Object Orietned Programming

Primitive data types – variable ktore store’uja pojedyncze proste wartosci (byte, int, float, Boolean, double, char).

The Structure, The Array – pozwalaja na store’owanie wielu kawałkow jakiś informacji. Strukutra pozwala na store’owanie roznych typow danych, arrayka umozliwia store’owanie jeden konrketny typ danych. Struktury maja jednak ten problem ze nie można w ramach ich definiować funkcji.

Object – instancja klasy.  
klasy – templatki dla obiektów.  
klasa rycerz - to definicja metod w ramach niej, propertiesów, ale bez inicjalizacji, tak ze kazda instancja klasy będzie miała inne wartości dla tych zmiennych. Klasa wiec reprezentuje wszystkie instancje (obiekty) jej pochodzenia. Gdzie Obiket to pojedyncza instancja klasy.

# Enkapsulacja Encapsulation

Enkapsulacja odnosi się do łączenia danych z metodami, tak ze dane mogą być wykorzystywane w ramach klasy. Czyli innymi słowy dane zamieszczamy w klasie i zabogiegamy by cokolwiek spoza klasy moglo działać z tymi danymi (nie oznacza to jednak ze nie mogą wchodzić w itnerakcje z innymi atrybutami innego obiektu). Działanie na membersrach innych klas może być poprzez atrybuty innego obiektu i jego metody.  
W ramach atrybutow klasy chcemy zdefiniować tez readonly tak żeby możliwy był tylko getter na taki atrybut bez setera. Zmienna mogą być tylko referowane wyświetlane nie zmieniane.

Dobra praktyka zakąłda ze zewnętrzne klasy nie powinny bezpośrednio edytować atrybutów obiektu. Czyli każdy kawałek powinien nie mieć dostępu/ polegac na wewnętrznych procesach w innych sekcjach kodu.

Podusmowujac enkapsulacja – umozliwa kontrole dostępu do danych i zabogiega by program konczy się w jakiś niechainych dziwnych stanach

# Abstrakcja

Abstrakcja – mowi o pokazywaniu tylko ogolnych możliwości bez detalu i zbędnych komplikacji. Użytkownicy klas nie powinni być niepokojeni przez wewnętrzne detale tych że klas. Klasy nie powinny bezpośrednio wchodzić w interakcje z danymi innych klas. Interfejsy odpowiadają na sposób prezentacji w jakie sekcje kodu komunikują się z innymi, osiągalne jest to poprzez metody których klasy są w stanie dostępu.

Stworzenie interfejsow przez kotre klasy mogą wchodzić w interakcje zapewnia ze każdy kawałek może być indywidualnie rozwijany.

# Inheritance

Dzeidziwczenie atrybutow/ metod z jednej klasy w innej klasie. Tworzenie heirarchi klas gdzie klasy sa od siebie zależne i posiadaja odnośniki.

W ramach dziedziczenia istotne sa access modifiery kotre zmieniają dostep do klas, metod i atrybutow. Mamy 3 modifiery: public, private, protected. Public – może być dostępny gdziekolwiek z programu. Private – Możliwy dostep tylko w ramach klasy której członkiem jest. Protected – pozwala na dostep w ramach subklas dla danej klasy.

# Polymorphism

Metody który mogą przyjmować wiele form. Mamy dwa typy polimoryfzmu:

dynamiczna – wystepuje podczas runtime’u programu. Wtedy metody szerują taką sama nazwe ale rozne implementacje. Impelemtnacje superclassy sa nadpisywane przez subclassy. Jeżeli istnacja jest istancja sueprclassy to wywolywana będzie metoda z superklasy, jak jest istnacja subklasy to z subklasy. To co jest wywoływane jest ustalane dynamiczne podczas działania programu.

Statyczny poliforyzm - wystepuje podczas czasu komplikowania programu niż podczas runtime’u. Odnosi się do wielu metod które maja takie same nazwy ale rozne argumenty w ramach tej samej klasy. Jest to znane jhako overloading. Sygnatury takich metod sa inne mimo ze nazwy takie same, roznia się argumentami, kolejnoscia itd.