

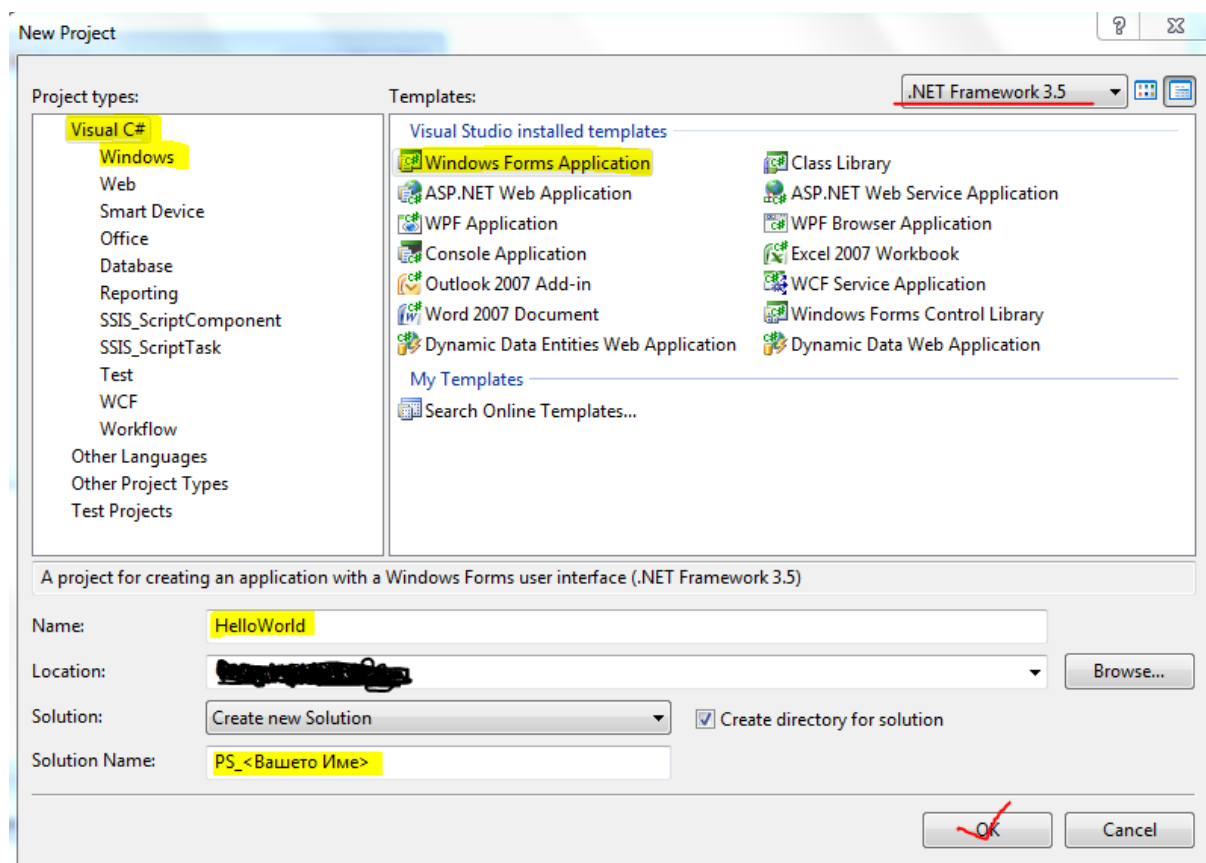
# Упражнение №1 по ПС

*Запознаване със средата на Visual Studio 2008. Основни панели и прозорци. Контроли и събития.*

Целта на това упражнение е да се запознаят студентите със средата за програмиране Visual Studio 2008 и да се научат основните похвати за управление на проекти в нея. Студентите трябва да могат да създават нови проекти, да добавят файлове и нови елементи към тях, да компилират и стартират направените приложения, да проверяват за грешки и да дебъгват.

## Създаване на нов проект.

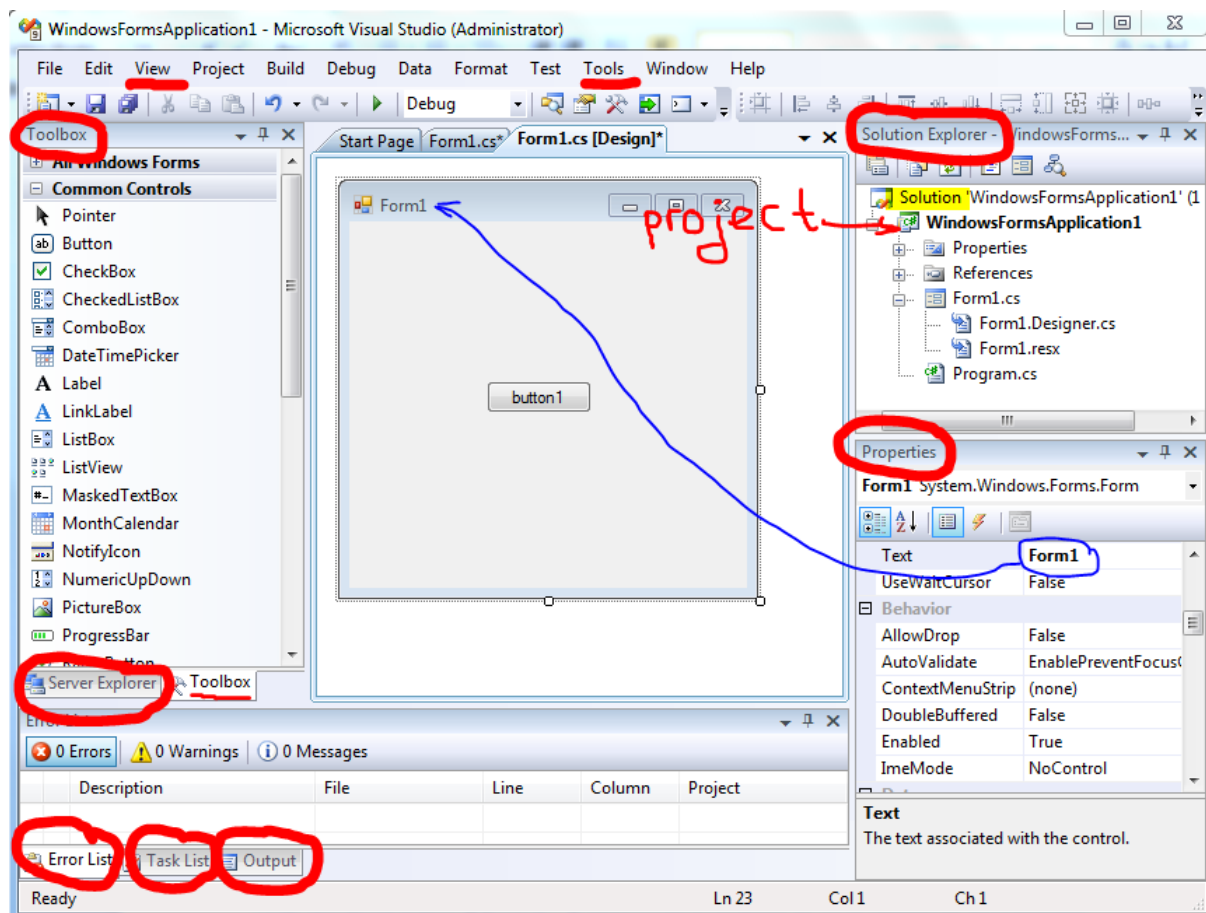
За да създадете нов проект отворете Visual Studio 2008 и от менюто “File” изберете “New” → “Project...”. Отваря се следния диалогов прозорец:



Езикът, с който ще работим, е C#. Ще създаваме приложение с типичните за Windows форми (Windows Forms Application). Кръстете вашия проект HelloWorld, но след това задайте друго име за Solution. Нека името представлява PS\_<№ на група>\_<име>, например PS\_74\_Toni. В Location оставете стандартното място за запис, но го запомнете, тъй като от тук нататък всички часове ще работите само с този ваш Solution. Препоръчително е да си носите собствени USB флаш памет, на които да си записвате проектите, за да е сигурно, че можете да си ги доработвате.

## Прозорци във Visual Studio 2008

След като е отворен един проект вашето Visual Studio 2008 трябва да има приблизително този вид:



По-надолу в упражнението са обяснени основните прозорци от програмната среда на Visual Studio 2008.

**Understanding Solutions & Projects** – В реални ситуации може да ни се наложи да реализираме огромни проекти включващи например няколко различни бази данни и още толкова различни графични интерфейси за отделните потребители. Всичко това може да бъде организирано в едно общо решение “Solution” с определено име (например на компанията). Един “Solution” се състои от проекти “Projects”. Името на проекта се свързва с крайното компилирано приложение. Проекта се състои от определени файлове в зависимост от типа избран при създаването му. Най-общо казано, в проекта ще пишем кода отговарящ за функционалността и интерфейса на нашето приложение. А в един “Solution” можем да имаме няколко отделни проекта.

**ToolBox** – Това е прозорец, в който са разположени всички контроли, които можем да използваме в нашите програми. Добавянето им към една форма става чрез провлачване или двойно щракване с мишката. За удобство контролите са разпределени по категории в зависимост от тяхната употреба.

\* Ако *ToolBox* прозореца е затворен, може да го отворите от меню „View“.

Добавете един бутон към вашата празна форма. Разгледайте и другите контроли. Добавяйте и изтривайте контроли за да видите техния вид.

Solution Explorer – Принципно се намира от дясната страна на вашия екран. Той съдържа всички проекти и файлове, с които работим. „Solution Explorer“ е организационен компонент, чрез който можем да управляваме нашите проекти. Всичко, което искаме добавим към нашия проект може да стане, чрез него, с натискане на десен бутон и избор на „Add“. Могат да се добавят нови форми, специализирани файлове, нови папки, потребителски контроли и други.

Освен нови файлове могат да се добавят и вече съществуващи такива чрез избора на „Add Existing Item...“.

Добавете към вашия проект нова папка с име „Notes“. Като изберете папката и върху нея натиснете десен бутон добавете нов текстов файл. Именувайте го „ReadMe.txt“. Той се отваря във вашето работно поле, така че можете да въведете желаните текст в него и да го запазите.

*\*Нещо полезно – ако в Solution Explorer натиснете десен бутон върху името на вашия проект и от контекстното меню изберете „Open Folder in Windows Explorer“, ще ви се отвори реалната директория с вашите файлове на компютъра в Windows Explorer.*

Properties Window – Той съдържа всички свойства на компонентите и контролите в нашия проект, които могат да се зададат в режим на дизайн. Промените направени в Properties прозореца ще се отразят на изгледа на нашата програма и ще са валидни за цялото и съществуване (от момента на стартиране, до затварянето на програмата). Ако искаме да променим някое от тези свойства по време на изпълнение на програмата, например при появата на дадено събитие трябва да напишем ръчно съответния код за промяна като се обърнем към съответното свойство на обекта, който ще променяме.

Добавете бутон към вашата форма. Кликнете два пъти върху бутона, за да се създаде OnClick метода, обработващ събитието „натискане на бутона“. В новосъздадения метод напишете името на вашия бутон (той е обект и най-вероятно се казва button1), след това напишете точка (.) и Intellisense ще ви подсказва всички свойства и методи, които са налични за този обект. Напишете (или изберете) BackColor, това е свойството определящо фона на бутона. За да го смените просто му присвоете нова стойност чрез оператор = и цвят чрез Color.<име на цвят>; .

Свойствата в Properties прозореца могат да се подреждат по 2 начина – по категории или по азбучен ред (обърнете внимание на двата бутона за подредба).

Важно свойство, на което трябва да се обърне внимание, е **Name**. То представлява името, с което ще се обръщаме към обекта в нашия код за да го манипулираме. Трябва да се разграничи от свойството **Text**, което съдържа просто текста изобразяван от контролата. Обикновено по подразбиране двете свойства са с еднакви стойности, но трябва да бъдат променени съобразно употребата на съответната контрола. Свойството Name трябва да

отговаря на изискванията за правилен идентификатор, като допълнително е хубаво той да е съобразен с вида на контролата и нейната роля. Това се прави с цел по-добра четливост на кода и е особено необходимо при съставянето на големи приложения. Пример за стандартно именуване на контроли е ако например имаме бутон “Close”, може да го кръстим btnClose. Ако имаме поле за въвеждане на име (TextBox контрола), е препоръчително свойството Name да има стойност txtName.

В Properties прозореца освен свойствата свързани с един обект можем да видим и всички събития, които могат да се предизвикат или обработят (иконата със светкавица).

Class View – Това е прозорец показващ класовете във вашия проект. Интересното е, че можете да видите дървовидната структура на наследяване на всеки един от класовете. Отваря се от меню “View” → “Class View”.

Server Explorer – Това е прозорец, който ще използваме по-нататък за да направим връзка към база данни или да създадем нова своя SQL база данни. Ако не е отворен, прозореца може да се зареди чрез избирането му от меню “View” → “Server Explorer”.

Tasks Window – Това е прозорец, в който могат да се записват задачи или коментари. Ако напишете коментар започващ с „вълшебната думичка“ TODO, той ще се отрази в прозореца на задачите и така лесно можете да си оставяте съобщения за това какво трябва да се свърши. Ако по един проект работят няколко разработчика, за да се избегнат недоразумения може да се добавят нови думи за отбелязване на специфичен коментар, както и приоритет на коментарите. От менюто “Tools” избирате “Options”, в появилия се диалогов прозорец отляво на раздел “Environment” избирате “Task List” и вдясно избирате приоритет и въвеждате ключовата дума за вашия коментар (например вашите инициали), след което натискате бутона “Add” за да потвърдите.

Пробвайте да добавите няколко коментара от описания вид на различни места в кода. Коментарите се отбелязват със символите //

Output Window – Това е прозорец, в който се извежда информация за процесите случващи се с вашия проект. Всеки път когато се компилира (Build) вашият проект, се извеждат съобщения за завършените процеси, като например къде е създаден .exe файла и дали са възникнали някакви проблеми. За да имаме успех трябва крайното съобщение да е „Build: 1 succeeded or up-to-date, 0 failed“. Ако пред FAILED стои число различно от 0, то значи трябва да поправяме нещо.

От менюто “Build” изберете “Build Solution” за да билднете вашия проект.

Errors Window – Този прозорец се появява автоматично след като при компилация са възникнали грешки. Чрез него лесно можете да видите къде точно се намират те. С двойно натискане върху грешката Visual Studio автоматично ви отвежда при нея. За бележете, че още при въвеждане на кода се правят проверки и неразпознатите неща се подчертават с назъбена линия. Ако отворите прозореца от менюто “View”, можете да се възползвате от моменталната проверка на редовете и да избегнете забравянето на символа ; ☺

\*Номерация на редовете в кода – За да включите номерацията на редовете във вашия код, изберете от менюто Tools → Options... В отворения се диалогов прозорец изберете “Text Editor”, после в дясно в раздел “Display” сложете тикче на “Line Numbers”.

Включете или изключете номерацията на редовете от кода.

\*За повече информация може да разгледате у вас видео уроците от този сайт (<http://www.vtc.com/products/Introduction-To-Microsoft-Visual-Studio-2008-Tutorials.htm>). Има някои разлики, тъй като демонстрациите са на език Visual Basic.

## Вашата HelloWorld програмка

Добавете един бутон към прозореца на вашата програма. Кръстете го **btnHello** и променете текста му на „Здрасти!“.

Натиснете двойно върху бутона, по този начин създаваме неговия OnClick метод.

В новосъздадения метод добавете следния код:

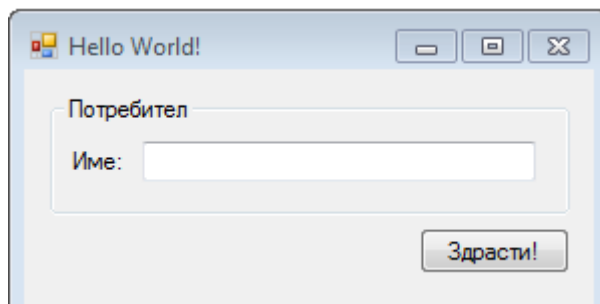
```
MessageBox.Show("Здрасти!!! Това е твоята първа програма  
на Visual Studio 2008!");
```

Стартирайте програмата с **F5** или зеления триъгълник.

Добавете GroupBox контрола от групата Containers, която ще обединява всички контроли с въвеждане на данните за един потребител. Дайте ѝ име gbUser и променете текста ѝ.

В горната контрола добавете Label и TextBox контроли. Именувайте ги подходящо (например lblName и txtName).

Прозореца ви трябва да изглежда по подобен начин:



Модифицирайте кода на бутона btnHello по следния начин:

```
MessageBox.Show("Здрасти "+ txtName.Text + "!!! \nТова е твоята първа програма на Visual Studio 2008!");
```

Стартирайте и наблюдавайте разликата в поведението като въведете текст в полето.

Направете проверка за дължината на текста, въведен от потребителя. Поставете ограничение за дължина на името поне 2 символа. Направете, така че само ако въведената информация отговаря на условията, да се извежда съобщението за поздрав, а в противен случай изведете съобщение с подкана за корекция. (Използвайте оператора "If-else").

Разучете възможностите за Debug, като изпълните стъпка по стъпка вашето приложение, добавите точки на прекъсване и наблюдавайте стойности на променливите. За да има повече код в програмата ви може да добавите необходимите компоненти и код за пресмятане на  $n!$ , където  $n$  се въвежда от потребителя.

Добавете метод, който се изпълнява при изход от програмата и проверява дали потребителя е сигурен дали иска да я затвори.

**Важно:** За следващото упражнение си носете USB флаш памет, за да си запишете проекта. А този може да си го изпратите на пощата, ако има Интернет.