«ԾՐԱԳՐԱՎՈՐՄԱՆ ՀԻՄՈՒՆՔՆԵՐ» դասընթաց

այլ ոլորտներից դեպի տեխնոլոգիական ոլորտ սկսնակների համար

edu2020.am

ԴԱՍ #15

ՀԱՅ-ՌՈՒՍԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ
ԲԱՐՁՐ ՏԵԽՆՈԼՈԳԻԱԿԱՆ
ԱՐԴՅՈՒՆԱԲԵՐՈՒԹՅԱՆ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

Դասերի ժառանգում

- Դասի հատկությունների փոխանցումը այլ դասի կոչվում է ժառանգում (inheritance)
- Ժառանգումը օբյեկտա-կողմնորոշված ծրագրավորման երկրորդ սկզբունքն է
- Ժառանգումը հնարավորություն է տալիս խուսափել կոդի կրկնություններից

Դասերի ժառանգում

- ժառանգում base-ից derived

```
class derived : access_mode base{
  //body of subclass
};
```

Դասերի ժառանգում

- ժառանգում base-ից derived

```
class derived : access_mode base{
  //body of subclass
};
```

Ժառանգված դասի անուն

Դասերի ժառանգում

- ժառանգում base-ից derived

```
class derived : access_mode base{
  //body of subclass
};
```

Բազային դասի անուն

Դասերի ժառանգում

- ժառանգում base-ից derived

```
class derived : access_mode base{
  //body of subclass
};
```

Ժառանգման ձև. կարող է լինել public, private, protected

person դաս

```
class person {
 public:
  person(std::string n, int a);
  ~person();
  void printInfo();
 protected:
  std::string name;
  int age;
};
```

```
person::person(std::string n, int a) :
  name(n), age(a) {
  std::cout << "person class constructor\n";
}
```

https://repl.it/@HaykAslanyan/inheritance

person դաս

```
class person {
 public:
  person(std::string n, int a);
  ~person();
  void printInfo();
 protected:
  std::string name;
  int age;
};
```

```
person::~person() {
  std::cout << "person class destructor\n";
}
```

https://repl.it/@HaykAslanyan/inheritance

person դաս

```
class person {
 public:
  person(std::string n, int a);
  ~person();
  void printInfo();
 protected:
  std::string name;
  int age;
};
```

```
void person::printInfo() {
  std::cout << name << " " << age << "\n";
}
```

https://repl.it/@HaykAslanyan/inheritance

student դաս

Ժառանգում person դասից

```
class student : public person {
 public:
  student(std::string n, int a, unsigned y, float g);
  ~student();
  void printInfo();
 protected:
  unsigned year;
  float gpa;
};
```

https://repl.it/@HaykAslanyan/inheritance



student դաս

Ժառանգում person դասից

Ժառանգում է person դասի name, age անդամները

```
class student : public person {
 public:
  student(std::string n, int a, unsigned y, float g);
  ~student();
  void printInfo();
 protected:
  unsigned year;
  float gpa;
};
```

https://repl.it/@HaykAslanyan/inheritance

student դաս

```
class student : public person {
 public:
  student(std::string n, int a, unsigned y, float g);
  ~student();
  void printInfo();
 protected:
  unsigned year;
  float gpa;
};
```

Օգտագործում ենք person դասի կառուցիչը

```
student::student(std::string n, int a, unsigned y, float g) :
  person(n, a), year(y), gpa(g) {
  std::cout << "student class constructor\n";
}
```

https://repl.it/@HaykAslanyan/inheritance



student դաս

```
class student : public person {
 public:
  student(std::string n, int a, unsigned y, float g);
  ~student();
  void printInfo();
 protected:
  unsigned year;
  float gpa;
};
```

Կանչում է person դասի փլուզիչը

```
student::~student() {
  std::cout << "student class destructor\n";
}
```

https://repl.it/@HaykAslanyan/inheritance

student դաս

```
class student : public person {
 public:
  student(std::string n, int a, unsigned y, float g);
  ~student();
  void printInfo();
 protected:
  unsigned year;
  float gpa;
};
```

Վերասահմանում ենք person դասի printInfo() ֆունկցիան

```
void student::printInfo() {
  std::cout << name << " " << age << " "
            << year << " " << gpa << "\n";
}
```

https://repl.it/@HaykAslanyan/inheritance

student դաս

```
int main() {
  //...
  std::cout << "Creating 2 student objects\n";
  student student1("Anahit", 20, 1, 3.8);
  student student2("Anush", 18, 3, 2.4);

  std::cout << "Printing 2 student objects info\n";
  student1.printInfo();
  student2.printInfo();
  std::cout << "Printing student object info with pointer\n";
  student* pointerOnStudent = &student1;
  pointerOnStudent->printInfo();
  //...
}
```

Կկանչվի student դասի printInfo() ֆունկցիան

https://repl.it/@HaykAslanyan/inheritance

worker դաս

Ժառանգում person դասից

```
class worker : public person {
 public:
  worker(std::string n, int a, unsigned s);
  ~worker();
  void printSalary();
 protected:
  unsigned salary;
};
```

https://repl.it/@HaykAslanyan/inheritance



worker դաս

Ժառանգում person դասից

Ժառանգում է person դասի name, age անդամները, printInfo() ֆունկցիան

```
class worker : public person {
 public:
  worker(std::string n, int a, unsigned s);
  ~worker();
  void printSalary();
 protected:
  unsigned salary;
};
```

https://repl.it/@HaykAslanyan/inheritance

worker դաս

```
class worker : public person {
 public:
  worker(std::string n, int a, unsigned s);
  ~worker();
  void printSalary();
 protected:
  unsigned salary;
};
```

Օգտագործում ենք person դասի կառուցիչը

```
worker::worker(std::string n, int a, unsigned s) :
  person(n, a), salary(s) {
  std::cout << "worker constructor\n";
}
```

https://repl.it/@HaykAslanyan/inheritance

worker դաս

```
class worker : public person {
 public:
  worker(std::string n, int a, unsigned s);
  ~worker();
  void printSalary();
 protected:
  unsigned salary;
};
```

Կանչում է person դասի փլուզիչը

```
worker::~worker() {
  std::cout << "worker destructor\n";
}
```

https://repl.it/@HaykAslanyan/inheritance

worker դասը

```
class worker : public person {
 public:
  worker(std::string n, int a, unsigned s);
  ~worker();
  void printSalary();
 protected:
  unsigned salary;
};
```

```
void worker::printSalary() {
  std::cout << "worker salary is " << salary << "\n";
}
```

https://repl.it/@HaykAslanyan/inheritance

worker դաս

```
int main() {
  //...
  std::cout << "Creating 2 worker objects\n";
  worker worker1("Taron", 34, 500000);
  worker worker2("Argishti", 45, 900000);
  std::cout << "Printing worker object info\n";
  worker1.printInfo();
  std::cout << "Printing 2 worker objects salary\n";
  worker1.printSalary();
  worker2.printSalary();
  //...
}
```

Կանչվում է person դասի printInfo() ֆունկցիան

Կանչվում է worker դասի printSalary() ֆունկցիան

https://repl.it/@HaykAslanyan/inheritance

programmer դաս

Ժառանգում worker դասից

```
class programmer : public worker {
 public:
  programmer(std::string n, int a, unsigned s, std::string l);
  ~programmer();
  void printInfo();
 private:
  std::string language;
};
```

https://repl.it/@HaykAslanyan/inheritance

programmer դաս

Ժառանգում worker դասից

Ժառանգում է name, age, salary անդամները, printSalary() ֆունկցիան

```
class programmer : public worker {
 public:
  programmer(std::string n, int a, unsigned s, std::string l);
  ~programmer();
  void printInfo();
 private:
  std::string language;
};
```

https://repl.it/@HaykAslanyan/inheritance

programmer դաս

```
class programmer : public worker {
 public:
  programmer(std::string n, int a, unsigned s, std::string l);
  ~programmer();
  void printInfo();
 private:
  std::string language;
};
```

```
programmer::programmer(std::string n, int a,
                       unsigned s, std::string l) :
  worker(n, a, s), language(l) {
  std::cout << "programmer constructor\n";
}
```

https://repl.it/@HaykAslanyan/inheritance

programmer դաս

```
class programmer : public worker {
 public:
  programmer(std::string n, int a, unsigned s, std::string l);
  ~programmer();
  void printInfo();
 private:
  std::string language;
};
```

Կանչում է worker դասի փլուզիչը

```
programmer::~programmer() {
  std::cout << "programmer destructor\n";
}
```

https://repl.it/@HaykAslanyan/inheritance

programmer դաս

Վերաասիմանում ենք person դասի printInfo() ֆունկցիան

```
class programmer : public worker {
 public:
  programmer(std::string n, int a, unsigned s, std::string l);
  ~programmer();
  void printInfo();
 private:
  std::string language;
};
```

```
void programmer::printInfo() {
  std::cout << name << " " << age << " "
            << salary << " " << language << "\n";
}
```

https://repl.it/@HaykAslanyan/inheritance

programmer դաս

```
int main() {
  //...
  std::cout << "Creating 2 programmer objects\n";
  programmer programmer1 ("Tigran", 29, 800000, "c++");
  programmer programmer2 ("Artashes", 39, 600000, "java");
  std::cout << "Printing objects info and salary\n";
  programmer1.printInfo();
  programmer1.printSalary();
  programmer2.printInfo();
  //...
}
```

Կանչվում է programmer դասի printInfo() ֆունկցիան

Կանչվում է worker դասի printSalary() ֆունկցիան

https://repl.it/@HaykAslanyan/inheritance

Դասերի ժառանգման ձևերը

- Եթե ժառանգում ենք բազային դասից **public** ձևով, ապա public անդամները և ֆունկցիաները ժառանգված դասում մնում են public, protected-ները՝ protected, իսկ private-ները հասանելի չեն
- Եթե ժառանգում ենք բազային դասից **protected** ձևով, ապա public և protected անդամները և ֆունկցիաները ժառանգված դասում դառնում են protected, իսկ private-ները հասանելի չեն
- Եթե ժառանգում ենք բազային դասից **private** ձևով, ապա public և protected անդամները և ֆունկցիաները ժառանգված դասում դառնում են private, իսկ private-ները հասանելի չեն

Դասերի ժառանգման ձևերը

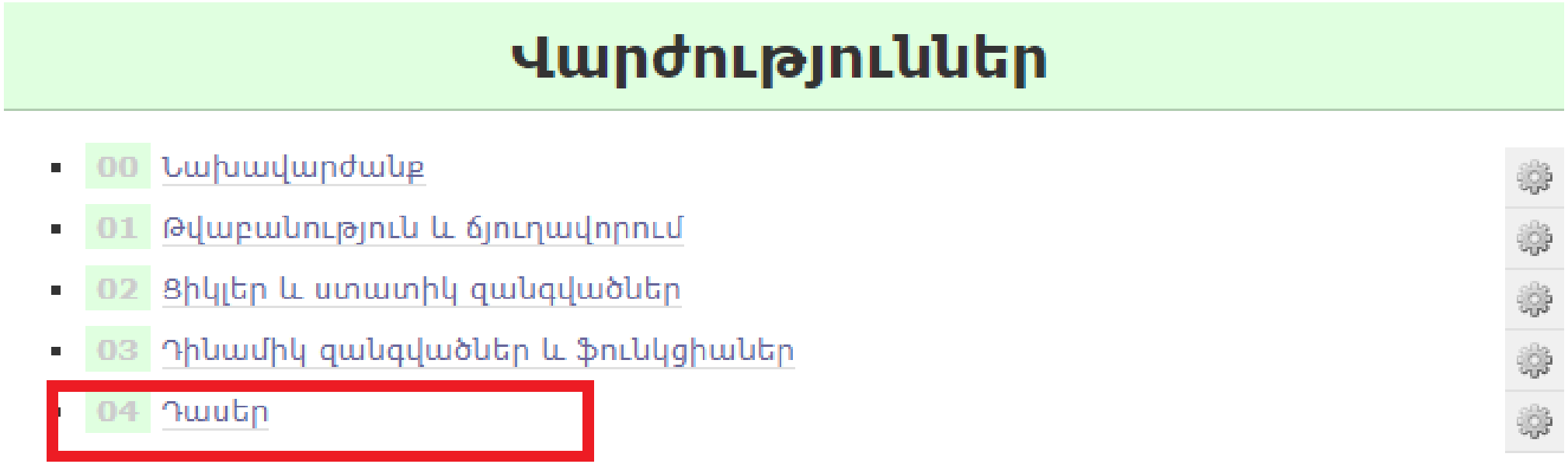
Բազային դասի հասանելիության սպեցիֆիկատորը	Ժառանգության ձևը		
	public	protected	private
public	public	protected	private
protected	protected	protected	private
private	Հասանելի չէ	Հասանելի չէ	Հասանելի չէ

Դասերի ժառանգում

- Գրել vehicle դասը և ժառանգել bus, car, track դասերը
- Գրել fruit դասը և ժառանգել apple, banana, orange դասերը
- Գրել animal դասը և ժառանգել dog, cat, lion դասերը

Տնային աշխատանք

Տնային աշխատանք 12-14, 17-20



Շնորհակալություն. Հարցե՞ր