
{
// Use the ChangePasswordDialog to change the
user's password
ChangePasswordDialog cpd = new
ChangePasswordDialog();
// Display the dialog
if (cpd.ShowDialog().Value)
{
// When the user closes the dialog by using the
OK button, the password should have been changed
// Display a message to confirm
MessageBox.Show("Password changed", "Password",
MessageBoxButton.OK,
MessageBoxImage.Information);
}
}
```

6. I Solution Explorer, expandera Controls, dubbelklicka därefter på ChangePasswordDialog.xaml.
7. I Solution Explorer, expandera ChangePasswordDialog.xaml, dubbelklicka därefter på ChangePasswordDialog.xaml.cs.
8. Studera programkod för metoden ok\_click:

```
// If the user clicks OK to change the password,
validate the information that the
user has provided
private void ok_Click(object sender,
RoutedEventArgs e)
```

```

{
// TODO: Exercise 2: Task 4a: Get the details of
the current user
// TODO: Exercise 2: Task 4b: Check that the old
password is correct for the
current user
// TODO: Exercise 2: Task 4c: Check that the new
password and confirm password
fields are the same
// TODO: Exercise 2: Task 4d: Attempt to change
the password
// If the password is not sufficiently complex,
display an error message
// Indicate that the data is valid
this.DialogResult = true;
}

```

9. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 2: Task 4a: Get the details of the current user.

10. I code editor, på tom rad efter kommentar, skriv in följande:

```

User currentUser;
if (SessionContext.UserRole == Role.Teacher)
{
currentUser = SessionContext.CurrentTeacher;
}
else
{
currentUser = SessionContext.CurrentStudent;
}

```

11. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 2: Task 4b: Check that the old password is correct for the current user.

12. I code editor, på tom rad efter kommentar, skriv in följande:

13.

```

string oldPwd = oldPassword.Password;
if (!currentUser.VerifyPassword(oldPwd))
{

```

```

MessageBox.Show("Old password is incorrect",
"Error", MessageBoxButton.OK,
MessageBoxImage.Error);
return;
}

```

14. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 2: Task 4c: Check that the new password and confirm password fields are the same.

15. I code editor, på tom rad efter kommentar, skriv in följande:

```

string newPwd = newPassword.Password;
string confirmPwd = confirm.Password;
if (String.Compare(newPwd, confirmPwd) != 0)
{
MessageBox.Show("The new password and confirm
password fields are different",
"Error", MessageBoxButton.OK,
MessageBoxImage.Error);
return;
}

```

16. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 2: Task Task 4d: Attempt to change the password.

17. I code editor, studera kommentar under denna rad, klicka på slutet av kommentar, klicka på Enter och skriv in följande:

```

if (!currentUser.SetPassword(newPwd))
{
MessageBox.Show("The new password is not
sufficiently complex", "Error",
MessageBoxButton.OK, MessageBoxImage.Error);
return;
}

```

**Arbetsuppgift 5: Kör applikationen och testa funktionalitet för att ändra lösenordet**

1. I meny Build, klicka på Build Solution.
2. I meny Debug, klicka på Start Without Debugging.

3. När applikation startar, skriv in vallee i rutan för Username, skriv in password99 och klicka på Log on.
4. I fönstret The School of Fine Arts, klicka på Change Password.
5. I fönstret Change Password, skriv in password99 i rutan för Old Password, skriv in pwd101 i rutan för New Password. Skriv in pws101 i rutan Confirm. Klicka på OK.
6. Verifiera att meddelandet: The new password is not sufficiently complex, klicka på OK.
7. Skriv in password101 i rutan för New Password, skriv in password101 i rutan Confirm. Klicka på OK.  
Verifiera att meddelandet: Password changed, visas, klicka därefter på OK.
8. Klicka på Log off.
9. Skriv in vallee i rutan för Username, skriv in password101 i rutan för Password, klicka på Log on.
10. Klicka på New Student.
11. I fönstret New Student Details, skriv in Luka i rutan First Name, skriv in Abrus i rutan för Last Name. Skriv in 1234 i rutan för lösenord och klicka på OK.
12. Verifiera att meddelandet Password must be at least 6 characters long. Student not created visas. Klicka på OK.
13. Klicka på New Student.
14. I fönstret New Student Details, skriv in Luka i rutan First Name, skriv in Abrus i rutan för Last Name. Skriv in abcdef i rutan för lösenord och klicka på OK.
15. Klicka på Enroll Student.
16. Verifiera att Luka Abrus visas i fönstret Assign Student.
17. Klicka på Close.
18. Klicka på Log off.
19. Stäng applikationen.
20. I meny File, klicka på Close Solution.

## Övning 3: Skapa skräddarsydd undantagshantering

### Arbetsuppgift 1: Implementera klassen ClassFullException

1. Starta Visual Studio, klicka på File – Open – Project/Solution.
2. Bläddra till C:\Eases\Modul\_7\Exercise 3, klicka på GradesPrototype.sln och sedan på Open.
3. I Visual Studio, klicka på View – Task List.
4. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 3: Task 1a: Add custom data: the name of the class that is full.
5. I code editor, studera kommentar, i slutet av denna klicka på Enter och skriv in följande:

```
private string _className;  
public virtual string ClassName  
{  
    get  
    {  
        return _className;  
    }  
}
```

6. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 3: Task 1b: Delegate functionality for the common constructors directly to the Exception class.
7. I code editor, i slutet av kommentar klicka på Enter och skriv in följande:

```
public ClassFullException()  
{  
}  
public ClassFullException(string message)  
: base(message)  
{  
}  
public ClassFullException(string message,  
Exception inner)  
: base(message, inner)
```

```
{
}
```

8. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 3: Task 1c: Add custom constructors that populate the `_className` field.
9. I code editor, i slutet av kommentar klicka på Enter och skriv in följande:

```
public ClassFullException(string message, string
cls)
: base(message)
{
  _className = cls;
}
public ClassFullException(string message, string
cls, Exception inner)
: base(message, inner)
{
  _className = cls;
}
```

## Arbetsuppgift 2: Kasta och fånga ClassFullException

1. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 3: Task 2a: Set the maximum class size for any teacher.
2. I code editor, i slutet av kommentar klicka på Enter och skriv in följande:

```
private const int MAX_CLASS_SIZE = 8;
```

3. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 3: Task 2b: If the class is already full, then another student cannot be enrolled.
4. I code editor, i slutet av kommentar klicka på Enter och skriv in följande:

```
if (numStudents == MAX_CLASS_SIZE)
```

```

{
// Throw a ClassFullException and specify the
class that is full
throw new ClassFullException("Class full: Unable
to enroll student", Class);
}

```

5. I fönstret Task List, dubbelklicka på arbetsuppgift TODO: Exercise 3: Task 2c: Catch and handle the ClassFullException.
6. I code editor, i slutet av kommentar klicka på Enter och skriv in följande:

```

catch (ClassFullException cfe)
{
MessageBox.Show(String.Format("{0}. Class: {1}",
cfe.Message, cfe.ClassName),
"Error enrolling student", MessageBoxButton.OK,
MessageBoxImage.Error);
}

```

### Arbetsuppgift 3: Bygg och testa lösningen

1. I meny Build, klicka på Build Solution.
2. I meny Debug, klicka på Start Without Debugging.
3. När applikation har startat, skriv in vallee i rutan för Username, skriv in password99 i rutan för Password, klicka på Log on.
4. I fönstret The School of Fine Arts, klicka på New Student.
5. I fönstret för New Student Details, skriv in följande och klicka på OK.

Fält	Värde
First Name	Walter
Last Name	Harp
Password	abcdef

6. Notera: Att ny elev kommer inte att listas i huvudapplikation, detta beror på att elev inte har blivit tilldelad någon klass.
7. I fönstret The School of Fine Arts, klicka på New Student.

8. I fönstret New Student Details, skriv in följande och klicka på OK.

Fält	Värde
First Name	Andrew
Last Name	Harris
Password	abcdef

9. I fönstret The School of Fine Arts, klicka på New Student.

10. I fönstret New Student Details, skriv in följande och klicka på OK.

Fält	Värde
First Name	Toni
Last Name	Poe
Password	abcdef

11. I fönstret The School of Fine Arts, klicka på New Student.

12. I fönstret New Student Details, skriv in följande och klicka på OK.

Fält	Värde
First Name	Ben
Last Name	Andrews
Password	abcdef

13. I fönstret The School of Fine Arts, klicka på Enroll Student.

14. I fönstret Assign Student, klicka på Walter Harp.

15. I rutan Confirm, klicka på Yes.

16. I fönstret The School of Fine Arts, klicka på Enroll Student.

17. I fönstret Assign Student, klicka på Andrew Harris.

18. I rutan Confirm, klicka på Yes.

19. I fönstret The School of Fine Arts, klicka på Enroll Student.

20. I fönstret Assign Student, klicka på Toni Poe.

21. I rutan Confirm, klicka på Yes.

22. I fönstret The School of Fine Arts, klicka på Enroll Student.

23. I fönstret Assign Student, klicka på Ben Adrews.

24. I rutan Confirm, klicka på Yes.



25. Verifiera att meddelandet: Class full: Unable to enroll student:Class: 3C, visas. Klicka på OK.
26. I fönstret Assign Student, klicka på Close.
27. Klicka på Log off.
28. Stäng applikationen.
29. I meny File, klicka på Close Solution.