Bachelor Universitaire technologique - Programmation Orientée Objet

S4

Programmation Orientée Objet

QCM

v0

Avogadro Amedeo

Durée : 60 minutes.

Aucun document n'est autorisé. L'usage de la calculatrice est interdit.

1 Présentation du système

1.1 Présentation générale

L'étude de ce devoir porte sur la mise en place d'une programmation orientée objet sur un système simplifié de tri de pièces. Le système est représenté sur la Figure 7 Ce système reçoit des pièces en provenance d'une chaine de production et les trie en fonction de leur taille. Plus précisément, les pièces arrivent depuis une goulotte inclinée jusqu'à un poste de mesure. Lorsque le convoyeur est libre, la pièce est transférée sur le convoyeur à l'aide du vérin *VTRANS* puis est poussée dans un des bacs selon sa taille : *PETIT*, *MOYEN* ou *GRAND*. Les pièces présentant un défaut restent sur le convoyeur jusqu'au bac 4 servant de mise au rebut.

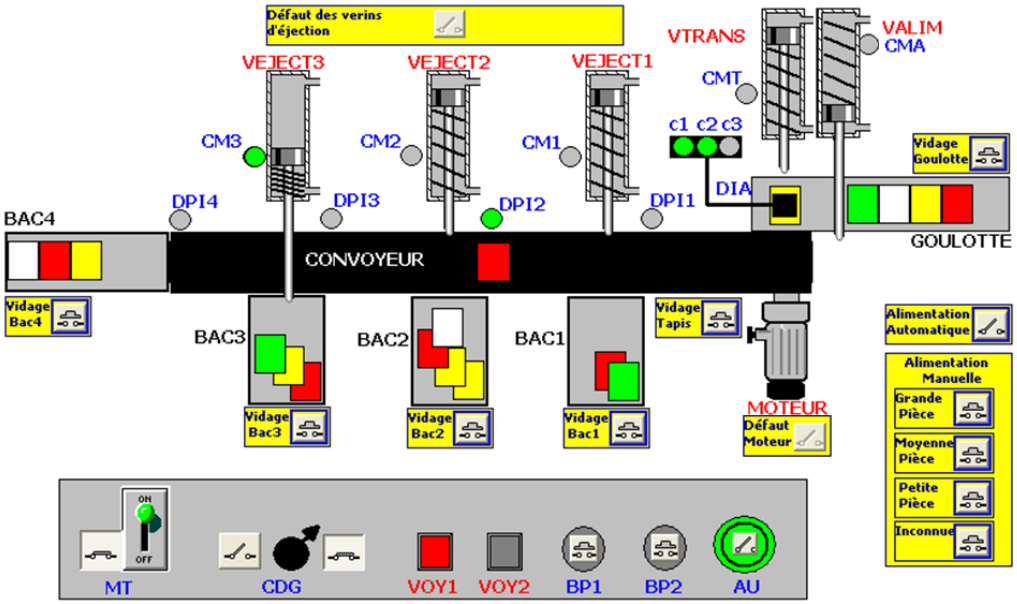

Figure 1 – Système de tri de pièces

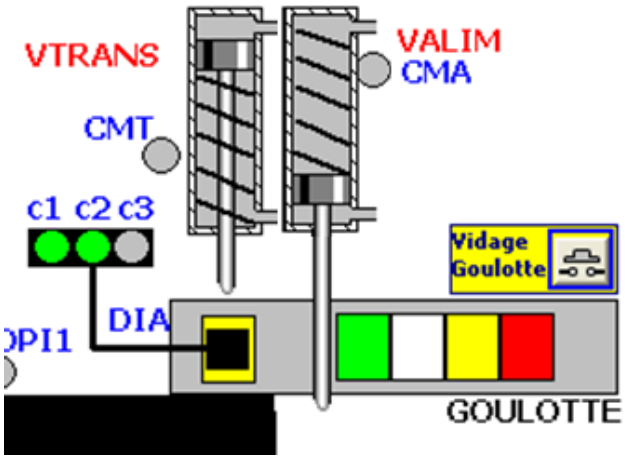

FIGURE 2 – Poste de mesure et de transfert

2 Le poste de mesure – transfert

2.1 Présentation

L'étude se concentre sur la partie de la machine qui permet de mesurer la taille des pièces et de les transférer sur le convoyeur.

La liste des capteurs et actionneurs est donnée dans le tableau suivant :

Capteurs			
Nom	**Type**	**Description**	**Variable**
CMA	Logique	Capteur fin de course vérin Alim	ixAlimRentre
CMT	Logique	Capteur fin de course du vérin de Transfert	ixTransfertSorti
DIA	Numérique	Capteur de hauteur de la pièce (cf Mesure des pièces ci-dessous)	

Actionneurs			
Nom	**Préactionneur**	**Description**	**Variable**
VALIM	Vérin pneumatique simple effet, normalement sorti. Distributeur monostable.	Vérin de blocage des pièces sur la goulotte.	qxAlimRentrer
VTRANS	Vérin pneumatique simple effet normalement rentré. Distributeur bistable.	Vérin de transfert des pièces sur le convoyeur.	qxTransfertSortir

La mesure des pièces est faite par le capteurs *DIA*. Il délivre trois signaux logiques *C1*, *C2* et *C3* en fonction de la hauteur de la pièce. La table de correspondance est donnée dans le tableau suivant :

Hauteur	**ixC1**	**ixC2**	**ixC3**
Petit	1	0	0
Moyen	1	1	0
Grand	1	1	1

Toute autre combinaison de signaux devra considérer la pièce associée comme étant défectueuse.

3 Questions

3.1 Question de cours

Question 1 Rappeler ce qu'est un pointeur *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 2 Qu'est-ce que le polymorphisme ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 3 Rappeler le rôle d'une interface *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

3.2 La classe Vérin

On propose de créer une classe abstraite *CVerin* dont hériteront les classes *CVerinSimpleEffet* et *CVerinDoubleEffet*.

Question 4 Quel est l'intérêt de définir une classe abstraite pour les vérins ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 5 Etablir la liste des missions à traiter pour la classe *CVerin*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 6 Etablir la liste des états possibles pour un vérin. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 7 Etablir la facette électrique d'un vérin simple effet. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 8 Quelle propriétés devront être associées à l'interface *ITF_ObjVerin* ? Préciser le rôle de chacune et les accesseurs à conserver. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 9 Ecrire la ligne de code définissant la classe *CVerinSimpleEffet*. Vous indiquerez les héritages et la (les) interface(s) qu'elle implémente *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 10 A partir de sa facette électrique et de ses propriétés, établir le contexte du bloc fonctionnel *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 11 Donner le corps des accesseurs associés aux propriétés définies. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 12 Etablir le constructeur *FB_Init* de la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 13 Etablir le constructeur d'état de la classe *CVerinSimpleEffet*. Compléter, au besoin, le contexte du bloc fonctionnel. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 14 Etablir la méthode de prise en compte des missions pour la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 15 Etablir la commande des actionneurs pour la classe CVerinSimpleEffet. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 16 Etablir le corps de la méthode *Schedule* de la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 17 Donner la déclaration et l'instanciation du bloc fonctionnel *CVerinSimpleEffet* dans le programme principal. Penser à lier les entrées et sorties. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

4 Mesures

Le poste de mesure prend en entrées les sorties du capteurs DIA afin de donner comme état de sortie la taille de la pièce mesurée. On s'intéresse au constructeur d'état de ce poste de mesure.

Question 18 Etablir le type énuméré *ETAT_MESURE* pour la classe *CMesures*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 19 Etablir le constructeur d'état de la classe *CMesures*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Bachelor Universitaire technologique - Programmation Orientée Objet

S4

Programmation Orientée Objet

QCM

v0

Bohr Niels

Durée : 60 minutes.

Aucun document n'est autorisé. L'usage de la calculatrice est interdit.

1 Présentation du système

1.1 Présentation générale

L'étude de ce devoir porte sur la mise en place d'une programmation orientée objet sur un système simplifié de tri de pièces. Le système est représenté sur la Figure 7 Ce système reçoit des pièces en provenance d'une chaine de production et les trie en fonction de leur taille. Plus précisément, les pièces arrivent depuis une goulotte inclinée jusqu'à un poste de mesure. Lorsque le convoyeur est libre, la pièce est transférée sur le convoyeur à l'aide du vérin *VTRANS* puis est poussée dans un des bacs selon sa taille : *PETIT*, *MOYEN* ou *GRAND*. Les pièces présentant un défaut restent sur le convoyeur jusqu'au bac 4 servant de mise au rebut.

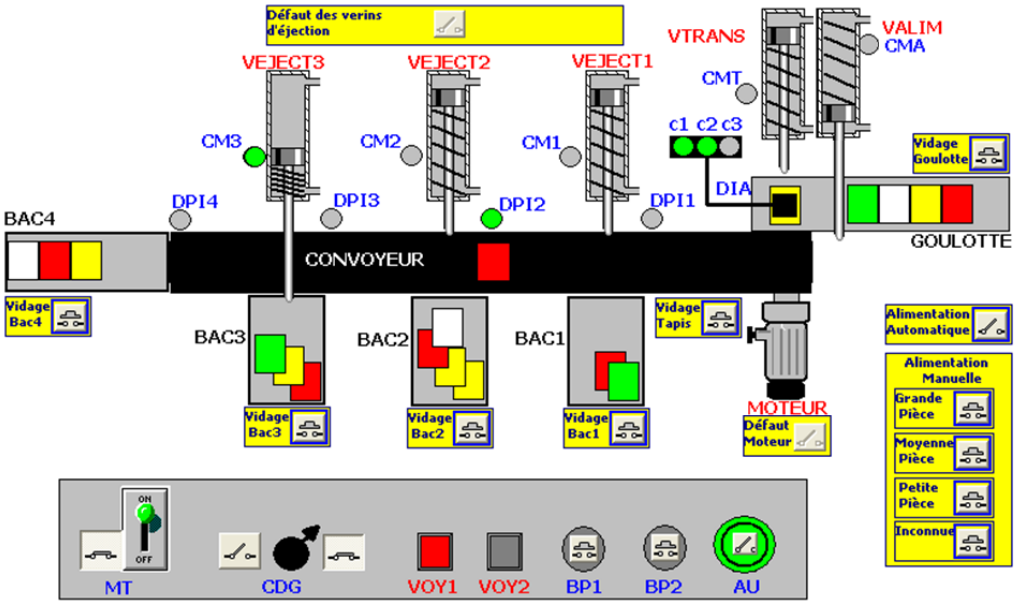

Figure 3 – Système de tri de pièces

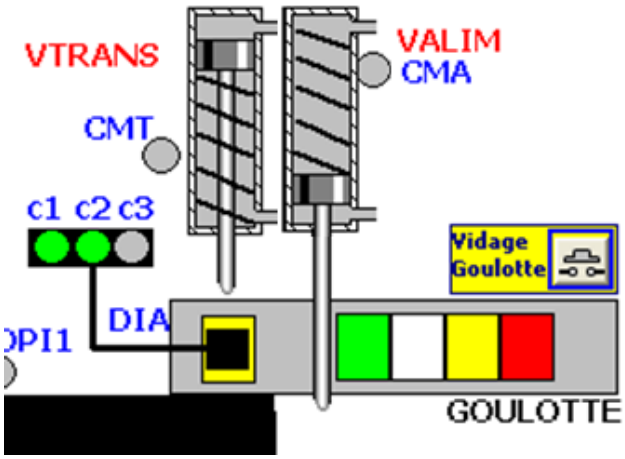

FIGURE 4 – Poste de mesure et de transfert

2 Le poste de mesure – transfert

2.1 Présentation

L'étude se concentre sur la partie de la machine qui permet de mesurer la taille des pièces et de les transférer sur le convoyeur.

La liste des capteurs et actionneurs est donnée dans le tableau suivant :

Capteurs			
Nom	**Type**	**Description**	**Variable**
CMA	Logique	Capteur fin de course vérin Alim	ixAlimRentre
CMT	Logique	Capteur fin de course du vérin de Transfert	ixTransfertSorti
DIA	Numérique	Capteur de hauteur de la pièce (cf Mesure des pièces ci-dessous)	

Actionneurs			
Nom	**Préactionneur**	**Description**	**Variable**
VALIM	Vérin pneumatique simple effet, normalement sorti. Distributeur monostable.	Vérin de blocage des pièces sur la goulotte.	qxAlimRentrer
VTRANS	Vérin pneumatique simple effet normalement rentré. Distributeur bistable.	Vérin de transfert des pièces sur le convoyeur.	qxTransfertSortir

La mesure des pièces est faite par le capteurs *DIA*. Il délivre trois signaux logiques *C1*, *C2* et *C3* en fonction de la hauteur de la pièce. La table de correspondance est donnée dans le tableau suivant :

Hauteur	**ixC1**	**ixC2**	**ixC3**
Petit	1	0	0
Moyen	1	1	0
Grand	1	1	1

Toute autre combinaison de signaux devra considérer la pièce associée comme étant défectueuse.

3 Questions

3.1 Question de cours

Question 1 Rappeler ce qu'est un pointeur *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 2 Rappeler le rôle d'une interface *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 3 Qu'est-ce que le polymorphisme ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 4 Qu'est-ce que le polymorphisme ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 5 Rappeler ce qu'est un pointeur *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 6 Rappeler le rôle d'une interface *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

3.2 La classe Vérin

On propose de créer une classe abstraite *CVerin* dont hériteront les classes *CVerinSimpleEffet* et *CVerinDoubleEffet*.

Question 7 Quel est l'intérêt de définir une classe abstraite pour les vérins ? *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 8 Etablir la liste des missions à traiter pour la classe *CVerin*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 9 Etablir la liste des états possibles pour un vérin. *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 10 Etablir la facette électrique d'un vérin simple effet. *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 11 Quelle propriétés devront être associées à l'interface *ITF_ObjVerin* ? Préciser le rôle de chacune et les accesseurs à conserver. *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 12 Ecrire la ligne de code définissant la classe *CVerinSimpleEffet*. Vous indiquerez les héritages et la (les) interface(s) qu'elle implémente *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 13 A partir de sa facette électrique et de ses propriétés, établir le contexte du bloc fonctionnel *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 14 Donner le corps des accesseurs associés aux propriétés définies. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 15 Etablir le constructeur *FB_Init* de la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 16 Etablir le constructeur d'état de la classe *CVerinSimpleEffet*. Compléter, au besoin, le contexte du bloc fonctionnel. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 17 Etablir la méthode de prise en compte des missions pour la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 18 Etablir la commande des actionneurs pour la classe CVerinSimpleEffet. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 19 Etablir le corps de la méthode *Schedule* de la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 20 Donner la déclaration et l'instanciation du bloc fonctionnel *CVerinSimpleEffet* dans le programme principal. Penser à lier les entrées et sorties. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

4 Mesures

Le poste de mesure prend en entrées les sorties du capteurs DIA afin de donner comme état de sortie la taille de la pièce mesurée. On s'intéresse au constructeur d'état de ce poste de mesure.

Question 21 Etablir le type énuméré *ETAT_MESURE* pour la classe *CMesures*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 22 Etablir le constructeur d'état de la classe *CMesures*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Bachelor Universitaire technologique - Programmation Orientée Objet

S4

Programmation Orientée Objet

QCM

v0

Copernic Nicolas

Durée : 60 minutes.

Aucun document n'est autorisé. L'usage de la calculatrice est interdit.

1 Présentation du système

1.1 Présentation générale

L'étude de ce devoir porte sur la mise en place d'une programmation orientée objet sur un système simplifié de tri de pièces. Le système est représenté sur la Figure 7 Ce système reçoit des pièces en provenance d'une chaine de production et les trie en fonction de leur taille. Plus précisément, les pièces arrivent depuis une goulotte inclinée jusqu'à un poste de mesure. Lorsque le convoyeur est libre, la pièce est transférée sur le convoyeur à l'aide du vérin *VTRANS* puis est poussée dans un des bacs selon sa taille : *PETIT*, *MOYEN* ou *GRAND*. Les pièces présentant un défaut restent sur le convoyeur jusqu'au bac 4 servant de mise au rebut.

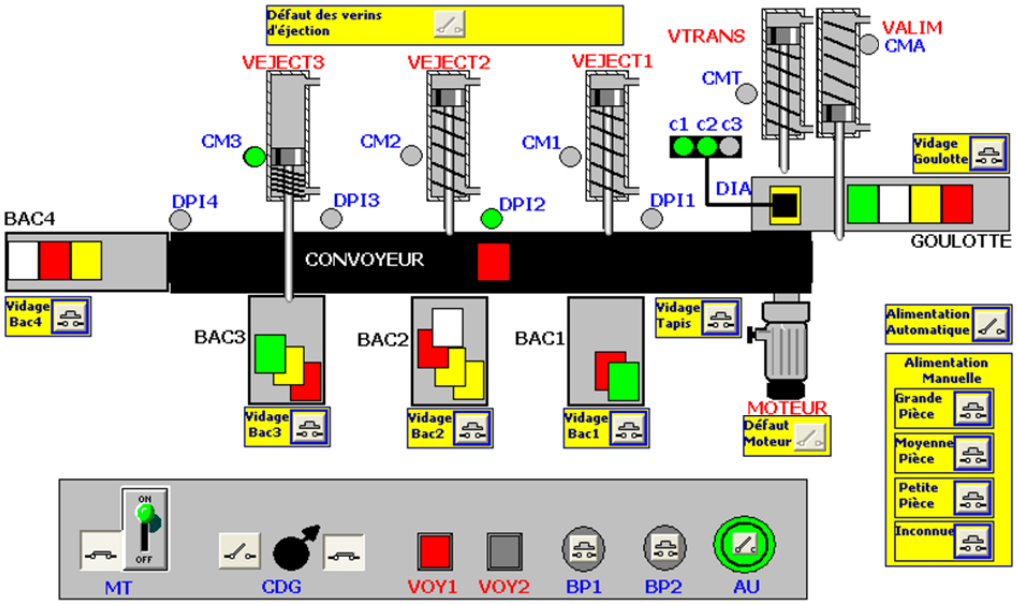

Figure 5 – Système de tri de pièces

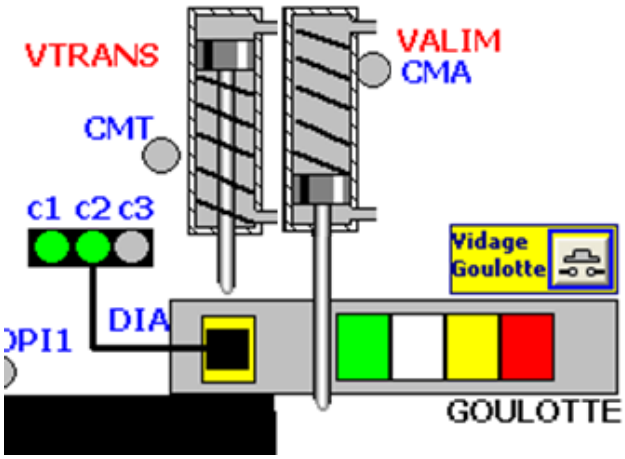

FIGURE 6 – Poste de mesure et de transfert

2 Le poste de mesure – transfert

2.1 Présentation

L'étude se concentre sur la partie de la machine qui permet de mesurer la taille des pièces et de les transférer sur le convoyeur.

La liste des capteurs et actionneurs est donnée dans le tableau suivant :

Capteurs			
Nom	**Type**	**Description**	**Variable**
CMA	Logique	Capteur fin de course vérin Alim	ixAlimRentre
CMT	Logique	Capteur fin de course du vérin de Transfert	ixTransfertSorti
DIA	Numérique	Capteur de hauteur de la pièce (cf Mesure des pièces ci-dessous)	

Actionneurs			
Nom	**Préactionneur**	**Description**	**Variable**
VALIM	Vérin pneumatique simple effet, normalement sorti. Distributeur monostable.	Vérin de blocage des pièces sur la goulotte.	qxAlimRentrer
VTRANS	Vérin pneumatique simple effet normalement rentré. Distributeur bistable.	Vérin de transfert des pièces sur le convoyeur.	qxTransfertSortir

La mesure des pièces est faite par le capteurs *DIA*. Il délivre trois signaux logiques *C1*, *C2* et *C3* en fonction de la hauteur de la pièce. La table de correspondance est donnée dans le tableau suivant :

Hauteur	**ixC1**	**ixC2**	**ixC3**
Petit	1	0	0
Moyen	1	1	0
Grand	1	1	1

Toute autre combinaison de signaux devra considérer la pièce associée comme étant défectueuse.

3 Questions

3.1 Question de cours

Question 1 Qu'est-ce que le polymorphisme ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 2 Qu'est-ce que le polymorphisme ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 3 Rappeler ce qu'est un pointeur *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 4 Rappeler le rôle d'une interface *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 5 Rappeler ce qu'est un pointeur *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 6 Rappeler le rôle d'une interface *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 7 Qu'est-ce que le polymorphisme ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 8 Rappeler ce qu'est un pointeur *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 9 Rappeler le rôle d'une interface *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

3.2 La classe Vérin

On propose de créer une classe abstraite *CVerin* dont hériteront les classes *CVerinSimpleEffet* et *CVerinDoubleEffet*.

Question 10 Quel est l'intérêt de définir une classe abstraite pour les vérins ? *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 11 Etablir la liste des missions à traiter pour la classe *CVerin*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 12 Etablir la liste des états possibles pour un vérin. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 13 Etablir la facette électrique d'un vérin simple effet. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 14 Quelle propriétés devront être associées à l'interface *ITF_ ObjVerin* ? Préciser le rôle de chacune et les accesseurs à conserver. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 15 Ecrire la ligne de code définissant la classe *CVerinSimpleEffet*. Vous indiquerez les héritages et la (les) interface(s) qu'elle implémente *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 16 A partir de sa facette électrique et de ses propriétés, établir le contexte du bloc fonctionnel *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 17 Donner le corps des accesseurs associés aux propriétés définies. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 18 Etablir le constructeur *FB_Init* de la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 19 Etablir le constructeur d'état de la classe *CVerinSimpleEffet*. Compléter, au besoin, le contexte du bloc fonctionnel. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 20 Etablir la méthode de prise en compte des missions pour la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 21 Etablir la commande des actionneurs pour la classe CVerinSimpleEffet. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 22 Etablir le corps de la méthode *Schedule* de la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 23 Donner la déclaration et l'instanciation du bloc fonctionnel *CVerinSimpleEffet* dans le programme principal. Penser à lier les entrées et sorties. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

4 Mesures

Le poste de mesure prend en entrées les sorties du capteurs DIA afin de donner comme état de sortie la taille de la pièce mesurée. On s'intéresse au constructeur d'état de ce poste de mesure.

Question 24 Etablir le type énuméré *ETAT_MESURE* pour la classe *CMesures*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 25 Etablir le constructeur d'état de la classe *CMesures*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Bachelor Universitaire technologique - Programmation Orientée Objet

S4

Programmation Orientée Objet

QCM

v0

Einstein Albert

Durée : 60 minutes.

Aucun document n'est autorisé. L'usage de la calculatrice est interdit.

1 Présentation du système

1.1 Présentation générale

L'étude de ce devoir porte sur la mise en place d'une programmation orientée objet sur un système simplifié de tri de pièces. Le système est représenté sur la Figure 7 Ce système reçoit des pièces en provenance d'une chaine de production et les trie en fonction de leur taille. Plus précisément, les pièces arrivent depuis une goulotte inclinée jusqu'à un poste de mesure. Lorsque le convoyeur est libre, la pièce est transférée sur le convoyeur à l'aide du vérin *VTRANS* puis est poussée dans un des bacs selon sa taille : *PETIT*, *MOYEN* ou *GRAND*. Les pièces présentant un défaut restent sur le convoyeur jusqu'au bac 4 servant de mise au rebut.

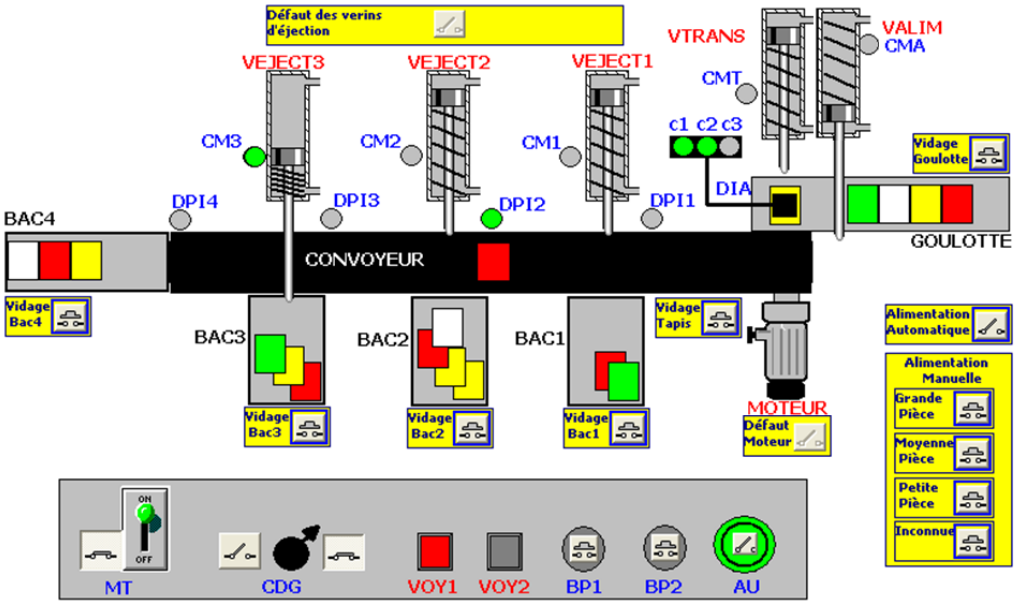

Figure 7 – Système de tri de pièces

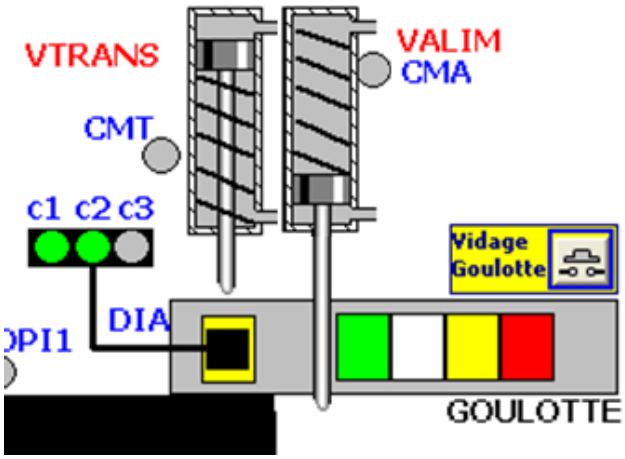

FIGURE 8 – Poste de mesure et de transfert

2 Le poste de mesure – transfert

2.1 Présentation

L'étude se concentre sur la partie de la machine qui permet de mesurer la taille des pièces et de les transférer sur le convoyeur.

La liste des capteurs et actionneurs est donnée dans le tableau suivant :

Capteurs			
Nom	**Type**	**Description**	**Variable**
CMA	Logique	Capteur fin de course vérin Alim	ixAlimRentre
CMT	Logique	Capteur fin de course du vérin de Transfert	ixTransfertSorti
DIA	Numérique	Capteur de hauteur de la pièce (cf Mesure des pièces ci-dessous)	

Actionneurs			
Nom	**Préactionneur**	**Description**	**Variable**
VALIM	Vérin pneumatique simple effet, normalement sorti. Distributeur monostable.	Vérin de blocage des pièces sur la goulotte.	qxAlimRentrer
VTRANS	Vérin pneumatique simple effet normalement rentré. Distributeur bistable.	Vérin de transfert des pièces sur le convoyeur.	qxTransfertSortir

La mesure des pièces est faite par le capteurs *DIA*. Il délivre trois signaux logiques *C1*, *C2* et *C3* en fonction de la hauteur de la pièce. La table de correspondance est donnée dans le tableau suivant :

Hauteur	**ixC1**	**ixC2**	**ixC3**
Petit	1	0	0
Moyen	1	1	0
Grand	1	1	1

Toute autre combinaison de signaux devra considérer la pièce associée comme étant défectueuse.

3 Questions

3.1 Question de cours

Question 1 Qu'est-ce que le polymorphisme ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 2 Rappeler ce qu'est un pointeur *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 3 Qu'est-ce que le polymorphisme ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 4 Rappeler le rôle d'une interface *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 5 Rappeler le rôle d'une interface *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 6 Rappeler ce qu'est un pointeur *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 7 Qu'est-ce que le polymorphisme ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 8 Rappeler le rôle d'une interface *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 9 Rappeler ce qu'est un pointeur *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 10 Qu'est-ce que le polymorphisme ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 11 Rappeler le rôle d'une interface *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 12 Rappeler ce qu'est un pointeur *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

3.2 La classe Vérin

On propose de créer une classe abstraite *CVerin* dont hériteront les classes *CVerinSimpleEffet* et *CVerinDoubleEffet*.

Question 13 Quel est l'intérêt de définir une classe abstraite pour les vérins ? *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 14 Etablir la liste des missions à traiter pour la classe *CVerin*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 15 Etablir la liste des états possibles pour un vérin. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 16 Etablir la facette électrique d'un vérin simple effet. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 17 Quelle propriétés devront être associées à l'interface *ITF_ ObjVerin* ? Préciser le rôle de chacune et les accesseurs à conserver. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 18 Ecrire la ligne de code définissant la classe *CVerinSimpleEffet*. Vous indiquerez les héritages et la (les) interface(s) qu'elle implémente *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 19 A partir de sa facette électrique et de ses propriétés, établir le contexte du bloc fonctionnel *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 20 Donner le corps des accesseurs associés aux propriétés définies. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 21 Etablir le constructeur *FB_Init* de la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 22 Etablir le constructeur d'état de la classe *CVerinSimpleEffet*. Compléter, au besoin, le contexte du bloc fonctionnel. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 23 Etablir la méthode de prise en compte des missions pour la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 24 Etablir la commande des actionneurs pour la classe CVerinSimpleEffet. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 25 Etablir le corps de la méthode *Schedule* de la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 26 Donner la déclaration et l'instanciation du bloc fonctionnel *CVerinSimpleEffet* dans le programme principal. Penser à lier les entrées et sorties. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 27 Quel est l'intérêt de définir une classe abstraite pour les vérins ? *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 28 Etablir la liste des missions à traiter pour la classe *CVerin*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 29 Etablir la liste des états possibles pour un vérin. *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 30 Etablir la facette électrique d'un vérin simple effet. *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 31 Quelle propriétés devront être associées à l'interface *ITF_ObjVerin* ? Préciser le rôle de chacune et les accesseurs à conserver. *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 32 Ecrire la ligne de code définissant la classe *CVerinSimpleEffet*. Vous indiquerez les héritages et la (les) interface(s) qu'elle implémente *(1 pt(s))* ☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 33 A partir de sa facette électrique et de ses propriétés, établir le contexte du bloc fonctionnel *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 34 Donner le corps des accesseurs associés aux propriétés définies. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 35 Etablir le constructeur *FB_Init* de la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 36 Etablir le constructeur d'état de la classe *CVerinSimpleEffet*. Compléter, au besoin, le contexte du bloc fonctionnel. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 37 Etablir la méthode de prise en compte des missions pour la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 38 Etablir la commande des actionneurs pour la classe CVerinSimpleEffet. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 39 Etablir le corps de la méthode *Schedule* de la classe *CVerinSimpleEffet*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 40 Donner la déclaration et l'instanciation du bloc fonctionnel *CVerinSimpleEffet* dans le programme principal. Penser à lier les entrées et sorties. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

4 Mesures

Le poste de mesure prend en entrées les sorties du capteurs DIA afin de donner comme état de sortie la taille de la pièce mesurée. On s'intéresse au constructeur d'état de ce poste de mesure.

Question 41 Etablir le type énuméré *ETAT_MESURE* pour la classe *CMesures*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 42 Etablir le constructeur d'état de la classe *CMesures*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 43 Etablir le type énuméré *ETAT_MESURE* pour la classe *CMesures*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 44 Etablir le constructeur d'état de la classe *CMesures*. *(1 pt(s))*

☐0 ☐1 ☐2 ☐3 ☐4 ☐5