## Лабораторно упражнение № 4

## Класове и обекти в C#

### I. Теоретична част

1. Класове и методи, модификатори на достъп.

Клас в C#, какво и в други езици за програмиране е тип данни, дефиниран от програмиста, който се състои от данни, наричани полета и функции за работа с тези данни, наречени методи.

Общият вид на декларацията на клас в C# е следния:

```
class ИмеКлас
{
    // тяло на клас
}
```

В C# методът е именувана последователност от конструкции. Методите в C# са аналози на функциите в процедурно-ориентираните програмни езици. Тъй като C# е изцяло обектно-ориентиран, на практика всички функции са методи на класовете т.е. езикът не позволява съществуването на външни функции*.

Общият вид на дефиниране на метод е следния:

```
тип ИмеМетод(списък формални параметри)
{
    // тяло на метод
}
```

където:

- тип - тип на върнатия резултат;
- ИмеМетод - идентификатор, определящ еднозначно метода;
- списък формални параметри - параметрите на метода, определени чрез тип и име (идентификатор)

Тялото на метода съдържа блок оператори, които се изпълняват при стартиране на метода. Както и в C и C++ в тялото на функцията-метод трябва да се съдържа оператор **return**, който връща резултата от изпълнението на метода. Типът на върнатата стойност съвпада с типа на метода.

Модификаторите на достъп определят по какъв начин членовете на дадения клас са видими. Чрез тях се реализира една от основните концепции в обектно-ориентираното програмиране – капсулация (encapsulation). Според този принцип, част от членовете на класа могат да бъдат видими само за методи, принадлежащи на класа и скрити за останалите.

---

* Под понятието „външна функция” се разбира функция, която е дефинирана на глобално ниво и не принадлежи на даден клас. Понятието е заимствано от език C++, където е възможно дефиниране на функция на глобално ниво.

Модификаторите на достъп в език C# определят следното:

- **public** – определя, че членовете са достъпни от всякъде от където се вижда класът.
- **private** – определя, че даденият член достъпен само за класа, в който е дефиниран. За членове на клас модификатор **private** е по подразбиране т.е. ако бъде пропуснат модификатор пред дефиницията на метод, например, то този метод е достъпен само за останалите методи на класа.
- **protected** – членът на дадения клас е достъпен както за класа, в който е определен, така и за производните класове. За останалите е недостъпен.
- **internal** – членът на дадения клас е достъпен за всички класове в множеството, за което е определен. Извън него е недостъпен.
- **protected internal** – определя, че членът на дадения клас е достъпен за собствените методи на класа и само за методите на тези наследени класове, които са в множеството.

**2. Дефиниране на програмни обекти (инстанции на класа). Извикване на методи**

Тъй като класът е референтен тип данни, създаването на обект-екземпляр на класа е чрез оператор new.

Общият вид на дефиницията на обект е:

**ИмеКлас имеОбект = new имеКлас();**

В C# има два типа методи статични методи и методи на екземплярите. Методите на екземплярите задължително изискват обект чрез който да бъдат извикани. В случая се използва един от най-често срещаните оператори в обектно-ориентираното програмиране - оператор . (точка). Общият вид на извикването на метод е следния:

**имеОбект.ИмеМетод(списък фактически параметри);**

Особеностите, дефинирането и извикването на станичните методи ще бъдат разгледани в една от следващите теми.

**II. Задачи за изпълнение**

1. Да се създаде конзолно приложение, с което се декларират класове описващи следните фигури: правоъгълник, триъгълник и кръг. Всеки един от тези класове да съдържа методи за въвеждане стойностите на полетата, извеждане стойностите на полетата и намиране лицето на фигурата. Да се създаде демонстрационна програма, с която се дефинират обекти от класовете и се намират лицата на фигурите.

2. Да се декларира клас, описващ точка в равнината чрез нейните координати x, y. Класът да съдържа следните методи:

- въвеждане на стойности за координатите;
- извеждане на стойностите на координатите;
- транслация на точка;
- ротация на точка;
- мащабиране на точка;

Упътване:

*Транслацията на точка е графична трансформация, при която точка с координати (x,y) се премества в позиция (x+Vx, y+Vy). Стойностите Vx,Vy определят дължината на вектора на транслация по осите x и y, съответно. Методът, предназначен за транслация на точка да изисква като параметри Vx и Vy, определящи вектора на транслация.*

*Ротацията на точка е графична трансформация, при която точката се завърта под определен ъгъл спрямо началото на координатната система. Новите координати на точката (x1, y1) се пресмятат по формулата:*

$x1 = x.Cos\ \alpha – y.Sin\ \alpha$

$y1 = x.Sin\ \alpha + y.Cos\ \alpha$

*като α е ъгълът на завъртане.*

*В програмната реализация ъгълът на завъртане се предава като параметър на метода.*

*Мащабирането е графична трансформация, при която точка с координати (x,y) се премества в позиции (x*kx, y*ky), като kx и ky са мащабни коефициенти по осите x и y. Методът за мащабиране да изисква като параметри тези мащабни коефициенти.*

2. Като се използва създадения клас, да се напише програма, която да въвежда три точки в равнината и дава възможност за извършване на графични трансформации върху всяка една от точките
3. Да се създаде конзолно приложение на C#, с което се декларира клас, описващ студент с трите имена, факултетен номер и специалност. Класът да съдържа методи за въвеждане и извеждане стойностите на полетата и метод, който отпечатва на екрана имената на студент по въведен факултетен номер. Да се създаде демонстрационна програма за работа с класа.
4. Да се създаде конзолно приложение на C#, с което се декларират класове описващи точка (Point) и правоъгълна област (Rectangle), описана чрез координатите на горния ляв ъгъл и размерите на областта. Класовете да съдържат методи за въвеждане и извеждане стойностите на полетата. Клас Rectangle да съдържа метод, които определя дали дадена точка попада в областта. Да се създаде демонстрационна програма за работа с класовете.