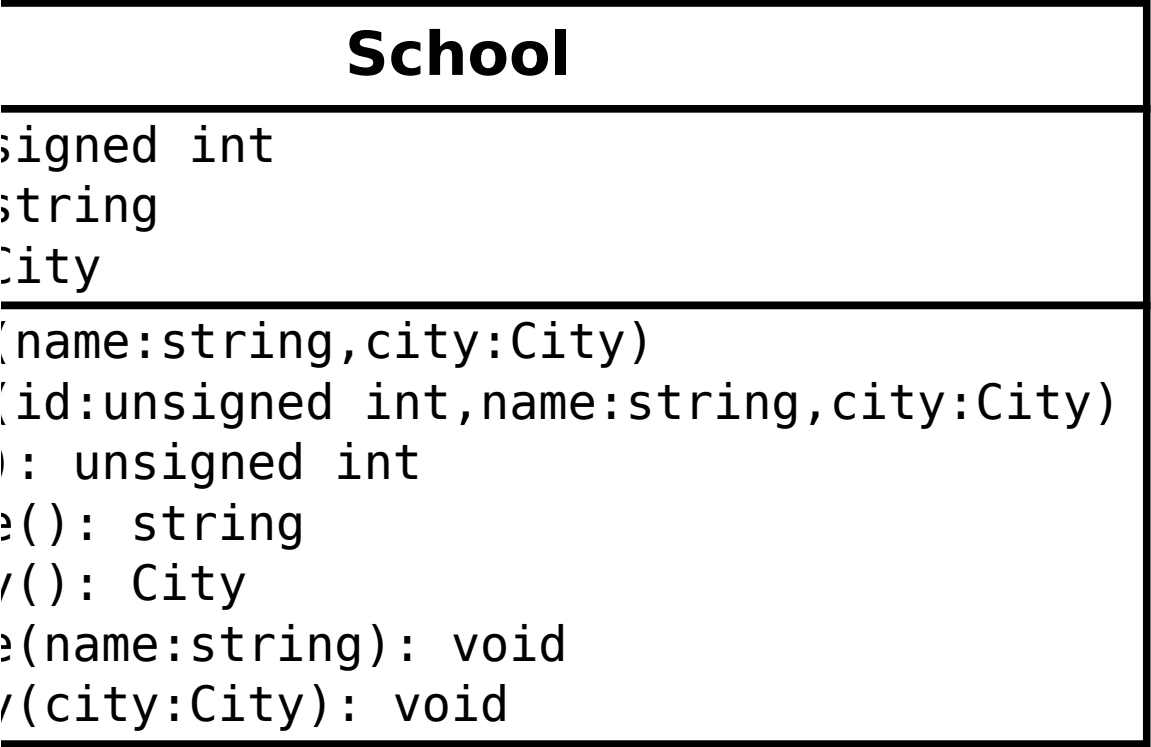
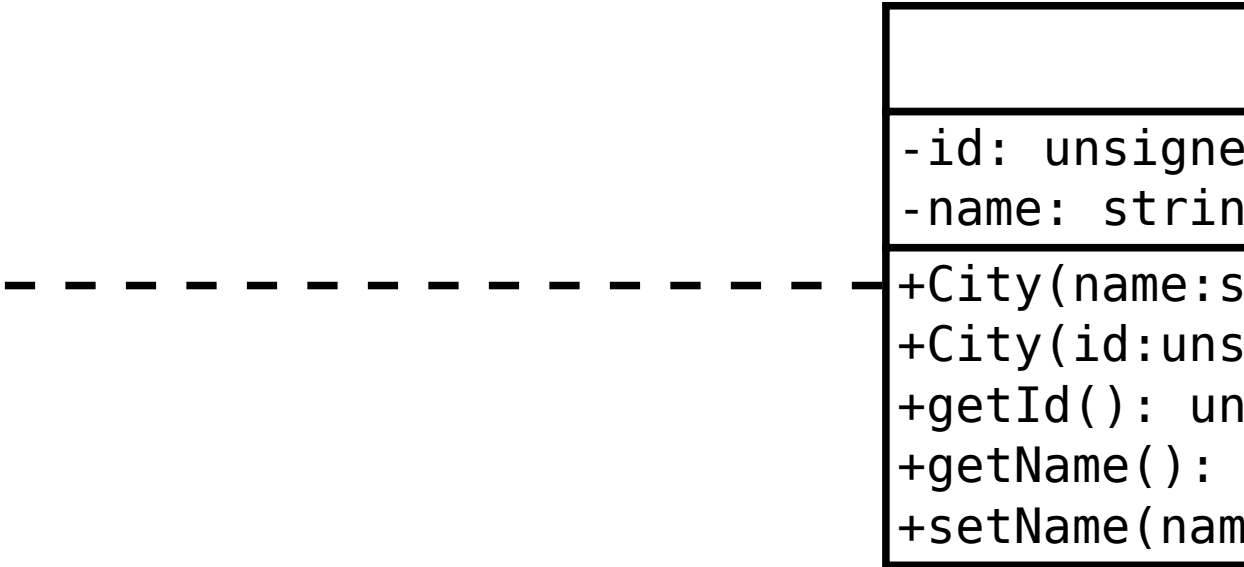


-id: uns -name: s -city: C
+School(+School(+getId() +getName +getCity +setName +setCity

.





City

d int

g

tring)

igned int,name:string)

signed int

string

e:string): void

Student

```
-id: unsigned int  
-name: string  
-school: string  
-members: []
```

```
+Class(name:string)  
+Class(id:unsigned int)  
+getId(): unsigned int  
+getName(): string  
+getSchool(): string  
+getMembers(): []  
+setName(name:string)  
+setSchool(school:string)  
+addMember(member:string)  
+removeMember(member:string)
```

TestType

```
-id: unsigned int  
-name: string
```

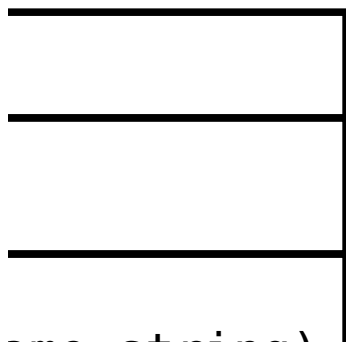
```
+TestType(name:string)
```

```
+TestType(id:unsigned int, name:string)
```

```

    unsigned int
    string
    School
    list<Student>
    name:string,school:School)
    :unsigned int,name:string,school:School)
    unsigned int
    ): string
    l(): School
    rs(): list<Student>
    name:string): void
    l(school:School): void
    r(student:Student): void
    nber(student:Student): void

```

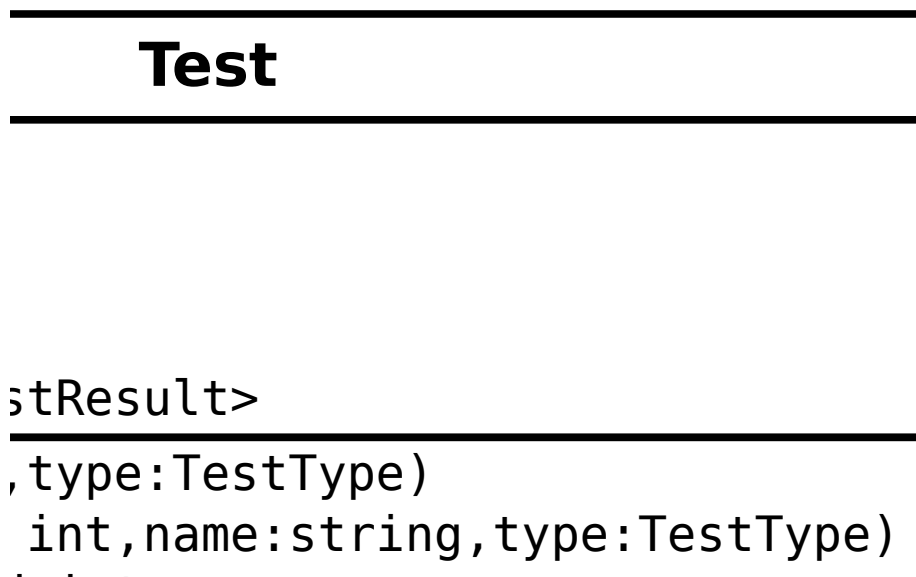


```

-id: unsigned int
-name: string
-type: TestType
-subject: Subject
-results: list<Test>
+Test(name:string,
+Test(id:unsigned

```





```
ned int
: string
string
string
```

```
irstName:string,lastName:string,
urrentClass:Class)
d:unsigned int,firstName:string,
astName:string,currentClass:Class)
unsigned int
ame(): string
me(): string
ts(): string
ame(firstName:string): void
me(lastName:string): void
ts(comments:string): void
```



TestResult



```
-id: unsigned int
-result: float
-student: Student
```



```
+testType(id:unsigned int,n:  
+getId(): unsigned int  
+getName(): string  
+setName(name:string): void
```

ame:string)

```
+getId(): unsigned  
+getName(): string  
+getType(): TestTy  
+getSubject(): Sub  
+getResults(): lis  
+setName(name:string)  
+setType(type:Test  
+setSubject(subjec
```

```
-id: unsigned int  
-name: string  
-isDefault: bool
```

```
+Subject(name:string)  
+Subject(id:unsigned in  
+getId(): unsigned int  
+getName(): string  
+getIsDefault(): bool  
+setName(name:string):  
+setIsDefault(isDefault
```

```
    int  
}  
type  
object  
TestResult  
TestResult(): void  
TestResult(testType): void  
TestResult(subject: Subject): void
```



Subject
TestResult(name: string, isDefault: bool=false)
TestResult(isDefault: bool): void

