Interfacce

- La classe Account contiene i metodi

```
void deposito(int amount)
boolean check(int amount)
```

- deposito e check sono i *nomi* dei metodi
- `void deposito(int)` è la *signature* del primo metodo
- L'insieme di tutte le signature (di metodi public) di una classe è chiamata *interfaccia* della classe

Progettazione classe

- Dall'insieme di cose che il software dovrà fare, isoliamo alcune cose e definiamo dati e operazioni di una classe
- Esempio
  - Per ogni studente si dovrà calcolare la media degli esami sostenuti
  - Per ciascuno studente dovrà essere possibile conoscere gli esami superati ed il voto riportato per ciascun esame
- Per individuare le classi è possibile ricorrere alla analisi grammaticale del testo
  - Nomi -> classi o attributi
  - Verbi -> operazioni

Progettazione classe

- Esempio
  - Per ogni studente si dovrà calcolare la media degli esami sostenuti
  - Per ciascuno studente si dovrà tenere gli esami sostenuti ed il voto riportato per ciascun esame
- Classi in verde
  - Studente
- Attributi in marrone
  - Esami sostenuti
  - Voto riportato
- Metodi in arancio
  - Calcolare la media
  - Tenere gli esami sostenuti

Classe Studente

- Dati

```
String nome;
String cognome;
int numeroEsami;
String[] esami;
```

- Operazioni

```
void nuovoEsame(String nomeEsame, int voto);
float media();
```

- Uso

```
// s rappresenta uno Studente
s.nuovoEsame("Italiano", 8);  // chiediamo a s di fare una operazione
```

Classe Studente

```
public class Studente {
  private String nome;
  private String cognome;
  ..
  public void nuovoEsame(String nomeEsame, int voto) {
  ..
  }

  public float media() {
  ..
  }
}
```


Creazione Oggetti: new

- Definito un tipo, ovvero una classe (dati + codice)
  - Posso dichiarare variabili di quel tipo

    `Studente s;`
    - `s` è una variabile di tipo `Studente`
- L'allocazione della memoria per il tipo deve essere fatta esplicitamente, ovvero bisogna creare un oggetto
  - Per creare oggetti (ovvero istanze) del tipo `Studente`

    `s = new Studente();`
    - Con `new` chiedo di occupare memoria per l'oggetto
    - `s` tiene il riferimento (ovvero indirizzo) all'oggetto
- L'oggetto creato
  - Contiene i dati (che la classe definisce)
  - Ha il comportamento descritto dalla classe di cui è istanza

Creazione Oggetti

```
Studente s = new Studente();
Studente s2 = new Studente();
```

- Sia s che s2 sono istanze della classe Studente
  - Sono istanze diverse
- Istanze diverse della stessa classe contengono dati diversi

```
Studente s3;
```

- s3 è una variabile di tipo studente
- s3 non contiene nessun riferimento

```
s3 = s2;
```

- s3 contiene il riferimento che è in s2
- Variabili diverse dello stesso tipo possono contenere riferimenti alla stessa istanza

Istanze e variabili

- Creazione di una nuova istanza

```
Account a = new Account();
```

- Le variabili possono contenere riferimenti ad oggetti o valori primitivi
- Un oggetto ha un tipo a runtime
- Una variabile ha un tipo dichiarato
- I metodi possono essere associati ad oggetti o a classi
  - Metodi static
    - Le chiamate a metodi static avviene sulla classe
- Gli attributi static sono attributi della classe

Oggetti

- Gli oggetti possiedono uno stato, insieme di dati, e una interfaccia, ovvero l'insieme di operazioni che offrono
  - I dati sono nascosti
  - L'interfaccia è nota
- Un oggetto reagisce eseguendo del codice quando un altro gli richiede una operazione, ovvero
  - Tramite una invocazione di metodo ad un oggetto
    - Si chiede all'oggetto di eseguire una certa operazione
  - L'invocazione del metodo si fa su una istanza

  s.media() invoca il metodo media dell'istanza s della classe Studente
- Non si pone l'enfasi sulla richiesta di operare sui dati

Modello client-server

- Basarsi su componenti che eseguono un certo numero di attività
  - Non chiamo una funzione di ordinamento su una certa lista, ma chiedo alla lista di ordinarsi
  - Non chiedo di conservare su file dei dati, ma chiedo al file di rendere le informazioni permanenti
- Progettare componenti che possono svolgere delle attività
- I client non conoscono i dati interni di altri componenti (server) ne come i componenti sono realizzati