클래스

- 클래스 정의
- 변수 / 메서드 정의
- 클래스 설계 및 구현
- 캡슐화
- 오버로딩
- 생성자
- 초기화 순서

7-1

클래스 정의

- 클래스 구조

```
[지정자] class 클래스_이름
{
    //멤버 변수 정의
    //생성자 정의
    //메서드 정의
}

public class Student
{
        ...
 }
```

7-2

변수 정의

- **변수 구조**

  [지정자] <자료형> 변수이름 ;

  ```
  public int number;
  public float price = 1500.0F;
  private String name;
  ```

7-3

메서드 정의

- 메서드 구조

```
[지정자] <자료형> 메서드명([매개인자])
{
    ....
     [return 값;]
 }

public void setName(String name){
...
}
public int number(){
     return 0;
}
```

7-4

클래스 설계 및 구현

- 요구사항 정의

학사관리 프로젝트를 하려고 한다.
학생은 학번, 성별, 이름, 전화번호 등의 정보를 가지고 있다.
학생의 자신의 전화번호를 수정할 수 있다.
학생의 자신의 정보를 출력할 수 있다.

7-5

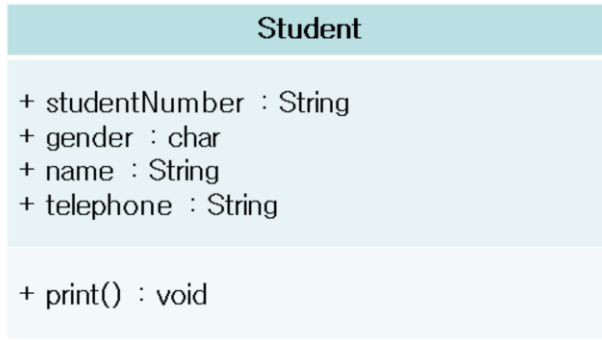
클래스 설계 및 구현

- 멤버 데이터와 동작 추출
  - 클래스
    - 학생
  - 데이터
    - 학번, 성별, 이름, 전화번호
  - 동작
    - 전화번호 수정
    - 학생정보 출력

7-6

클래스 설계 및 구현

- 클래스 설계



7-7

클래스 설계 및 구현

- **클래스 구현 및 실행 결과**

```
>java StudentTest
학번 : 1234
성별 : W
이름 : jeon
전화번호 : 111-222-3333
```

멤버 데이터를 직접 접근 함으로
유효하지 않는 값도 입력 가능하다.

7-8

•클래스 설계 및 구현

•Student.java

```
public class Student {
  public String studentNumber;
  public char gender;
  public String name;
  public String telephone;

  public void print(){
    System.out.println("학번 : "+ studentNumber);
    System.out.println("성별 : "+ gender);
    System.out.println("이름 : "+ name);
    System.out.println("전화번호 : "+ telephone);
  }
}
```

• StudentTest.java

```
public class StudentTest {
  public static void main(String[] args) {
    Student s1=new Student();
    s1.studentNumber="1234";
    s1.gender='W';
    s1.name="jeon";
    s1.telephone="111-222-3333";

    s1.print();

  }
}
```

캡슐화

- 멤버 데이터는 정보 은닉한다.
  - public → private
- 멤버 데이트를 접근하는 메서드를 추가한다.
- 데이터의 값을 초기화하는 setXxx()와 값을 리턴받을 수 있는 getXxx()를 작성한다.
- 메서드를 작성할 때는 접근할 멤버 데이터의 이름을 응용한다.
  - setName(), getName()

7-10

캡슐화

- **클래스 설계**

Student
- studentNumber : String - gender : char - name : String - telephone : String
+ setStudentNumber(studentNumber : String) : void + setGender(gender : char) : void + setName(name : String) : void + setTelephone(telephone : String) : void + getStudentNumber() : String + getGender() : char + getName() : char + getTelephone() : char + print() : void

7-11

•캡슐화

•Student.java

```
public class Student {
  private String studentNumber;
  private char gender;
  private String name;
  private String telephone;

  public String getStudentNumber() { return studentNumber; }
  public char getGender() { return gender; }
  public void setGender(char gender) { this.gender = gender; }
  public void setStudentNumber(String studentNumber) {
    this.studentNumber = studentNumber;
  }

  public String getName() { return name; }
  public void setName(String name) { this.name = name;}
  public String getTelephone() { return telephone;}
  public void setTelephone(String telephone) {
    this.telephone = telephone;
  }

  public void print(){
    System.out.println("학번 : "+ studentNumber);
    System.out.println("성별 : "+ gender);
    System.out.println("이름 : "+ name);
    System.out.println("전화번호 : "+ telephone);
  }
}
```

•캡슐화

•StudentTest.java

```
public class StudentTest {
  public static void main(String[] args) {
    Student s1=new Student();
    s1.setStudentNumber("20010102");
    s1.setGender('W');
    s1.setName("김민성");
    s1.setTelephone("010-222-3333");

    s1.print();

  }
}
```

캡슐화

- **캡슐화 실행 결과**

```
>java StudentTest
학번 : 20010102
성별 : W
이름 : 김민성
전화번호 : 010-222-3333
```

7-14

오버로딩

- 메서드나 생성자가 이름은 같고 매개인자수나 타입이 다르게 정의된 것을 오버로딩(Overliading)이라고 한다.
- 주로 다양한 자료형에 따른 같은 동작을 수행할 때 정의된다.
- 메서드

  print(int)
  print(boolean)
  print(short)

- 생성자

  new String()
  new String(String)
  new String(byte[])

7-15

생성자

- 생성자는 new 뒤에서 한번만 사용된다.
- 클래스에 생성자가 없으면 Default Constructor를 VM이 기본 제공한다. 단 생성자가 1개라도 있으면 기본 제공되지 않는다.
- 생성자는 메서드와 구분하기 위해서 이름은 클래스 이름과 같다.
- 생성자는 리턴타입이 없다.
- 생성자는 초기화 할 때 한번만 호출한다.
- 여러 생성자를 정의할때는 생성자 Overloading해야 한다.

```
[지정자] 클래스명([매개인자]){
}
```

7-16

생성자

- **클래스 설계**

Student
- studentNumber : String - gender : char - name : String - telephone : String

+ Student()
+ Student(studentNumber : String,
gender : char, name : String,
telephone : String) : void
 	
//setter
//getter
 	
+ print() : void

7-17

생성자

- 클래스 구현

```
 public Student(){
      this("",'W',"","");
}
public Student(String studentNumber, char gender, String name,
                String telephone) {

      this.studentNumber = studentNumber;
      this.gender = gender;
      this.name = name;
      this.telephone = telephone;
}
```

7-18

생성자

- 실행 결과

```
Student s1=new Student();
s1.print();
Student s2=new Student("20010102",'M',  "김민규", "010-2222-3333" );
s2.print();
```

```
>java StudentTest
학번 :
성별 : W
이름 :
전화번호 :
학번 : 20010102
성별 : M
이름 : 김민규
전화번호 : 010-2222-3333
```

7-19

초기화

- 객체의 초기화
  - 멤버 데이터의 기본형
  - 명시적인 초기화
  - 생성자에 의한 초기화

7-20

LAB

7-21